assignment05

Jeff Nguyen

18/09/2020

**Regression Analysis on impact of COVID-19 on a 5-stock portfolio.**  
**Ngoc Son (Jeff) Nguyen**  
**University of Southern California**  
**Marshall School of Business**  
**FBE 506 Quantitative Method in Finance**  
**Fall 2020**  
**Directed by Professor Mohammad Safarzadeh**

# **Table of Contents**

# **Abstract**

# **Introduction**

We selected the following 5 securities to base our analysis of impact of COVID-19 on a CAPM model of 5 stocks upon.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ticker | Security | Sector | Industry | Founded | Full Time Employees |
| MSFT | Microsoft Corporation | Technology | Software-Infrastructure | 1975 | 163,000 |
| GWPH | GW Pharmaceuticals PLC | Healthcare | Drug Manufacturers-General | 1998 | 901 |
| DIS | The Walt Disney Company | Communication Services | Entertainment | 1923 | 223,000 |
| CAT | Caterpillar INC | Industrials | Farm & Heavy Construction Machinery | 1925 | 102,300 |
| AMZN | Amazon.com INC | Consumer Cyclical | Internet Retail | 1994 | 1,125,300 |

All information and data related to the securities are obtained from Yahoo Finance: MSFT, GWPH, DIS, CAT, and AMZN.

The objective of the study of the study is using the Modern Portfolio Theory to model a portfolio of five securities from different industries using adjusted closing price data from January 02, 2014 to December 31, 2018 to achieve the following:

       1) Estimated the CAPM for the five securities portfolio.  
       2) Understand the impact of COVID-19 on the alpha and market risk of the CAPM model.  
       3) Compare the MPT portfolio to a similarly diversified portfolio (State Street’s SPDR S&P 500 Trust ETF).  
       4) Use the CAPM model to forecast the returns on the portfolio.

# **Data Analysis**

## **1) Estimated the CAPM for the five securities portfolio**

### 1. Select at least five stocks from different industries (for the list of the firms in different industries see, <https://biz.yahoo.com/p/sum_conameu.html>).

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 <- "2014-01-01"  
end\_date <- "2018-12-31"  
  
# Get data  
getSymbols("MSFT", 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] "MSFT"

getSymbols("GWPH", src = "yahoo", , from = start\_date, to = end\_date)

## [1] "GWPH"

getSymbols("DIS", src = "yahoo", , from = start\_date, to = end\_date)

## [1] "DIS"

getSymbols("CAT", src = "yahoo", , from = start\_date, to = end\_date)

## [1] "CAT"

getSymbols("AMZN", src = "yahoo", , from = start\_date, to = end\_date)

## [1] "AMZN"

getSymbols("^GSPC", src = "yahoo", , from = start\_date, to = end\_date) # S&P 500

## [1] "^GSPC"

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"

# Adjusted Prices  
adjMSFT <- MSFT$MSFT.Adjusted  
adjGWPH <- GWPH$GWPH.Adjusted  
adjDIS <- DIS$DIS.Adjusted  
adjCAT <- CAT$CAT.Adjusted  
adjAMZN <- AMZN$AMZN.Adjusted  
  
# Get adjusted returns data  
rMSFT <- diff(log(to.monthly(MSFT)$MSFT.Adjusted))  
rGWPH <- diff(log(to.monthly(GWPH)$GWPH.Adjusted))  
rDIS <- diff(log(to.monthly(DIS)$DIS.Adjusted))  
rCAT <- diff(log(to.monthly(CAT)$CAT.Adjusted))  
rAMZN <- diff(log(to.monthly(AMZN)$AMZN.Adjusted))  
rGSPC <- diff(log(to.monthly(GSPC)$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 statistics  
MSFT\_return\_mean <- mean(rMSFT, na.rm = TRUE)  
GWPH\_return\_mean <- mean(rGWPH, na.rm = TRUE)  
DIS\_return\_mean <- mean(rDIS, na.rm = TRUE)  
CAT\_return\_mean <- mean(rCAT, na.rm = TRUE)  
AMZN\_return\_mean <- mean(rAMZN, na.rm = TRUE)  
GSPC\_return\_mean <- mean(rGSPC, na.rm = TRUE)  
TNX\_return\_mean <- mean(rTNX, na.rm = TRUE)  
  
MSFT\_return\_var <- var(rMSFT, na.rm = TRUE)  
GWPH\_return\_var <- var(rGWPH, na.rm = TRUE)  
DIS\_return\_var <- var(rDIS, na.rm = TRUE)  
CAT\_return\_var <- var(rCAT, na.rm = TRUE)  
AMZN\_return\_var <- var(rAMZN, na.rm = TRUE)  
GSPC\_return\_var <- var(rGSPC, na.rm = TRUE)  
  
# Excess Returns  
reMSFT <- rMSFT - rTNX  
reGWPH <- rGWPH - rTNX  
reDIS <- rDIS - rTNX  
reCAT <- rCAT - rTNX  
reAMZN <- rAMZN - rTNX  
  
# Information Tables:  
pricTabl <- data.frame(MSFT, GWPH, DIS, CAT, AMZN)  
  
# Creates data frame of asset prices  
retTabl <- data.frame(rMSFT, rGWPH, rDIS, rCAT, rAMZN)  
  
# Creates data frame of returns  
EretTabl <- data.frame(reMSFT, reGWPH, reDIS, reCAT, reAMZN)   
  
# Excess return data frame  
retTabl <- retTabl[-1,] # remove missing data due to lagging  
EretTabl <- EretTabl[-1,] # remove missing data due to lagging  
priceMat <- matrix(c(MSFT, GWPH, DIS, CAT, AMZN), nrow= length(MSFT), ncol=5, byrow=TRUE) # creates a matrix of prices  
  
# Variance/Covariance Matrix  
asset.names <- c("MSFT", "GWPH", "DIS", "CAT", "AMZN")  
  
# Create a list of row and col names for the var/cov matrix  
VCV <- matrix(c(cov(retTabl)), nrow=5, ncol = 5, byrow=TRUE) # create a var/cov matrix by finding cov of the assets in retTab2  
dimnames(VCV) <- list(asset.names, asset.names) # assigns asset.names to the VCV matrix  
  
#Calculate Returns  
rm <- matrix(colMeans(retTabl, na.rm=TRUE)) # creates an average return matrix, omitting missing values  
erm <- matrix(colMeans(EretTabl, na.rm=TRUE)) # creates an average excess return matrix, omitting missing values  
tnxy = mean((rTNX)[-1,]) # calculates the average bond yield excluding Jan (risk free rate)  
  
#Create Return Table  
retmat <- matrix(c(rm, erm), ncol=2)  
dimnames(retmat) = list(asset.names, c("Return ", "Excess Return"))

First we want to look at the data statistics

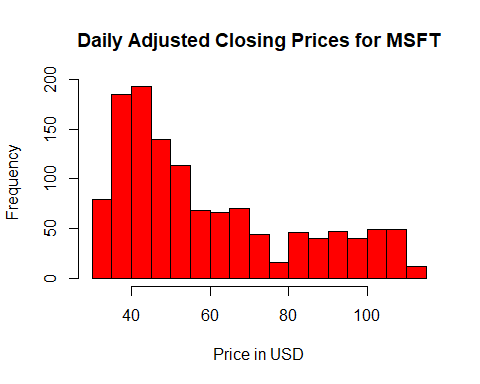
|  |  |  |  |
| --- | --- | --- | --- |
| Instruments | Mean Returns | Variance of Returns | Beta (5Y Monthly) |
| MSFT | 0.0185744 | 0.0035553 | .87 |
| GWPH | 0.0090021 | 0.0293563 | 1.96 |
| DIS | 0.0078206 | 0.0026173 | 1.08 |
| CAT | 0.0076711 | 0.0057854 | .98 |
| AMZN | 0.024 | 0.006961 | 1.3 |

Parameters of indices:

