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

Data Analysis

Data Observation

```r
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 <- "2014-01-01"
end_date <- "2018-12-31"

# Get data for JPM, FB and the 10 year T-bill (TNX)
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"
```

```r
getSymbols("GWPH", src = "yahoo", , from = start_date, to = end_date)
```

```
## [1] "GWPH"
```

```r
getSymbols("DIS", src = "yahoo", , from = start_date, to = end_date)
```

```
## [1] "DIS"
```

```r
getSymbols("CAT", src = "yahoo", , from = start_date, to = end_date)
```

```
## [1] "CAT"
```

```r
getSymbols("AMZN", src = "yahoo", , from = start_date, to = end_date)
```

```
## [1] "AMZN"
```

```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
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:
pricTab1 <- data.frame(MSFT, GWPH, DIS, CAT, AMZN)

# Creates data frame of asset prices
retTab1 <- data.frame(rMSFT, rGWPH, rDIS, rCAT, rAMZN)

# Creates data frame of returns
EretTab1 <- data.frame(reMSFT, reGWPH, reDIS, reCAT, reAMZN)

# Excess return data frame
retTab1 <- retTab1[-1,]  # remove missing data due to lagging
EretTab1 <- EretTab1[-1,]  # remove missing data due to lagging
priceMat1 <- matrix(c(MSFT, GWPH, DIS, CAT, AMZN), nrow= length(MSFT), ncol=5, byrow=TRUE)  # creates a

# 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(retTab1)), nrow=5, ncol = 5, byrow=TRUE) # create a var/cov matrix by finding cov o
dimnames(VCV) <- list(asset.names, asset.names)  # assigns asset.names to the VCV matrix

#Calculate Returns
rm <- matrix(colMeans(retTab1, na.rm=TRUE))  # creates an average return matrix, omitting missing value
erm <- matrix(colMeans(EretTab1, na.rm=TRUE))  # creates an average excess return matrix, omitting miss
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

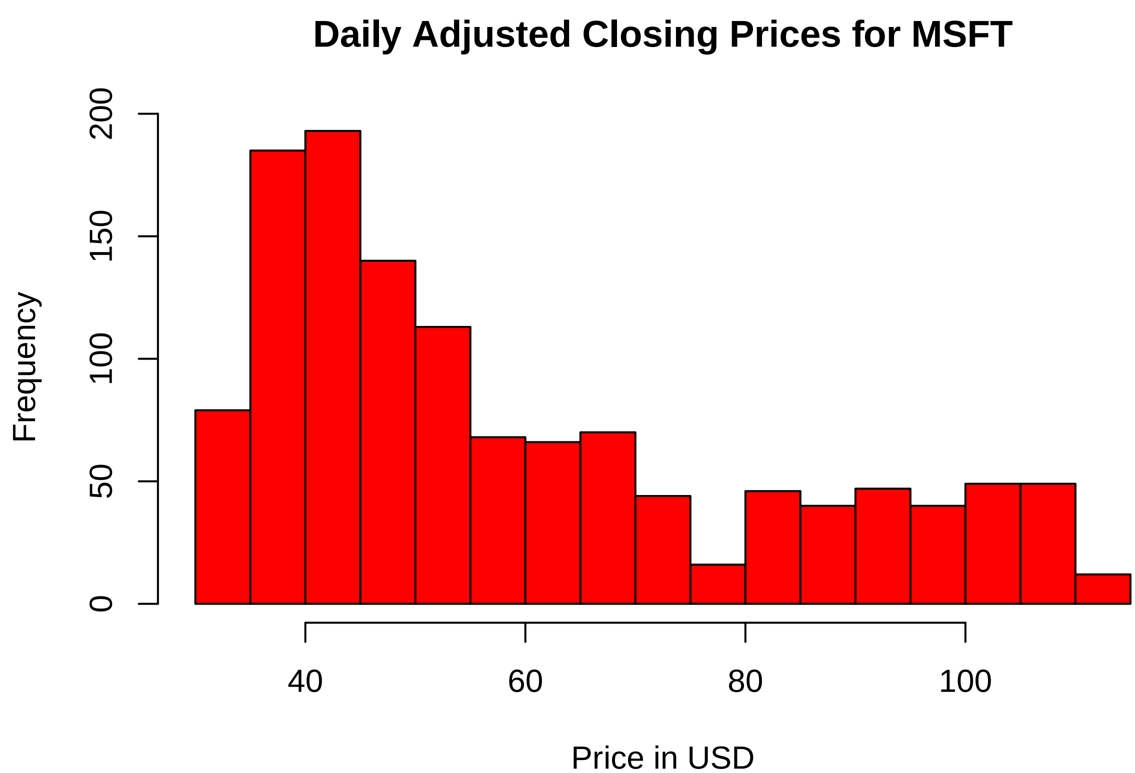
Instruments	Mean Returns	Variance of Returns	Beta (5Y Monthly)
MSFT	0.0185744	0.0035553	.87
GWPH	0.0090021	0.0293563	1.96
DIS	0.0078206	0.0026173	1.08
CAT	0.0076711	0.0057854	.98
AMZN	0.024	0.006961	1.3

