assignment01_question_1_2

Jeff Nguyen

January 23, 2021

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

Assignment 01, Question 1&2 Student Name: Ngoc Son (Jeff) Nguyen

Question 1

To estimate the β coefficients of the single stock (AAPL) CAPM portfolio, we follow these steps:

1. Download data from Yahoo Finance. The following tickers are downloaded: AAPL, GSPC (representing the S&P 500) and TNX (representing the US Treasury Bills.)
2. Calculate the following statistics of the portfolio using adjusted monthly closing price (converted from daily): monthly returns (AAPL, GSPC and TNX), mean returns.
3. The expected risk premium of the portfolio based on the CAPM model is given as:

$$
\begin{aligned}
E[R_a - R_f] =& \beta * (R_m - R_f) \\
&\text{or} \\
R_a =& R_f + \beta * (R_m - R_f) \\
R_a - R_f =& \alpha_{Jensen} + \beta * (R_m - R_f) \\
&\text{or} \\
Y =& \alpha_{Jensen} + \beta * X + epsilon \\
\text{with} \\
Y =& R_a - R_f \\
X =& R_m - R_f \\
\beta =& \text{Market risk or systematic risk} \\
\epsilon =& \text{stochastic error term}
\end{aligned} \tag{1}
$$

```
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.
```

```r
# Set start date and end date of data
start_date <- "2018-01-01"
end_date <- "2020-12-31"

# Get data
getSymbols("AAPL", 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] "AAPL"
```

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

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

```r
getSymbols("^TNX", src = "yahoo", from = start_date, to = end_date) # TNX (10-year T-bill)
```

```
## Warning: ^TNX contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.

## [1] "^TNX"
```

```r
# Adjusted Prices
adjAAPL <- AAPL$AAPL.Adjusted

# Get adjusted returns data
rAAPL <- diff(log(to.monthly(AAPL)$AAPL.Adjusted))
rGSPC <- diff(log(to.monthly(GSPC)$GSPC.Adjusted))
rTNX <- (to.monthly(TNX)$TNX.Adjusted) / 1200 # Using monthly rate
```

```
## Warning in to.period(x, "months", indexAt = indexAt, name = name, ...): missing
## values removed from data
```

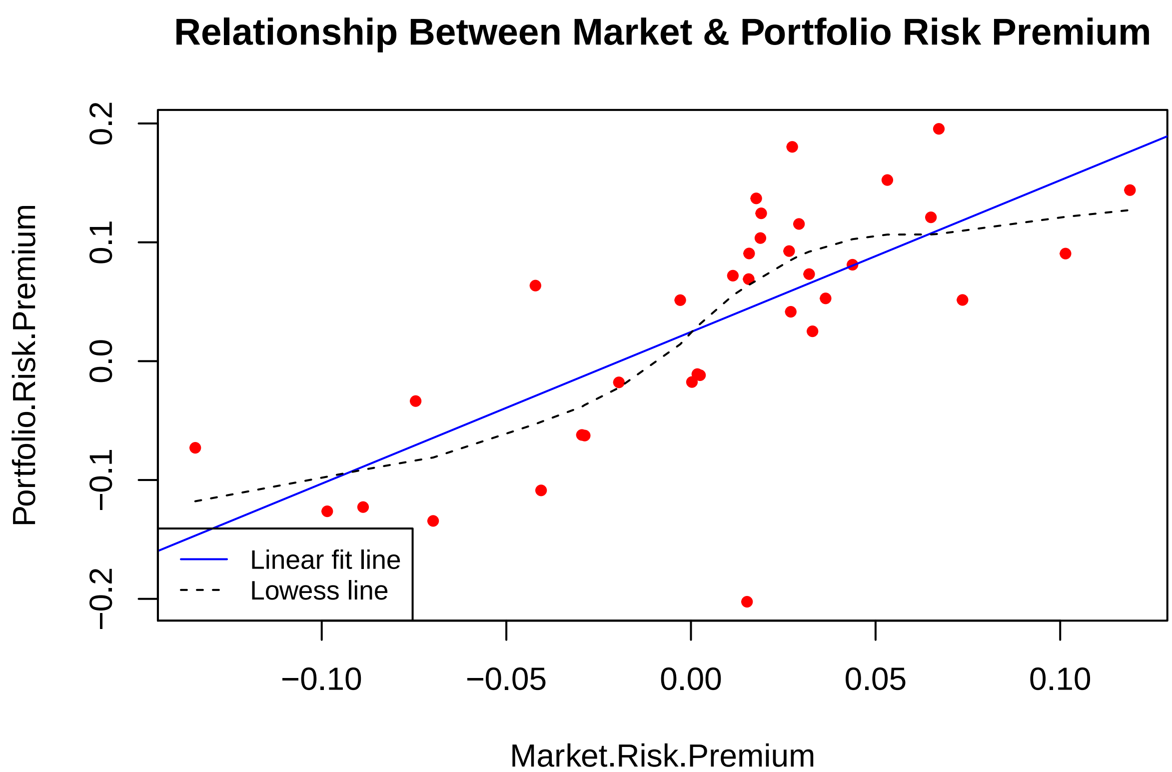
```
# Calculate Portfolio Risk Premium
Y <- na.omit(rAAPL - rTNX)
names(Y)[1] <- "Portfolio Risk Premium"  # Rename column
Y_bar <- mean(Y)
Y_bar
```

```
## [1] 0.03269352
```

```
# Calculate Market Risk Premium
X <- na.omit(rGSPC - rTNX)
X_bar <- mean(X)
names(X)[1] <- "Market Risk Premium"  # Rename column

# Create a data frame of X, Y
data1 <- data.frame(X, Y)
Y_data <- data1$Portfolio.Risk.Premium
X_data <- data1$Market.Risk.Premium
```