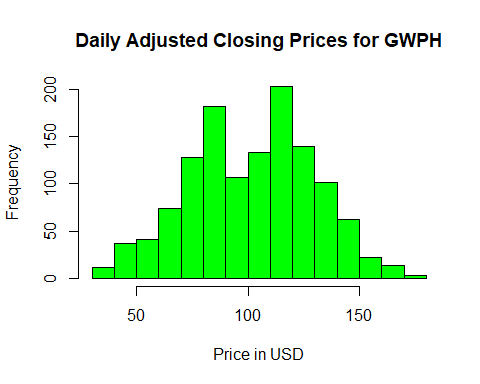
|  |  |  |  |
| --- | --- | --- | --- |
| Instruments | Mean Returns | Variance of Returns | Beta |
| S&P 500 | 0.0056356 | 0.0010169 | N/A |
| 10-Year T-bill | 0.0019378 | 0 | N/A |

We look at distribution of adjusted closing prices for each security:

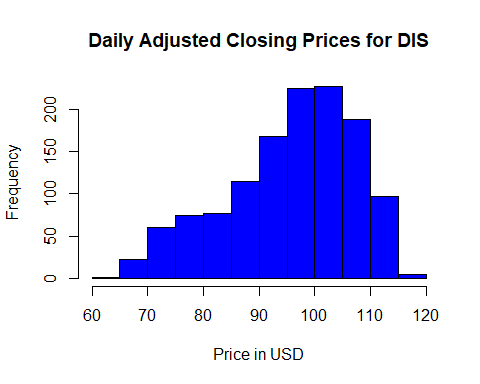
hist(adjMSFT,   
 main='Daily Adjusted Closing Prices for MSFT',   
 xlab='Price in USD',   
 col='red',  
 )



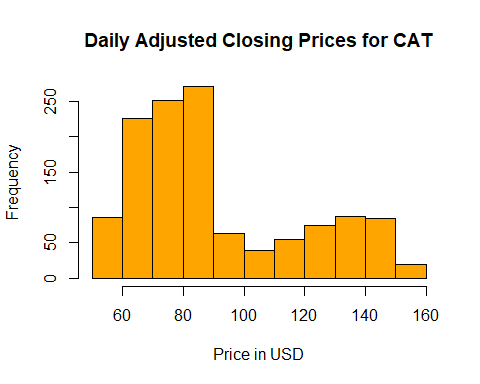
hist(adjGWPH,   
 main='Daily Adjusted Closing Prices for GWPH',   
 xlab='Price in USD',   
 col='green',  
 )



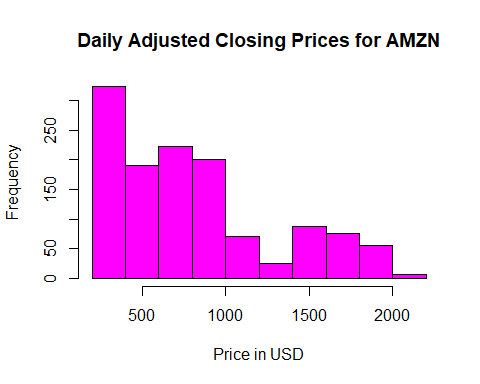
hist(adjDIS,   
 main='Daily Adjusted Closing Prices for DIS',   
 xlab='Price in USD',   
 col='blue',  
 )



hist(adjCAT,   
 main='Daily Adjusted Closing Prices for CAT',   
 xlab='Price in USD',   
 col='orange',  
 )



hist(adjAMZN,   
 main='Daily Adjusted Closing Prices for AMZN',   
 xlab='Price in USD',   
 col='magenta',  
 )



## CAPM Portfolio Construction

Question covered: 2,3,4,5  
**Methodology**  
       2a) Find the optimum weights using MPT.  
       2b) Allocate $100.00 among the selected stocks using adjusted closing prices at 2018M12. 2019M1 will have a value of 100 as an index.  
       2c) Using the adjusted closing prices from 2018M12 to 2020M8 calculate the holding values of the portfolio (assume fixed holdings with no re-balancing taking place over time).  
       3) Find the tangency point of the Capital Allocation Line (CAL) and the efficient frontier.  
       4) Calculate the CAL equation and graph CAL and the efficient frontier.  
       5) Estimate CAPM for your portfolio and graph the estimated of the CAPM and the average return of your portfolio as a point relative to SML.

### 2a) Find the optimum weights using MPT

Since the investor’s objective is to minimize risk subjected to a minimum return of the risk free asset–US Treasury Bill, in this case–we solve the constrained optimization problem.  
Let denotes the weight of the investment in asset i , and assume all money is invested in i, meaning .

Formulating the Markowitz portfolio problem:

Let denotes a target expected return level. Formulate the problem:

To solve this, form the Lagrangian function:

Because there are two constraints ( and ) there are two Langrange multipliers and . The first order condition for a minimum are the linear equations:

Simplify, we have:

Rewrite in matrix form:

or

The solution for is:

The variance-covariance matrix is as follow:

VCV

## MSFT GWPH DIS CAT AMZN  
## MSFT 0.003555316 0.001599836 0.001128235 0.002042991 0.002538589  
## GWPH 0.001599836 0.029356292 0.002528779 0.005980634 0.005396791  
## DIS 0.001128235 0.002528779 0.002617304 0.001243413 0.001405934  
## CAT 0.002042991 0.005980634 0.001243413 0.005785368 0.002040476  
## AMZN 0.002538589 0.005396791 0.001405934 0.002040476 0.006961024

The monthly risk-free rate is:

# Optimum Portfolio  
ZOPT <- solve(VCV,erm) # multiply inverse of VCV to excess return to find z  
WOPT <- ZOPT/sum(ZOPT) # calculates weights  
dimnames(WOPT) <- list(asset.names, "Weights") #label the weight matrix  
  
# Calculate stats  
ROPT <- t(WOPT)%\*%rm # calculate optimal portfolio's return  
VOPT <- t(WOPT)%\*%VCV%\*%WOPT # calculate optimal portfolio's variance  
SDOPT <- VOPT^0.5 # calculate optimal portfolio's std dev  
SRatio <-(ROPT-tnxy)/(SDOPT) # calculate optimal portfolio's Sharpe ratio  
  
# Create Optimal Stats Table  
PTBL <- matrix(c(ROPT, VOPT, SDOPT, SRatio), nrow = 4) # create a matrix of return, variance, std dev, Sharpe  
optstat.names <- c("Return", "Variance", "Std Dev", "Sharpe") # labels for PTBL matrix  
  
dimnames(PTBL) <- list(optstat.names, "Opt. Portfolio") # label the optimal portfolio matrix values

The optimal portfolio weights are as follow:

WOPT

## Weights  
## MSFT 0.75172435  
## GWPH -0.03902504  
## DIS 0.01785490  
## CAT -0.18682397  
## AMZN 0.45626977

The statistics of the optimal portfolio is:

PTBL

## Opt. Portfolio  
## Return 0.02326846  
## Variance 0.00436794  
## Std Dev 0.06609039  
## Sharpe 0.32282226

### 2b) Allocate $100.00 among the selected stocks using adjusted closing prices at 2018M12. 2019M1 will have a value of 100 as an index.

# Set start date and end date of data  
start\_date1 <- "2018-12-01"  
end\_date1 <- "2020-08-31"  
  
# Get data  
getSymbols("MSFT", src = "yahoo", from = start\_date1, to = end\_date1)

## [1] "MSFT"

getSymbols("GWPH", src = "yahoo", , from = start\_date1, to = end\_date1)

## [1] "GWPH"

getSymbols("DIS", src = "yahoo", , from = start\_date1, to = end\_date1)

## [1] "DIS"

getSymbols("CAT", src = "yahoo", , from = start\_date1, to = end\_date1)

## [1] "CAT"

getSymbols("AMZN", src = "yahoo", , from = start\_date1, to = end\_date1)

## [1] "AMZN"

getSymbols("^GSPC", src = "yahoo", , from = start\_date1, to = end\_date1) # S&P 500

## [1] "^GSPC"

getSymbols("^TNX", src = "yahoo", from=start\_date1, to=end\_date1) # 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"

rMSFT1 <- diff(log(to.monthly(MSFT)$MSFT.Adjusted))  
rGWPH1 <- diff(log(to.monthly(GWPH)$GWPH.Adjusted))  
rDIS1 <- diff(log(to.monthly(DIS)$DIS.Adjusted))  
rCAT1 <- diff(log(to.monthly(CAT)$CAT.Adjusted))  
rAMZN1 <- diff(log(to.monthly(AMZN)$AMZN.Adjusted))  
rGSPC1 <- diff(log(to.monthly(GSPC)$GSPC.Adjusted))  
rTNX1 <- to.monthly(TNX)$TNX.Adjusted /1200 # Using monthly rate

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

rTNX1 <- rTNX1[-1,] # remove missing data due to lagging  
mean\_rTNX1 <- mean(rTNX1, na.rm=TRUE)  
  
# Adjusted Prices  
adjMSFT1 <- MSFT$MSFT.Adjusted  
adjGWPH1 <- GWPH$GWPH.Adjusted  
adjDIS1 <- DIS$DIS.Adjusted  
adjCAT1 <- CAT$CAT.Adjusted  
adjAMZN1 <- AMZN$AMZN.Adjusted  
  
investedAmount <- 100  
  
sharesMSFT <- as.numeric(investedAmount \* WOPT[1] / adjMSFT1[1])  
sharesGWPH <- as.numeric(investedAmount \* WOPT[2] / adjGWPH1[1])  
sharesDIS <- as.numeric(investedAmount \* WOPT[3] / adjDIS1[1])  
sharesCAT <- as.numeric(investedAmount \* WOPT[4] / adjCAT1[1])  
sharesAMZN <- as.numeric(investedAmount \* WOPT[5] / adjAMZN1[1])  
  
holdings <- data.frame("Holding Value"=sharesMSFT\*adjMSFT1 +   
 sharesGWPH\*adjGWPH1 +  
 sharesDIS\*adjDIS1 +   
 sharesCAT\*adjCAT1 +  
 sharesAMZN\*adjAMZN1)  
names(holdings)[1] <- "Port. Holdings Val" # rename column

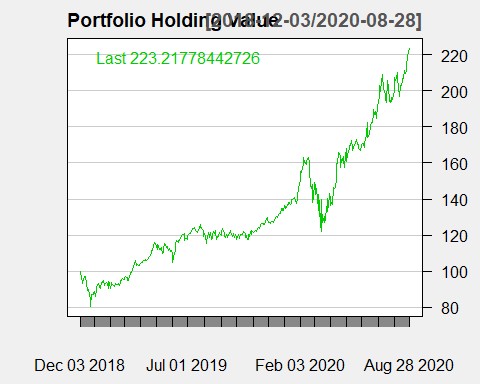
Based on the optimal weighting, to allocate $100 to the portfolio, we would be purchase the following amount of each security:

|  |  |  |
| --- | --- | --- |
| Ticker | Weights | Stock to purchase |
| MSFT | 0.7517244 | 0.6859734 |
| GWPH | -0.039025 | -0.0309109 |
| DIS | 0.0178549 | 0.0157364 |
| CAT | -0.186824 | -0.1424304 |
| AMZN | 0.4562698 | 0.0257436 |

### 2c) Using the adjusted closing prices from 2018M12 to 2020M8 calculate the holding values of the portfolio (assume fixed holdings with no re-balancing taking place over time).

We can then observe the fluctuations in the holding value of the portfolio from the period starting December 01 2018 to August 31, 2020 as follow.

chartSeries(holdings, name="Portfolio Holding Value", type="line", theme=chartTheme("white"))



By inspection we can see the portfolio experience a sharp sell off of almost 20% in December 2018, coincide with the broad U.S.market selloff due to a combination of the FED hiking the federal funds rate by 25 basis points to a targeted range of 2.25% to 2.5% (JeffCoxCNBCcom) and corporations followed suit by cutting profit forecasts and try temper expectations for earnings growth in 2019 after a big 2018 (Moyer).

The second visibly sharp sell off of the portfolio holding value also coincides with the broad market sell off in the mid March 2020 with investors raising cash in a risk-on environment when COVID-19 lockdowns started going into effects in the U.S.

### 3) Find the tangency point of the Capital Allocation Line (CAL) and the efficient frontier.

The tangency point of the Capital Allocation Line is the point where the weights of the portfolio is optimal, represented by the point which is .

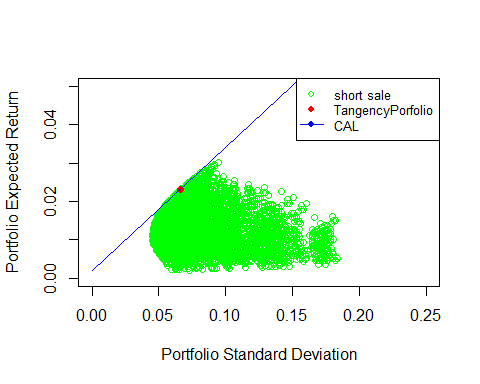
### 4) Calculate the CAL equation and graph CAL and the efficient frontier.

The efficient frontier is the portfolio possibility curve represented by the equation:

# Efficient Frontier and CAL  
j <- 0 # set value for iterative loop variable t  
return\_p <- rep(0, 50000)  
sd\_p <- rep(0, 50000)  
  
# create a matrix of 0 to fill later with sd of different weights  
vect\_0 <- rep(0, 50000)  
  
# create a matrix of 0  
fractions <- matrix(vect\_0, 10000, 5)  
  
# create a matrix of 0 to fill with weights  
# iterate through weights for asset 1-5 from -20% to 100% by 10%  
for (a in seq(-.2, 1, 0.1))   
 {  
 for (b in seq(-.2, 1, 0.1))  
 {  
 for (c in seq(-.2, 1, 0.1))  
 {  
 for (d in seq(-.2, 1, 0.1))  
 {  
 for (e in seq(-.2, 1, 0.1))  
 {  
 #test that the weights are equal to 1  
 if (a+b+c+d+e==1)   
 {  
 # increment j by 1 if a+b+c+d+e is equal to 1 (valid weights)  
 j=j+1  
 # load a,b,c,d,e values into row j of the matrix  
 fractions[j,] <- c(a,b,c,d,e)  
 # calculate the std dev of the portfolio at a given weight of assets  
 sd\_p[j] <- (t(fractions[j,])%\*%VCV%\*%fractions[j,])^.5  
 # calculate the return of the portfolio at a given weight of assets  
 return\_p[j] <- fractions[j,]%\*%rm  
 }  
 }  
 }  
 }  
 }  
 }  
# assign filled vector spots in return\_p to the R\_p matrix to omit empty spots  
Rport <- return\_p[1:j]  
  
# assign filled vector spots in sd\_p to the sigma\_p matrix to omit empty spots  
StdDev\_p <- sd\_p[1:j]  
  
# Create Capital Asset Line  
# Create x-coordinates for CAL points  
f <- seq(0,.24, .24)  
  
# Calculate corresponding y-coordinates  
CAL <- tnxy + SRatio \* f

## Warning in SRatio \* f: Recycling array of length 1 in array-vector arithmetic is deprecated.  
## Use c() or as.vector() instead.

#Plot the portfolio possibilities curve:  
plot(StdDev\_p, Rport, col="green1", xlab="Portfolio Standard Deviation", ylab= "Portfolio Expected Return", xlim=c(0, .25), ylim= c(0, .05))  
  
#Plot of tangency point in red  
points(SDOPT, ROPT, col= "red", pch=16, bg="red")  
  
#Plot of CAL in blue  
points(f, CAL, col= "blue", type="l")  
  
legend("topright",c("short sale", "TangencyPorfolio", "CAL"), cex=.8, col=c("green1", "red","blue"),   
 lty =c(0,0,1),pch=c(1,16,16))



### 5) Estimate CAPM for your portfolio and graph the estimated of the CAPM and the average return of your portfolio as a point relative to SML.