We can observe the relationship between the market (S&P 500) risk premium and the portfolio (AAPL) risk premium as follow:

```
plot(data1, col='red', main="Relationship Between Market & Portfolio Risk Premium"
     , pch=20, cex=1)

# Add fit lines
abline(lm(Y_data~X_data), col="blue")  # Regression line Y ~ X
lines(lowess(X_data,Y_data), col="black", lty=2)  # Lowess line (X,Y)

legend("bottomleft",c("Linear fit line", "Lowess line"), cex=.8, col=c("blue", "black")
       , lty=1:2)
```



a. Relationship of AAPL with market

Upon inspection, we observe a strong positive correlation between the AAPL portfolio and the market. Therefore, we can conclude that AAPL follow the market relatively closely, but not necessarily linearly.

b. Hypothesis testing

Next, we attempts to fit an equation of a line: $Y = \alpha_{Jensen} + \beta * X + \epsilon$

Hypothesis for regression:

$$
\begin{aligned}
H_0 &: \alpha = .05 \\
H_a &: \alpha \neq .05 \\
&\text{and} \\
H_0 &: \beta = .05 \\
H_a &: \beta \neq .05
\end{aligned} \tag{2}
$$

```
fit1 <- lm(Y_data~X_data, data=data1)
summary(fit1)
```

```
##
## Call:
## lm(formula = Y_data ~ X_data, data = data1)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.246364 -0.040481  0.000741  0.049734  0.120726
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02455    0.01195   2.055   0.0479 *
## X_data       1.27630    0.22000   5.801 1.73e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0702 on 33 degrees of freedom
## Multiple R-squared:  0.5049, Adjusted R-squared:  0.4899
## F-statistic: 33.66 on 1 and 33 DF,  p-value: 1.731e-06
```

The estimated equation is $Y = .02455 - 1.27630 * X$, where the p_{value} for the intercept $.0479 < .05$. Therefore, we reject the null hypothesis at 95% confidence level that the intercept α_{Jensen} statistically is no different from .05. Thus, we reject the null hypothesis $H_0 : \alpha = .05$ and accept the null hypothesis $H_a : \alpha \neq .05$.
The coefficient $\beta = 1.27630$ represents the increase in portfolio risk premium relative to increase in the market portfolio risk premium. The p_{value} for β is $1.73e - 06 < .05$, implying that the coefficient β statistically is significant at 95% or more, and we reject the null hypothesis $H_0 : \beta = .05$ and accept the alternative hypothesis $H_a : \beta \neq .05$.

Goodness of Fit:
Through inspection of the linear regression result, we observe the $R^2 = .5049$ value to not be close to 1 at all. $R^2 = .5049$ implies that 50.49 of the variations in the portfolio risk premium is explained by the market risk premium.