The expected risk premium of the portfolio based on the CAPM model is given as:

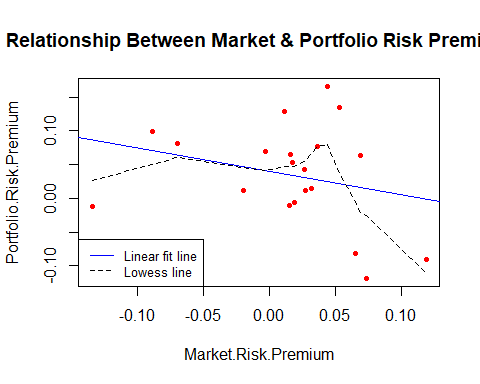
Here, the risk premium of the S&P 500 is the independent variable and the expected risk premium of the portfolio is the dependent variable.

Hypothesis for regression:

# Calculate and normalized the CAPM holdings  
ra <- diff(log(to.monthly(holdings)[,1]))  
  
Y <- na.omit(ra - rTNX1)  
names(Y)[1] <- "Portfolio Risk Premium" # Rename column  
Y\_bar <- mean(Y)  
Y\_bar

## [1] 0.03515135

X <- na.omit(rGSPC1 - rTNX1)  
mean\_X <- mean(X)  
  
names(X)[1] <- "Market Risk Premium" # Rename column  
data1 <- data.frame(X, Y)  
  
plot(data1, col='red', main="Relationship Between Market & Portfolio Risk Premium", pch=20, cex=1)  
  
# Add fit lines  
abline(lm(Y~X), col="blue") # Regression line Y ~ X  
lines(lowess(X,Y), 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)



Through inspection, we observe the cluster observation scattering in a big range from left to right. This implies a weak linear relationship between the Market Portfolio Risk Premium (the independent variable on the x-axis) and the CAPM Portfolio Risk Premium (the dependent variable on the y-axis).

Next, we attempts to fit an equation of a line:

fit1 <- lm(Y~X, data=data1)  
summary(fit1)

##   
## Call:  
## lm(formula = Y ~ X, data = data1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.13365 -0.04123 0.01465 0.03414 0.14071   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.04050 0.01728 2.343 0.0308 \*  
## X -0.34504 0.29305 -1.177 0.2544   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07458 on 18 degrees of freedom  
## Multiple R-squared: 0.07151, Adjusted R-squared: 0.01992   
## F-statistic: 1.386 on 1 and 18 DF, p-value: 0.2544

The estimated equation is , where the for the intercept .  
Therefore, we reject the null hypothesis at 95% confidence level that the intercept statistically is no different from zero. Thus, we reject the null hypothesis and accept the null hypothesis .  
The coefficient represents the increase in portfolio risk premium relative to increase in the market portfolio risk premium. The for is , implying that the coefficient statistically is insignificant at 95% or more, and we accept the null hypothesis and reject the alternative hypothesis .

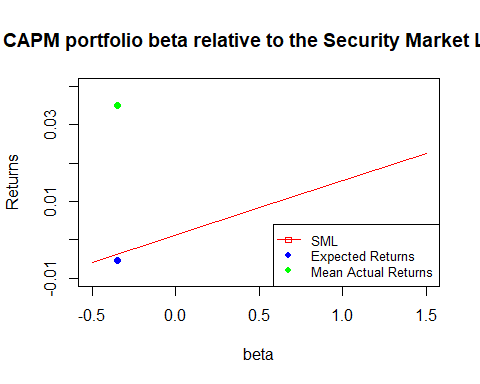
Goodness of Fit:  
Through inspection, we observe the value to not be close to 1 at all. implies that of the variations in the portfolio risk premium is explained by the market risk premium.

Standard Error of Regression:  
We can see that the Standard Error of Regression is .  
From this, we can calculate the forecasting efficiency statistic to be:  
 This statistic implies that this 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.

The Security Market Line:

# Generate the SML equation  
slope\_SML <- (mean\_X - mean\_rTNX1) / (1-0)  
#slope\_SML  
SML <- function(beta) mean\_rTNX1 + slope\_SML \* beta  
  
# Plot the SML  
beta <- seq(-.5, 1.5)  
plot(beta, SML(beta), col="red", type="l", main="CAPM portfolio beta relative to the Security Market Line", xlab="beta", ylab="Returns", ylim=c(-.01, .04))  
  
# Plot the expected returns  
points(-.34504, -.34504\*mean\_X, col="blue", pch=16)  
  
# Plot the average returns  
points(-.34504, Y\_bar, col="green", pch=16)  
  
legend("bottomright",c("SML", "Expected Returns", "Mean Actual Returns"), cex=.8,   
 col=c("red", "blue", "green"), lty=c(1,0,0), pch=c(0,16,16))



The Security Market Line pass through the point and , which are and .  
Relative to its market risk of , the expected return is and the average return is . We can observe that at this estimated , the expected return is below the security market line and the actual average return is above the security market line.

## 6) Understand the impact of COVID-19 on the alpha and market risk of the CAPM model.

Question covered: 6).

**Methodology** Test whether the closing of the economy due to COVID-19 had any effect on Jensen alpha and the market risk of the CAPM model. (6)

We run regression for the model to see if the pandemic has any effect on Jensen alpha:

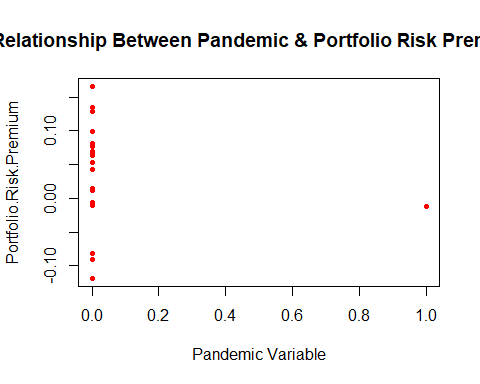
Hypothesis for regression:

We examine the potential relationship between the pandemic variable and the portfolio risk premium by plotting a scatter plot as follow:

# Generating the dummy variable  
D <- rep(0, 20)  
D[15] <-1  
D

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

data2 <- data.frame(D, Y)  
names(data2)[1] <- "Pandemic Variable"  
  
plot(data2, col='red', main="Relationship Between Pandemic & Portfolio Risk Premium", pch=20, cex=1)



Through inspection, we can see there is not much of a linear relationship between these variables.

Let’s assume linearity, we run regression on as hypothesized above.

fit2 <- lm(Y~X+D)  
summary(fit2)

##   
## Call:  
## lm(formula = Y ~ X + D)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.119935 -0.048422 -0.008044 0.028370 0.143203   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.05447 0.01806 3.017 0.00778 \*\*  
## X -0.72150 0.34667 -2.081 0.05284 .   
## D -0.16285 0.09052 -1.799 0.08979 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07034 on 17 degrees of freedom  
## Multiple R-squared: 0.22, Adjusted R-squared: 0.1282   
## F-statistic: 2.398 on 2 and 17 DF, p-value: 0.121

The estimated equation is , where the for the intercept .  
Therefore, we reject the null hypothesis at 95% confidence level that the intercept statistically is no different from zero. Thus, we reject the null hypothesis and accept the null hypothesis .  
In addition, the coefficient represents the increase in portfolio risk premium relative to increase in the pandemic variable. The for is , implying that the coefficient statistically is insignificant at 95% or more, and we accept the null hypothesis and reject the alternative hypothesis .

Goodness of Fit:  
Through inspection, we observe the value to not be close to 1 at all. implies that of the variations in the portfolio risk premium is explained by the pandemic variable.

Standard Error of Regression:  
We can see that the Standard Error of Regression is .  
From this, we can calculate the forecasting efficiency statistic to be:  
 This statistic implies that this is not a good forecasting model.

Now, we want to explore the affect of the pandemic lockdowns on market risk. Thus, we want to explore the following regression model:

Hypothesis for regression:

# Generate the new variable dx = D\*X  
DX <- D\*X  
fit3 <- lm(Y~X+D+DX)  
summary(fit3)

##   
## Call:  
## lm(formula = Y ~ X + D + DX)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.119935 -0.048422 -0.008044 0.028370 0.143203   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.05447 0.01806 3.017 0.00778 \*\*  
## X -0.72150 0.34667 -2.081 0.05284 .   
## D -0.16285 0.09052 -1.799 0.08979 .   
## DX NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07034 on 17 degrees of freedom  
## Multiple R-squared: 0.22, Adjusted R-squared: 0.1282   
## F-statistic: 2.398 on 2 and 17 DF, p-value: 0.121

The estimated equation is

We see implying that the slope dummy is an unnecessary predictor. The coefficient is not estimated.

Let’s run the regression with the slope dummy separately, so the equation of the regression becomes:

Again, stating the hypothesis for regression:

# Generate the new variable dx = D\*X  
fit4 <- lm(Y~X+DX)  
summary(fit4)

##   
## Call:  
## lm(formula = Y ~ X + DX)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.119935 -0.048422 -0.008044 0.028370 0.143203   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.05447 0.01806 3.017 0.00778 \*\*  
## X -0.72150 0.34667 -2.081 0.05284 .   
## DX 1.21306 0.67427 1.799 0.08979 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07034 on 17 degrees of freedom  
## Multiple R-squared: 0.22, Adjusted R-squared: 0.1282   
## F-statistic: 2.398 on 2 and 17 DF, p-value: 0.121

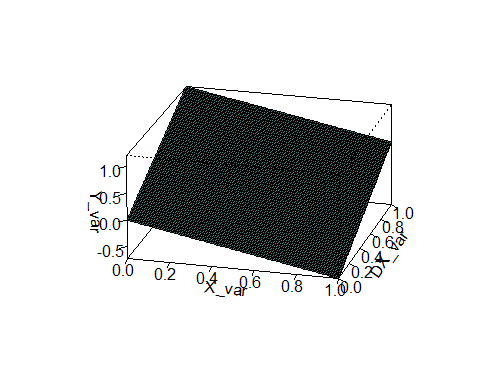
The estimated equation is , where the for the intercept .  
Therefore, we reject the null hypothesis at 95% confidence level that the intercept statistically is no different from zero. Thus, we reject the null hypothesis and accept the null hypothesis .  
In addition, the coefficient represents the increase in market risk premium relative to increase in the pandemic variable. The for is , implying that the coefficient statistically is insignificant at 95% or more, and we accept the null hypothesis and reject the alternative hypothesis .

Goodness of Fit:  
Through inspection, we observe the value to not be close to 1 at all. implies that of the variations in the portfolio risk premium is explained by the pandemic variable.

Standard Error of Regression:  
We can see that the Standard Error of Regression is .  
From this, we can calculate the forecasting efficiency statistic to be:  
 This statistic implies that this is not a good forecasting model.

To visualize:

X\_var <- seq(0,1,.01)  
DX\_var <- X\_var  
f <- function(X, DX) {fit4$coefficient[2]\*X + fit4$coefficient[3]\*DX}  
Y\_var <- outer(X\_var, DX\_var, f)  
persp(X\_var, DX\_var, Y\_var, col="cadet blue",   
 theta=15, phi=20, r=50, d=0.1, expand=0.5, ltheta=90, lphi=20, shade=0.75, ticktype="detailed", nticks=5)



## 3) Compare the CAPM portfolio to a diversified State Street’s SPDR S&P 500 Trust ETF portfolio (7,8,9)

Questions covered: 7,8,9.

**Methodology**  
       7) Calculate CV, Sharpe, Treynor, and Sortino ratios for your portfolio and compare them to a similarly diversified portfolio of Vanguard, Fidelity, or any other similar portfolio.  
       8) Calculate 2% and 3% VaRs as a percentage of the mean return of your portfolio when the risk horizon is one year, six months, and one month. Calculate the same VaRs for the selected portfolio in item 6. Compare the VaRs of your portfolio to the ones of the market portfolios.  
       9) Graph the scatter diagram of your portfolio and comment on trends, outliers, structural breaks and any other special features. Graph the scatter diagram of your portfolio and S&P 500 on the same coordinate system and compare the trends.

### 7) Calculate CV, Sharpe, Treynor, and Sortino ratios for your portfolio and compare them to a similarly diversified portfolio of Vanguard, Fidelity, or any other similar portfolio.

# Calculate statistics  
MSFT1\_return\_mean <- mean(rMSFT1, na.rm = TRUE)  
GWPH1\_return\_mean <- mean(rGWPH1, na.rm = TRUE)  
DIS1\_return\_mean <- mean(rDIS1, na.rm = TRUE)  
CAT1\_return\_mean <- mean(rCAT1, na.rm = TRUE)  
AMZN1\_return\_mean <- mean(rAMZN1, na.rm = TRUE)  
GSPC1\_return\_mean <- mean(rGSPC1, na.rm = TRUE)  
TNX1\_return\_mean <- mean(rTNX1, na.rm = TRUE)  
  
MSFT1\_return\_var <- var(rMSFT1, na.rm = TRUE)  
GWPH1\_return\_var <- var(rGWPH1, na.rm = TRUE)  
DIS1\_return\_var <- var(rDIS1, na.rm = TRUE)  
CAT1\_return\_var <- var(rCAT1, na.rm = TRUE)  
AMZN1\_return\_var <- var(rAMZN1, na.rm = TRUE)  
GSPC1\_return\_var <- var(rGSPC1, na.rm = TRUE)  
  
# Excess Returns  
reMSFT1 <- rMSFT1 - rTNX1  
reGWPH1 <- rGWPH1 - rTNX1  
reDIS1 <- rDIS1 - rTNX1  
reCAT1 <- rCAT1 - rTNX1  
reAMZN1 <- rAMZN1 - rTNX1  
  
# Information Tables:  
#pricTabl1 <- data.frame(MSFT, GWPH, DIS, CAT, AMZN)  
  
# Creates data frame of asset prices  
retTabl1 <- data.frame(rMSFT1, rGWPH1, rDIS1, rCAT1, rAMZN1)  
  
# Creates data frame of returns  
EretTabl1 <- data.frame(reMSFT1, reGWPH1, reDIS1, reCAT1, reAMZN1)   
  
# Excess return data frame  
retTabl1 <- retTabl1[-1,] # remove missing data due to lagging  
EretTabl1 <- EretTabl1[-1,] # remove missing data due to lagging  
priceMat1 <- matrix(c(MSFT, GWPH, DIS, CAT, AMZN), nrow= length(MSFT), ncol=5, byrow=TRUE) # creates a matrix of prices  
  
# Variance/Covariance Matrix  
asset.names <- c("MSFT", "GWPH", "DIS", "CAT", "AMZN")  
  
# Create a list of row and col names for the var/cov matrix  
VCV1 <- matrix(c(cov(retTabl1)), nrow=5, ncol = 5, byrow=TRUE) # create a var/cov matrix by finding cov of the assets in retTab2  
dimnames(VCV1) <- list(asset.names, asset.names) # assigns asset.names to the VCV matrix  
  
#Calculate Returns  
rm1 <- matrix(colMeans(retTabl1, na.rm=TRUE)) # creates an average return matrix, omitting missing values  
erm1 <- matrix(colMeans(EretTabl1, na.rm=TRUE)) # creates an average excess return matrix, omitting missing values  
tnxy1 = mean((rTNX1)[-1,]) # calculates the average bond yield excluding Jan (risk free rate)  
  
#Create Return Table  
retmat1 <- matrix(c(rm1, erm1), ncol=2)  
dimnames(retmat1) = list(asset.names, c("Return ", "Excess Return"))  
  
# Calculate stats  
ROPT1 <- t(WOPT)%\*%rm1 # calculate optimal portfolio's return  
VOPT1 <- t(WOPT)%\*%VCV1%\*%WOPT # calculate optimal portfolio's variance  
SDOPT1 <- VOPT1^0.5 # calculate optimal portfolio's std dev  
SRatio1 <-(ROPT1-tnxy1)/(SDOPT1) # calculate optimal portfolio's Sharpe ratio  
  
mean\_Holdings1 <- as.numeric(colMeans(holdings[1])) # Mean   
sd\_Holdings1 <- as.numeric(sqrt(var(holdings)))  
CV1 <- sd\_Holdings1 / mean\_Holdings1 # calculate portfolio's coefficient of variation  
  