Standard Error of Regression:
We can see that the Standard Error of Regression is $S.E. = .0702$.
From this, we can calculate the forecasting efficiency statistic to be:

$$\begin{aligned} \frac{S.E.}{\overline{Y}} &= \frac{.0702}{0.0326935} \\ &= 214.72\% > 10\% \end{aligned} \tag{3}$$

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 confirm our initial observation that the portfolio risk premium and the market portfolio risk premium has a weak linear relationship.

c. Ex-post forecast

Doing the ex-post forecast:

```r
# Split the data
data1_1 <- data1[1:32,]
data1_2 <- data1[33:35,]

# run regression for data 1_1
reg_expost <- lm(Y_data~X_data, data=data1_1)

# Predicting using data1_2
pred_expost <- predict(reg_expost, newdata=data1_2, se.fit=TRUE)
```

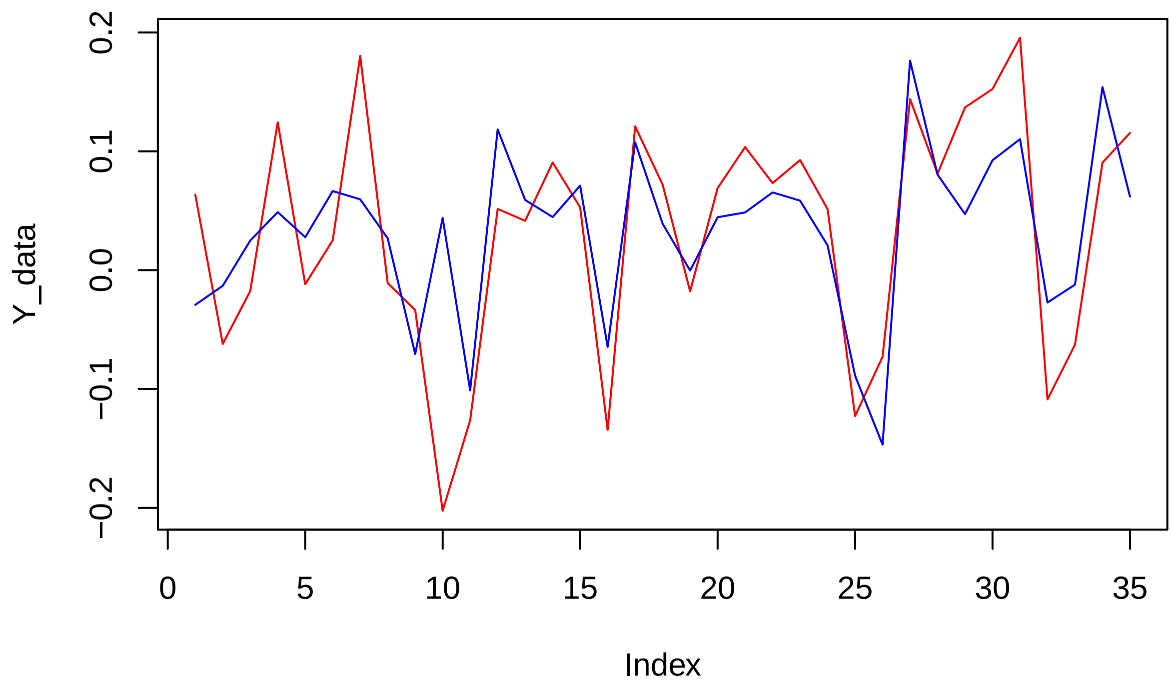
```
## Warning: 'newdata' had 3 rows but variables found have 35 rows
```

```
pred_expost
```

```
## $fit
##             1             2             3             4             5 
## -0.0292002574 -0.0131451118  0.0248951196  0.0488360917  0.0276878725 
##             6             7             8             9            10 
##  0.0665657205  0.0595701459  0.0267712897 -0.0706107568  0.0439407076 
##            11            12            13            14            15 
## -0.1011675606  0.1184197113  0.0590590793  0.0446591692  0.0710986451 
##            16            17            18            19            20 
## -0.0645662607  0.1075013673  0.0390494651 -0.0003509916  0.0445131260 
##            21            22            23            24            25 
##  0.0485682615  0.0654040513  0.0584888325  0.0208562611 -0.0887812600 
##            26            27            28            29            30 
## -0.1467899782  0.1763078180  0.0803862413  0.0471139581  0.0924391116 
##            31            32            33            34            35 
##  0.1102453466 -0.0272426414 -0.0121696291  0.1540243698  0.0618960263 
## 
## $se.fit
##          1          2          3          4          5          6          7 
## 0.01595675 0.01425577 0.01194172 0.01218770 0.01189715 0.01322447 0.01273813 
##          8          9         10         11         12         13         14 
## 0.01190965 0.02139796 0.01202315 0.02594596 0.01895118 0.01270635 0.01204374 
##         15         16         17         18         19         20         21 
## 0.01358752 0.02053906 0.01752341 0.01191629 0.01316210 0.01203945 0.01217724 
##         22         23         24         25         26         27         28 
## 0.01313729 0.01267153 0.01203997 0.02406709 0.03313511 0.02745178 0.01443535 
##         29         30         31         32         33         34         35 
## 0.01212337 0.01571159 0.01787432 0.01573314 0.01416327 0.02404551 0.01288935 
## 
## $df
## [1] 33
## 
## $residual.scale
## [1] 0.07019913
```