TrRatio1 <- (ROPT1 - tnxy1)/ -0.34504 # calculate portfolio's Treynor Ratio  
  
# Calculating downside deviation using lower partial moment of order 2  
mar <- tnxy1 # Assuming the 10 year T-bill returns as minimum acceptable returns MAR  
rHoldings <- diff(log(holdings[,1])) # calculate portfolio holding returns data  
dev\_rHoldings\_mar <- na.omit(rHoldings - mar) # Deviation from MAR, this is a data frame, remove NA values  
devNegative\_rHoldings\_mar <- subset(dev\_rHoldings\_mar, dev\_rHoldings\_mar < 0) # Get the subset of negative values  
downsideDev\_Holdings <- var(devNegative\_rHoldings\_mar) # Calculate the Lower Partial Moment  
sd\_downsideDev\_Holdings <- sqrt(downsideDev\_Holdings) # Downside deviation  
  
SoRatio1 <- (ROPT1 - tnxy1) / sd\_downsideDev\_Holdings # calculate portfolio Sortino Ratio  
  
# Create Optimal Stats Table  
PTBL1 <- matrix(c(CV1, SRatio1, TrRatio1, SoRatio1), nrow = 4) # create a matrix of return, variance, std dev, Sharpe  
optstat.names <- c("CV", "Sharpe", "Treynor", "Sortino") # labels for PTBL matrix  
  
dimnames(PTBL1) <- list(optstat.names, "Holdings") # label the optimal portfolio matrix values

# Get data  
getSymbols("SPY", src = "yahoo", from = start\_date1, to = end\_date1)

## [1] "SPY"

rSPY1 <- na.omit(diff(log(to.monthly(SPY)$SPY.Adjusted)))  
mean\_rSPY1 <- mean(rSPY1)  
var\_rSPY1 <- var(rSPY1)  
sd\_rSPY1 <- sqrt(var\_rSPY1)  
  
adjSPY1 <- SPY$SPY.Adjusted  
mean\_adjSPY1 <- mean(adjSPY1)  
var\_adjSPY1 <- var(adjSPY1)  
sd\_adjSPY1 <- sqrt(var\_adjSPY1)  
  
SRatioSPY1 <-(mean\_rSPY1-tnxy1) / sd\_rSPY1 # calculate optimal portfolio's Sharpe ratio  
CVSPY1 <- sd\_adjSPY1 / mean\_adjSPY1 # calculate portfolio's coefficient of variation  
TrRatioSPY1 <- (mean\_rSPY1 - tnxy1)/ 1 # calculate portfolio's Treynor Ratio  
  
# Calculating downside deviation using lower partial moment of order 2  
marSPY1 <- tnxy1 # Assuming the 10 year T-bill returns as minimum acceptable returns MAR  
dev\_rSPY1\_mar <- na.omit(rSPY1 - mar) # Deviation from MAR, this is a data frame, remove NA values  
devNegative\_rSPY1\_mar <- subset(dev\_rSPY1\_mar, dev\_rSPY1\_mar < 0) # Get the subset of negative values  
downsideDev\_SPY1 <- var(devNegative\_rSPY1\_mar) # Calculate the Lower Partial Moment  
sd\_downsideDev\_SPY1 <- sqrt(downsideDev\_SPY1) # Downside deviation  
  
SoRatioSPY1 <- (mean\_rSPY1 - tnxy1) / sd\_downsideDev\_SPY1 # calculate portfolio Sortino Ratio  
  
# Create Optimal Stats Table  
SPYTBL1 <- matrix(c(CVSPY1, SRatioSPY1, TrRatioSPY1, SoRatioSPY1), nrow = 4) # create a matrix of return, variance, std dev, Sharpe  
optstat.names <- c("CV", "Sharpe", "Treynor", "Sortino") # labels for PTBL matrix  
  
dimnames(SPYTBL1) <- list(optstat.names, "SPY") # label the optimal portfolio matrix values  
  
SPY\_PTBL1 <- data.frame(PTBL1, SPYTBL1)

Compare the key financial ratio of the portfolio to that of the State Street’s SPDR S&P 500 Trust ETF portfolio (SPY), we see:

SPY\_PTBL1

## Holdings SPY  
## CV 0.2475044 0.08583704  
## Sharpe 0.7609424 0.29444966  
## Treynor -0.1369133 0.01710602  
## Sortino 2.8067889 0.32133508

**Remarks:** **Coefficient of Variations:** The SPY has extremely low dispersion of value (based on Adjusted Closing Price) compared to that of the holding portfolio. This implies the holding portfolio carry significantly more volatility in comparison to the SPY.  
**Sharpe Ratio:** The holding portfolio has higher Sharpe ratio compare to the SPY. Assuming returns are normally distributed, this implies the holding portfolio generates significantly higher returns than the SPY.  
**Treynor Ratio:** The SPY has a higher and positive Treynor Ratio compared to a lower and negative Treynor Ratio. This implies that the SPY is much better at generating excess returns per unit of risk that it takes on. Compared to the holding portfolio, the SPY is a more suitable investment.  
**Sortino Ratio:** The holding portfolio has significantly higher Sortino Ratio compared to the SPY. This implies that it is earning returns more efficiently than the SPY given downside deviations: earning more turns per unit of the bad risk it is taking on.

### 8) Calculate 2% and 3% VaRs as a percentage of the mean return of your portfolio when the risk horizon is one year, six months, and one month. Compare the VaRs of your portfolio to the ones of the market portfolios.

# 2% VaR Holding Portfolio  
VaR\_threshold <- 2  
  
# Set scaling factor based on the period of evaluation  
# In this example: {1: '1 year', 1/2: '6 months', 1/12: '1 month', 1/250: '1 day'}  
  
# 2% 1 year VAR  
scalingFactor <- 1  
z\_stat <- qnorm(VaR\_threshold/100, 0, 1, lower.tail = TRUE)  
VaR\_1year\_p\_2percent <- ROPT1\*scalingFactor + SDOPT1\*sqrt(scalingFactor) \* z\_stat  
VaR\_1year\_SPY\_2percent <- mean\_rSPY1\*scalingFactor + sd\_rSPY1\*sqrt(scalingFactor) \* z\_stat  
  
# 2% 6 month VAR  
scalingFactor <- 1/2  
VaR\_6month\_p\_2percent <- ROPT1\*scalingFactor + SDOPT1\*sqrt(scalingFactor) \* z\_stat  
VaR\_6month\_SPY\_2percent <- mean\_rSPY1\*scalingFactor + sd\_rSPY1\*sqrt(scalingFactor) \* z\_stat  
  
# 2% 1 month VAR  
scalingFactor <- 1/12  
VaR\_1month\_p\_2percent <- ROPT1\*scalingFactor + SDOPT1\*sqrt(scalingFactor) \* z\_stat  
VaR\_1month\_SPY\_2percent <- mean\_rSPY1\*scalingFactor + sd\_rSPY1\*sqrt(scalingFactor) \* z\_stat  
#########################################  
  
# 3% VaR Holding Portfolio  
VaR\_threshold <- 3  
  
# Set scaling factor based on the period of evaluation  
# In this example: {1: '1 year', 1/2: '6 months', 1/12: '1 month', 1/250: '1 day'}  
  
# 3% 1 year VAR  
scalingFactor <- 1  
z\_stat <- qnorm(VaR\_threshold/100, 0, 1, lower.tail = TRUE)  
VaR\_1year\_p\_3percent <- ROPT1\*scalingFactor + SDOPT1\*sqrt(scalingFactor) \* z\_stat  
VaR\_1year\_SPY\_3percent <- mean\_rSPY1\*scalingFactor + sd\_rSPY1\*sqrt(scalingFactor) \* z\_stat  
  