Plot the ex-post forecast

```
plot(Y_data, type="l", col="red")
lines(reg_expost$fitted.values, col="blue")
```



We can observe that the forecast (blue line) has trend that somewhat follow the actual result (red line) with the forecast overall having less variations than actual result.

d. Ex-ante forecast.

```
library(forecast)

# Run regression using all data in sample (data1)
reg_exante <- lm(Y_data~X_data, data=data1)
reg_exante
```

```
##
## Call:
## lm(formula = Y_data ~ X_data, data = data1)
##
## Coefficients:
## (Intercept)       X_data
##     0.02455      1.27630
```

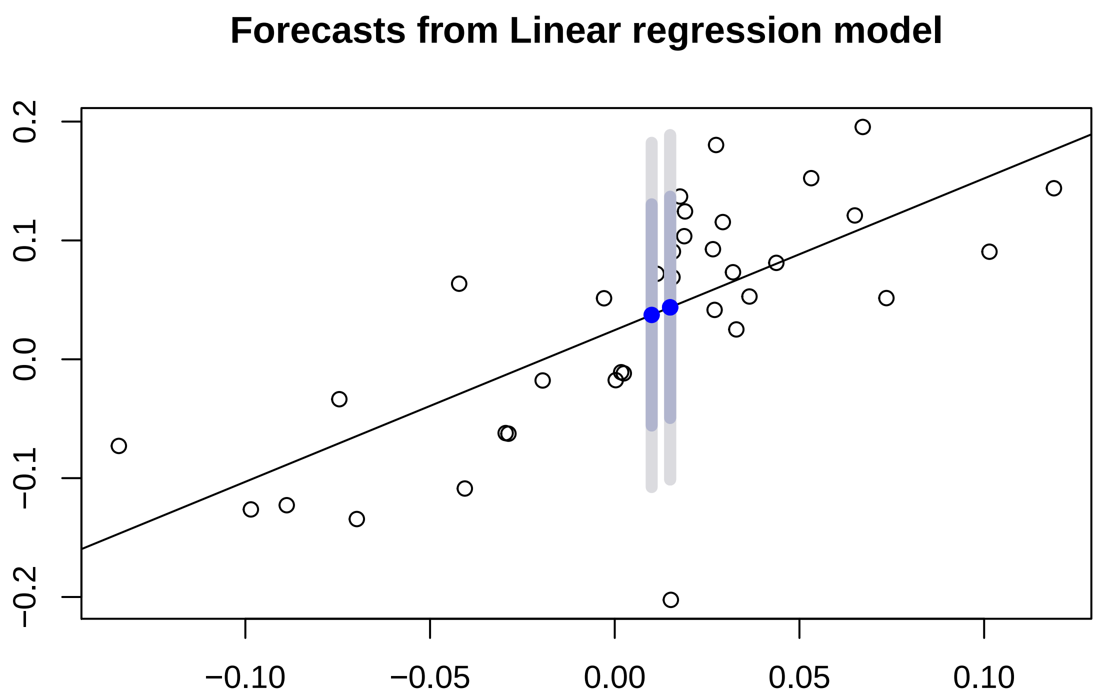
```
pred_exante <- forecast(reg_exante, newdata=data.frame(X_data=c(.010,.015,.015)))
pred_exante
```

```
##   Point Forecast       Lo 80     Hi 80      Lo 95     Hi 95
## 1     0.03731554 -0.05579452 0.1304256 -0.1075407 0.1821717
```

```
## 2      0.04369704 -0.04944022 0.1368343 -0.1012015 0.1885956
## 3      0.04369704 -0.04944022 0.1368343 -0.1012015 0.1885956
```