# 3% 6 month VAR  
scalingFactor <- 1/2  
VaR\_6month\_p\_3percent <- ROPT1\*scalingFactor + SDOPT1\*sqrt(scalingFactor) \* z\_stat  
VaR\_6month\_SPY\_3percent <- mean\_rSPY1\*scalingFactor + sd\_rSPY1\*sqrt(scalingFactor) \* z\_stat  
  
# 3% 1 month VAR  
scalingFactor <- 1/12  
VaR\_1month\_p\_3percent <- ROPT1\*scalingFactor + SDOPT1\*sqrt(scalingFactor) \* z\_stat  
VaR\_1month\_SPY\_3percent <- mean\_rSPY1\*scalingFactor + sd\_rSPY1\*sqrt(scalingFactor) \* z\_stat  
#########################################  
  
# Generate comparison table  
VaR\_stat\_p <- matrix(c(VaR\_1year\_p\_2percent, VaR\_6month\_p\_2percent, VaR\_1month\_p\_2percent,   
 VaR\_1year\_p\_3percent, VaR\_6month\_p\_3percent, VaR\_1month\_p\_3percent), nrow = 6)   
  
VaR\_stat\_SPY <- matrix(c(VaR\_1year\_SPY\_2percent, VaR\_6month\_SPY\_2percent, VaR\_1month\_SPY\_2percent,   
 VaR\_1year\_SPY\_3percent, VaR\_6month\_SPY\_3percent, VaR\_1month\_SPY\_3percent), nrow = 6)   
  
optstat.names <- c("2% 1yr", "2% 6mth", "2% 1month",   
 "3% 1yr", "3% 6mth", "3% 1month")  
  
dimnames(VaR\_stat\_p) <- list(optstat.names, "Holdings")  
dimnames(VaR\_stat\_SPY) <- list(optstat.names, "SPY")  
  
VaR\_stat <- data.frame(VaR\_stat\_p, VaR\_stat\_SPY)

The VaR statistics is as follow:

VaR\_stat

## Holdings SPY  
## 2% 1yr -0.07899253 -0.10093927  
## 2% 6mth -0.06590240 -0.07518002  
## 2% 1month -0.03276381 -0.03291141  
## 3% 1yr -0.06825518 -0.09089145  
## 3% 6mth -0.05830995 -0.06807514  
## 3% 1month -0.02966421 -0.03001085

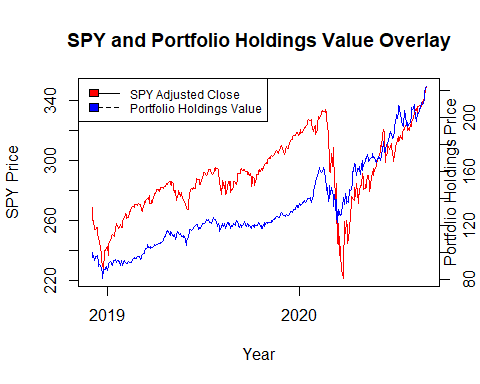
Through inspection, we observe that on shorter time horizon (1 month), both the holding and SPY portfolio has comparable value at risk. When the time horizon increase, the value at risk for the SPY portfolio starts to increase more than that of the holding portfolio.

### 9) Graph the scatter diagram of your portfolio and comment on trends, outliers, structural breaks and any other special features. Graph the scatter diagram of your portfolio and S&P 500 on the same coordinate system and compare the trends.

plot(as.zoo(adjSPY1[, "SPY.Adjusted"]), main = "SPY and Portfolio Holdings Value Overlay",   
 xlab = "Year", ylab = "SPY Price", col = "Red")  
  
par(new=TRUE)  
  
plot(holdings[, 1], type="line", xaxt="n", yaxt = "n", xlab = "", ylab = "", col = "Blue")

## Warning in plot.xy(xy, type, ...): plot type 'line' will be truncated to first  
## character

axis(4)  
mtext("Portfolio Holdings Price", side = 4)  
  
# Add legend  
legend("topleft", c("SPY Adjusted Close", "Portfolio Holdings Value"), lty = 1:3, cex = 0.75, fill = c("red", "blue"))



**Remarks**  
**1. Seasonal Trend:**  
Both display similar seasonal trend with up and down spikes in price days after days.  
**2. Cyclical Trend:**  
Both is affected by business cycle with similar peaks and troughs. SPY has more prominent troughs compared to that of the portfolio holdings. Especially during the period of March 2020 with the COVID-19 lockdowns started rolling across the countries.  
**3. Auto-correlation:**  
Both behave similar in this regard where both rises for some times when they rise and vice versa.  
**4. Randomness:**  
Price of both is unpredictable via inspection.  
**5. Time:**  
Both follows similar time trend.  
**6. Structural Break:**  
Both does not experience any structural break during this time frame. During the COVID-19 March sell-off, price recovered quickly in a V-shaped fashion.  
**7. Outliers:**  
SPY looks to have several outliers corresponded to the March 2020 sell-off when the indicies retreated 30% in a short period of a few weeks before quickly recovered.

## 4) Forecast the returns of the portfolio based on the CAPM model

Question covered: (10,11,12)  
**Methodology**        10) Using the CAPM equation of your portfolio do two periods ex-post forecasting of the returns to your portfolio and compare your forecast to the actual returns. Find the accuracy statistics of your forecast and report them.  
       11) Do two-periods ex-ante forecasting of returns to your portfolio assuming that the monthly risk premiums to S&P 500 will be 1.10% in 2020M9 and 1.25 in 2020M10.  
       12) Do forecasting of the returns to your portfolio for the period 2014M1-2020M8. Find the accuracy statistics of your forecast. Do a naïve forecasting of the returns to your portfolio for the period 2014M1-2020M8. Find the accuracy statistics of your forecast. Compare the forecasting accuracy criterion for the two forecasting methods. Which one results in a better forecasting outcome?

### 10) Using the CAPM equation of your portfolio do two periods ex-post forecasting of the returns to your portfolio and compare your forecast to the actual returns. Find the accuracy statistics of your forecast and report them.

Rerun the regression with the whole data set from 2018M12 to 2020M10:

# Set start date and end date of data  
start\_date2 <- "2018-12-01"  
end\_date2 <- "2020-10-31"  
  
# Get data  
getSymbols("MSFT", src = "yahoo", from = start\_date2, to = end\_date2)

## [1] "MSFT"

getSymbols("GWPH", src = "yahoo", , from = start\_date2, to = end\_date2)

## [1] "GWPH"

getSymbols("DIS", src = "yahoo", , from = start\_date2, to = end\_date2)

## [1] "DIS"

getSymbols("CAT", src = "yahoo", , from = start\_date2, to = end\_date2)

## [1] "CAT"

getSymbols("AMZN", src = "yahoo", , from = start\_date2, to = end\_date2)

## [1] "AMZN"

getSymbols("^GSPC", src = "yahoo", , from = start\_date2, to = end\_date2) # S&P 500

## [1] "^GSPC"

getSymbols("^TNX", src = "yahoo", from=start\_date2, to=end\_date2) # 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"

rMSFT2 <- diff(log(to.monthly(MSFT)$MSFT.Adjusted))  
rGWPH2 <- diff(log(to.monthly(GWPH)$GWPH.Adjusted))  
rDIS2 <- diff(log(to.monthly(DIS)$DIS.Adjusted))  
rCAT2 <- diff(log(to.monthly(CAT)$CAT.Adjusted))  
rAMZN2 <- diff(log(to.monthly(AMZN)$AMZN.Adjusted))  
rGSPC2 <- diff(log(to.monthly(GSPC)$GSPC.Adjusted))  
rTNX2 <- to.monthly(TNX)$TNX.Adjusted /1200 # Using monthly rate

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

rTNX2\_day <- TNX$TNX.Adjusted /1200  
rTNX2 <- rTNX2[-1,] # remove missing data due to lagging  
  
# Adjusted Prices  
adjMSFT2 <- MSFT$MSFT.Adjusted  
adjGWPH2 <- GWPH$GWPH.Adjusted  
adjDIS2 <- DIS$DIS.Adjusted  
adjCAT2 <- CAT$CAT.Adjusted  
adjAMZN2 <- AMZN$AMZN.Adjusted  
adjGSPC2 <- GSPC$GSPC.Adjusted  
  
holdings2 <- data.frame("Holding Value"=sharesMSFT\*adjMSFT2 +   
 sharesGWPH\*adjGWPH2 +  
 sharesDIS\*adjDIS2 +   
 sharesCAT\*adjCAT2 +  
 sharesAMZN\*adjAMZN2)  
names(holdings2)[1] <- "Port. Holdings Val" # rename column  
  
# Calculate and normalized the CAPM holdings  
ra2 <- diff(log(to.monthly(holdings2)[,1]))  
  
Y\_data <- na.omit(ra - rTNX2)

## Warning in `-.default`(ra, rTNX2): longer object length is not a multiple of  
## shorter object length

names(Y\_data)[1] <- "Port. Risk. Prem."  
X\_data <- na.omit(rGSPC2 - rTNX2)  
names(X\_data)[1] <- "Mrkt. Risk. Prem."  
# Market risk premium  
data <- data.frame(Y\_data, X\_data)  
names(data)[1] <- "Port. Risk. Prem."  
names(data)[2] <- "Mrkt. Risk. Prem."

Do the ex-post forecast:

fit\_expost <- lm(Y\_data~X\_data, data)  
fit\_expost

##   
## Call:  
## lm(formula = Y\_data ~ X\_data, data = data)  
##   
## Coefficients:  
## (Intercept) X\_data   
## 0.0294 -0.1985

pred\_expost <- predict(fit\_expost, data)  
pred\_expost

## Jan 2019 Feb 2019 Mar 2019 Apr 2019 May 2019 Jun 2019   
## 0.014801553 0.024035661 0.026275703 0.022162793 0.043266751 0.016500005   
## Jul 2019 Aug 2019 Sep 2019 Oct 2019 Nov 2019 Dec 2019   
## 0.027148346 0.033277459 0.026298421 0.025667607 0.023048641 0.024124368   
## Jan 2020 Feb 2020 Mar 2020 Apr 2020 May 2020 Jun 2020   
## 0.029978470 0.047033623 0.056057428 0.005796511 0.020718019 0.025893837   
## Jul 2020 Aug 2020 Sep 2020 Oct 2020   
## 0.018843081 0.016073153 0.037460709 0.035115960

Evaluating the errors:

actual\_expost <- data[, "Port. Risk. Prem."]  
sqrt(mean((pred\_expost - actual\_expost)^2))

## [1] 0.07607907

This error value means the model is off by on average. Significant!

### 11) Do two-periods ex-ante forecasting of returns to your portfolio assuming that the monthly risk premiums to S&P 500 will be 1.10% in 2020M9 and 1.25 in 2020M10.

Do the ex-ante forecast:

data1 <- data[1:20,]  
data2 <- data[21:22,]  
  
fit\_exante <- lm(Y\_data~X\_data, data=data1)  
fit\_exante

##   
## Call:  
## lm(formula = Y\_data ~ X\_data, data = data1)  
##   
## Coefficients:  
## (Intercept) X\_data   
## 0.0294 -0.1985

pred\_exante <- predict(fit\_exante, newdata=data2)

## Warning: 'newdata' had 2 rows but variables found have 22 rows

pred\_exante

## 1 2 3 4 5 6   
## 0.014801553 0.024035661 0.026275703 0.022162793 0.043266751 0.016500005   
## 7 8 9 10 11 12   
## 0.027148346 0.033277459 0.026298421 0.025667607 0.023048641 0.024124368   
## 13 14 15 16 17 18   
## 0.029978470 0.047033623 0.056057428 0.005796511 0.020718019 0.025893837   
## 19 20 21 22   
## 0.018843081 0.016073153 0.037460709 0.035115960

Evaluating the errors:

actual\_exante <- data[21:22, "Port. Risk. Prem."]  
sqrt(mean((pred\_exante - actual\_exante)^2))

## [1] 0.1043799

This error value means the model is off by on average. Even more significant than the ex-post forecast.

1. Do forecasting of the returns to your portfolio for the period 2014M1-2020M8. Find the accuracy statistics of your forecast. Do a naïve forecasting of the returns to your portfolio for the period 2014M1-2020M8. Find the accuracy statistics of your forecast. Compare the forecasting accuracy criterion for the two forecasting methods. Which one results in a better forecasting outcome?

Run a naive forecasting :

# Naive forecasting  
error <- diff(Y\_data)  
error

## Port. Risk. Prem.  
## Jan 2019 NA  
## Feb 2019 0.130407624  
## Mar 2019 0.052833381  
## Apr 2019 0.012097574  
## May 2019 0.004786171  
## Jun 2019 -0.163247376  
## Jul 2019 0.211072656  
## Aug 2019 -0.117627551  
## Sep 2019 -0.021790003  
## Oct 2019 0.004055114  
## Nov 2019 0.021229597  
## Dec 2019 0.028332669  
## Jan 2020 0.026380169  
## Feb 2020 0.029152706  
## Mar 2020 -0.110675000  
## Apr 2020 -0.079707917  
## May 2020 0.257338225  
## Jun 2020 -0.113316970  
## Jul 2020 0.081771690  
## Aug 2020 -0.070349143  
## Sep 2020 -0.181111949  
## Oct 2020 0.130318457

The residual of regression is:

fit\_exante$residuals

## Jan 2019 Feb 2019 Mar 2019 Apr 2019 May 2019 Jun 2019   
## -0.133331936 -0.012158420 0.038434919 0.054645403 0.038327616 -0.098153014   
## Jul 2019 Aug 2019 Sep 2019 Oct 2019 Nov 2019 Dec 2019   
## 0.102271302 -0.021485363 -0.036296329 -0.031610400 -0.007761836 0.019495106   
## Jan 2020 Feb 2020 Mar 2020 Apr 2020 May 2020 Jun 2020   
## 0.040021172 0.052118725 -0.067580080 -0.097027080 0.145389637 0.026896849   
## Jul 2020 Aug 2020 Sep 2020 Oct 2020   
## 0.115719294 0.048140079 -0.154359426 -0.021696219

As discussed above, ex-post (in-sample) forecasting yields a better forecast outcome with lower error.

The Mean Absolute Percent Error statistics:

The Mean Absolute Deviation statistics:

The Real Mean Squared Error statistics:

# Citations

“Amazon.com, Inc. (AMZN) Stock Price, News, Quote & History.” Yahoo! Finance, Yahoo!, 14 Nov. 2020, ca.finance.yahoo.com/quote/amzn/?p=amzn.

“Caterpillar, Inc. (CAT) Stock Price, News, Quote & History.” Yahoo! Finance, Yahoo!, 13 Nov. 2020, ca.finance.yahoo.com/quote/CAT/?p=CAT.

JeffCoxCNBCcom. “Fed Hikes Rate, Lowers 2019 Projection to 2 Increases.” CNBC, CNBC, 19 Dec. 2018, www.cnbc.com/2018/12/19/fed-hikes-rates-by-a-quarter-point-.html.

“GW Pharmaceuticals Plc (GWPH) Stock Price, News, Quote & History.” Yahoo! Finance, Yahoo!, 13 Nov. 2020, ca.finance.yahoo.com/quote/GWPH/?p=GWPH.

Moyer, Liz. “We’re Finding out Now Why the Stock Market Tanked in December.” CNBC, CNBC, 9 Jan. 2019, www.cnbc.com/2019/01/09/markets-december-tumble-may-have-hinted-at-profit-revisions-to-come.html.

“Microsoft Corporation (MSFT) Stock Price, News, Quote & History.” Yahoo! Finance, Yahoo!, 14 Nov. 2020, ca.finance.yahoo.com/quote/msft/?p=msft.

“Walt Disney Company (The) (DIS) Stock Price, News, Quote & History.” Yahoo! Finance, Yahoo!, 14 Nov. 2020, ca.finance.yahoo.com/quote/dis/?p=dis.