Graphing the regression and its forecast:

```r
plot(pred_exante, type='l')
```



Question 2

```r
# Get data for JPM, FB and the 10 year T-bill (TNX)
getSymbols("TSLA", src = "yahoo", from = start_date, to = end_date)
```

```
## [1] "TSLA"
```

```r
# Get adjusted returns data for 01/2014
rTSLA <- diff(log(to.monthly(TSLA)$TSLA.Adjusted))

# Calculate statistics
mean_rAAPL <- mean(rAAPL, na.rm = TRUE)
mean_rTSLA <- mean(rTSLA, na.rm = TRUE)
mean_rTNX <- mean(rTNX, na.rm = TRUE)
```

```r
sd_rAAPL <- sd(rAAPL, na.rm = TRUE)
sd_rTSLA <- sd(rTSLA, na.rm = TRUE)
```

a. Coefficient of variation

For AAPL:

```r
# Coefficient of variation of Adjusted Returns of AAPL
cv_rAAPL <- sd_rAAPL / mean_rAAPL
```

$$
\begin{aligned}
CV_{AAPL} =& \frac{\sigma}{\mu} \\
=& \frac{0.0981094}{0.0342826} \\
=& 2.8617831
\end{aligned} \tag{4}
$$
$$
\begin{aligned}
\text{where} & \\
\sigma =& \text{Standard deviation of Adjusted Returns} \\
\mu =& \text{Mean of Adjusted Returns}
\end{aligned}
$$

For TSLA:

```r
# Coefficient of variation of Adjusted Returns of TSLA
cv_rTSLA <- sd_rTSLA / mean_rTSLA
```

$$
\begin{aligned}
CV_{TSLA} =& \frac{\sigma}{\mu} \\
=& \frac{0.2064002}{0.0652246} \\
=& 3.1644534
\end{aligned} \tag{5}
$$
$$
\begin{aligned}
\text{where} & \\
\sigma =& \text{Standard deviation of Adjusted Returns} \\
\mu =& \text{Mean of Adjusted Returns}
\end{aligned}
$$

b. Sharpe Ratio

In this case, each portfolio carry 100% of AAPL and of TSLA so weighting is 100% of each stock for each portfolio.

For AAPL:

```r
# Sharpe Ratio of a 100% AAPL portfolio
sharpe_rAAPL <- (mean_rAAPL - mean_rTNX) / sd_rAAPL
```

$$
\begin{aligned}
Sharpe_{AAPL} =& \frac{E(r_p) - r_f}{\sigma_p} \\
=& \frac{0.0342826 - 0.0016079}{0.0981094} \\
=& 0.3330437 \\
\text{where} & \\
r_p =& \text{Mean of Adjusted Returns} \\
r_f =& \text{Risk free rate} \\
\sigma_p =& \text{Standard deviation of adjusted returns}
\end{aligned}
\tag{6}
$$

For TSLA:

```r
# Sharpe Ratio of a 100% TSLA portfolio
sharpe_rTSLA <- (mean_rTSLA - mean_rTNX) / sd_rTSLA
```

$$
\begin{aligned}
Sharpe_{TSLA} =& \frac{E(r_p) - r_f}{\sigma_p} \\
=& \frac{0.0652246 - 0.0016079}{0.2064002} \\
=& 0.3082202 \\
\text{where} & \\
r_p =& \text{Mean of Adjusted Returns} \\
r_f =& \text{Risk free rate} \\
\sigma_p =& \text{Standard deviation of adjusted returns}
\end{aligned}
\tag{7}
$$

c. Treynor Ratio

For AAPL:

```r
# Treynor Ratio of a 100% AAPL portfolio
beta_AAPL <- 1.28
treynor_rAAPL <- (mean_rAAPL - mean_rTNX) / beta_AAPL
```

$$
\begin{aligned}
Treynor_{AAPL} =& \frac{r_p - r_f}{\beta_p} \\
=& \frac{0.0342826 - 0.0016079}{1.28} = \quad 0.0255271
\end{aligned}
\tag{8}
$$

For TSLA:

```r
# Treynor Ratio of a 100% TSLA portfolio
beta_TSLA <- 2.19
treynor_rTSLA <- (mean_rTSLA - mean_rTNX) / beta_TSLA
```

$$
\begin{aligned}
Treynor_{AAPL} =& \frac{r_p - r_f}{\beta_p} \\
=& \frac{0.0652246 - 0.0016079}{2.19} = \quad 0.0290487
\end{aligned}
\tag{9}
$$

d. Sortino Ratio

For the purpose of calculating the Sortino ratio, we'll use the mean of Treasury bill daily risk-free rate for the period as the Minimum Acceptable Returns (MAR).

For AAPL:

```
# Calculating downside deviation using lower partial moment of order 2
# Assuming the 10 year T-bill returns as minimum acceptable returns MAR
mar <- mean_rTNX

# Deviation from MAR, this is a data frame, remove NA values
dev_rAAPL_mar <- na.omit(rAAPL - mar)

# Get the subset of negative values
devNegative_rAAPL_mar <- subset(dev_rAAPL_mar, dev_rAAPL_mar < 0)

# Calculate the Lower Partial Moment
downsideDev_AAPL <- var(devNegative_rAAPL_mar)

# Downside deviation
sd_downsideDev_AAPL <- sqrt(downsideDev_AAPL)

# Sortino Ratio
sortino_rAAPL <- (mean_rAAPL - mean_rTNX) / sd_downsideDev_AAPL
```

$$\begin{aligned} Sortino_{AAPL} &= \frac{r_p - r_f}{\sigma_{downside}} \\ &= \frac{0.0342826 - 0.0016079}{0.0596496} \\ &= 0.5477771 \\ \text{where} & \\ r_p &= \text{Mean of Adjusted Returns} \\ r_f &= \text{Risk free rate} \\ \sigma_{downside} &= \text{Standard deviation of the downside} \end{aligned} \tag{10}$$

For TSLA:

```
# Calculating downside deviation using lower partial moment of order 2
# Assuming the 10 year T-bill returns as minimum acceptable returns MAR
mar <- mean_rTNX

# Deviation from MAR, this is a data frame, remove NA values
dev_rTSLA_mar <- na.omit(rTSLA - mar)

# Get the subset of negative values
devNegative_rTSLA_mar <- subset(dev_rTSLA_mar, dev_rTSLA_mar < 0)

# Calculate the Lower Partial Moment
downsideDev_TSLA <- var(devNegative_rTSLA_mar)

# Downside deviation
```

```r
sd_downsideDev_TSLA <- sqrt(downsideDev_TSLA)

# Sortino Ratio
sortino_rTSLA <- (mean_rTSLA - mean_rTNX) / sd_downsideDev_TSLA
```

$$
\begin{aligned}
Sortino_{TSLA} =& \frac{r_p - r_f}{\sigma_{downside}} \\
=& \frac{0.0652246 - 0.0016079}{0.07707} \\
=& 0.8254408 \\
\text{where} & \\
r_p =& \text{Mean of Adjusted Returns} \\
r_f =& \text{Risk free rate} \\
\sigma_{downside} =& \text{Standard deviation of the downside}
\end{aligned} \tag{11}
$$

Remarks:

1. From the CV ratio, we can see that TSLA returns have larger variations than AAPL.
2. From the Sharpe ratio, we can see that AAPL generate better returns per unit of risk.
3. From the Treynor ratio, we can see that TSLA generate better EXCESS returns per unit of risk.
4. From the Sortino ratio, we can see that TSLA generate better risk-adjusted returns when considered downside risk.

Overall TSLA returns are more volatile but potentially generate better risk-adjusted returns and excess returns compare to AAPL.