Parameters of indices:

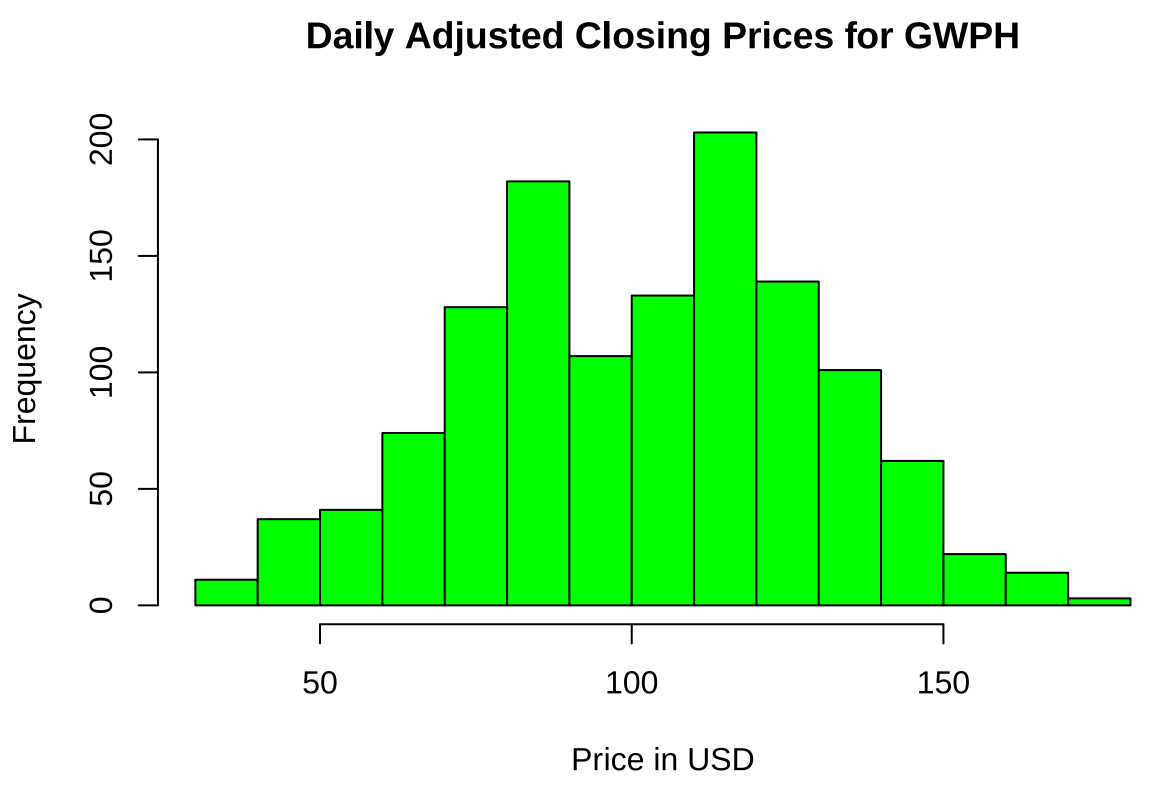
Instruments	Mean Returns	Variance of Returns	Beta
S&P 500	0.0056356	0.0010169	N/A
10-Year T-bill	0.0019378	0	N/A

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

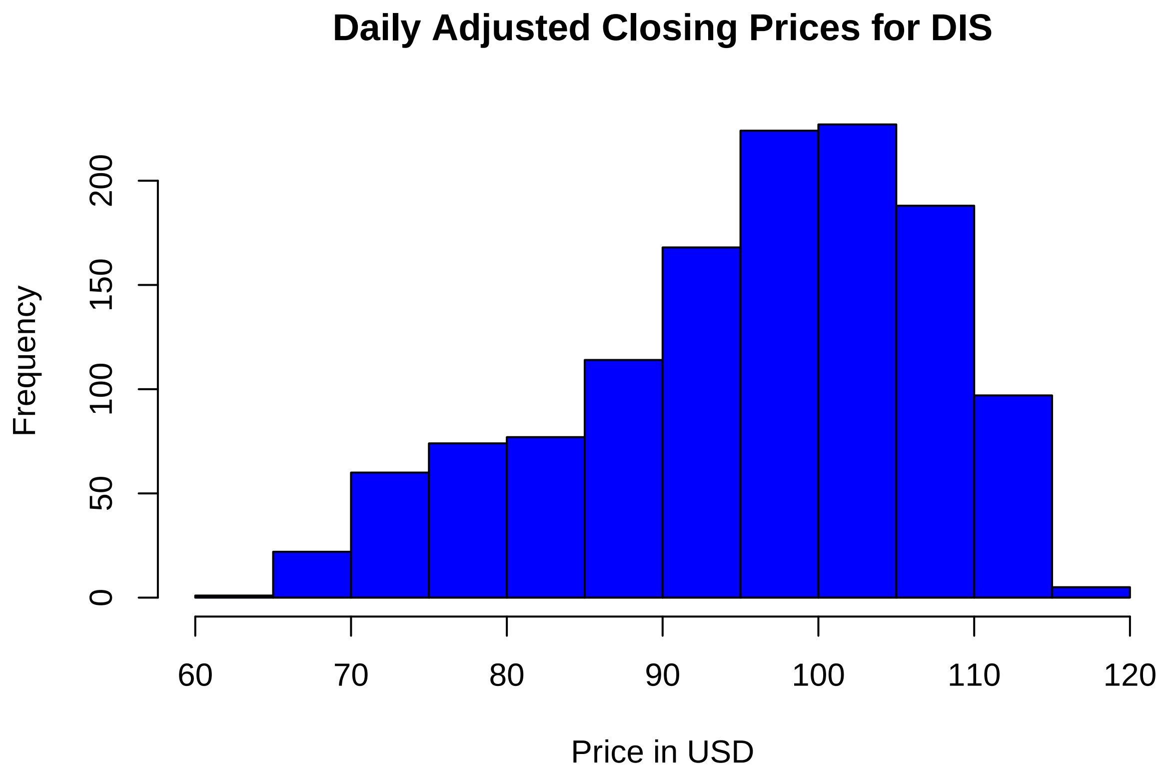
```
hist(adjMSFT,
     main='Daily Adjusted Closing Prices for MSFT',
     xlab='Price in USD',
     col='red',
    )
```



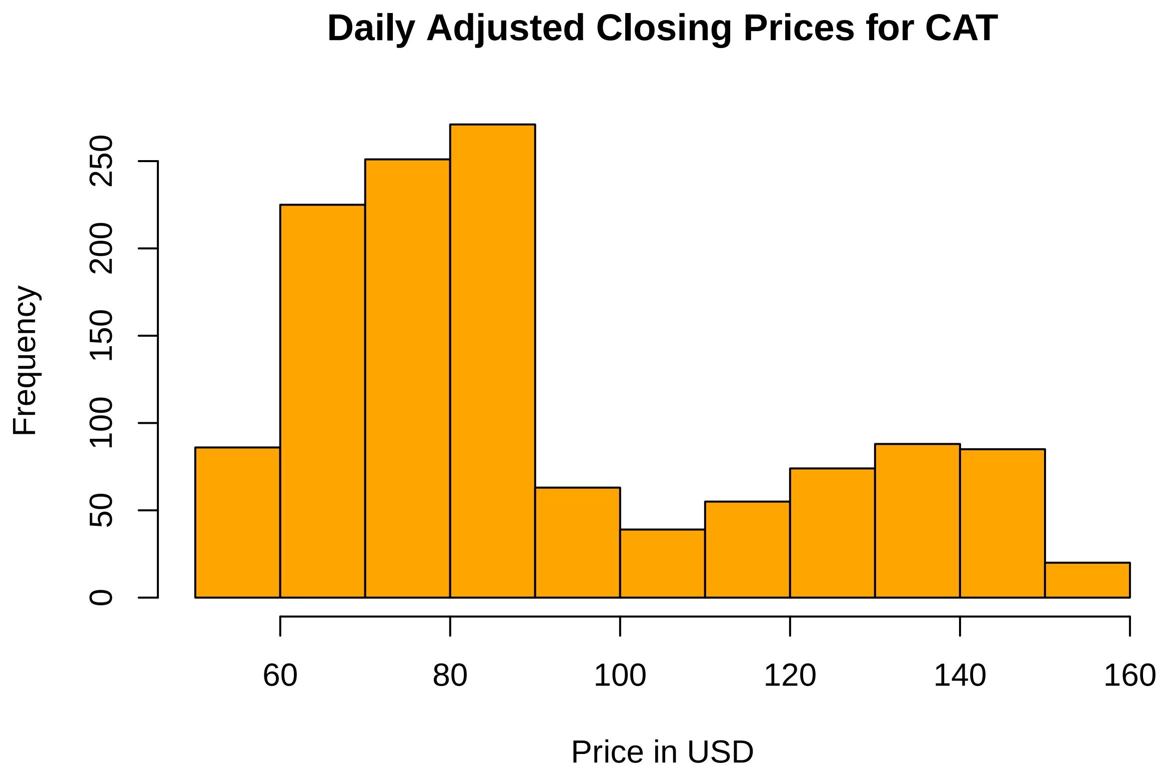
```r
hist(adjGWPH,
     main='Daily Adjusted Closing Prices for GWPH',
     xlab='Price in USD',
     col='green',
    )
```



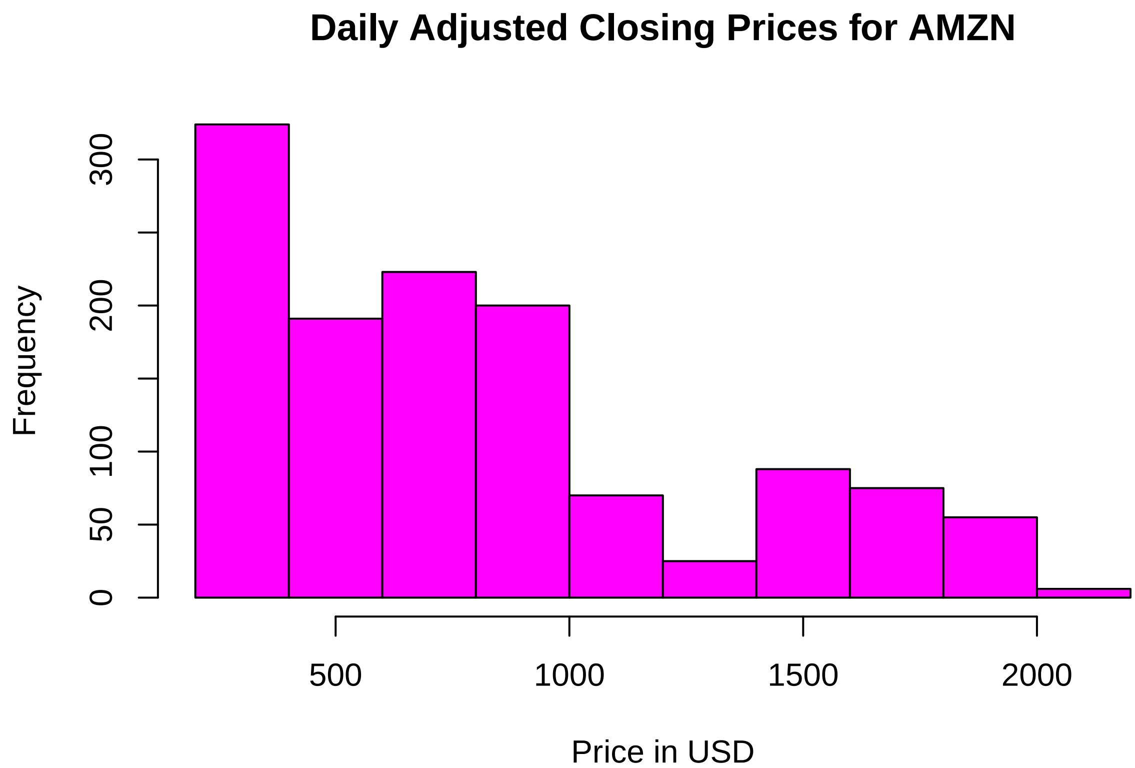

```r
hist(adjDIS,
     main='Daily Adjusted Closing Prices for DIS',
     xlab='Price in USD',
     col='blue',
    )
```




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




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



CAPM Portfolio Construction (2,3,4,5)

Methodology:

1) Construct a portfolio of the selected securities and observe the efficient frontier. (2)
2) Allocate $100–equal weight–among selected securities.(3)
3) Observe the holding value of the portfolio from December 2018 to August 2020.(3)
4) Estimate CAPM for the portfolio and observe its β and returns relative to the Security Market Line. (3,4,5)

1) Construct a portfolio of the selected securities and observe the efficient frontier. (2)

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 x_i denotes the weight of the investment in asset i ($i = 1, 2, 3, 4, 5$), and assume all money is invested in i, meaning $\sum x_i = x_1 + x_2 + x_3 + x_4 + x_5 = 1$.

The returns of the portfolio is:

$$R_{p,x} = x_1 * r_1 + x_2 * r_2 + x_3 * r_3 + x_4 * r_4 + x_5 * r_5$$

The expected returns on the portfolio is:

$$\begin{aligned} \mu_{p,x} &= E[R_{p,x}] \\ &= x_1 * \mu_1 + x_2 * \mu_2 + x_3 * \mu_3 + x_4 * \mu_4 + x_5 * \mu_5 \end{aligned} \tag{1}$$

The variance of the portfolio returns is:

$$\sigma^2_{p,x} = var(R_{p,x})$$

Formulating the Markowitz portfolio problem:

The investor's objective is:

$$\begin{aligned}
&max \quad \mu_p = w' * \mu \quad \text{s.t.} \\
&\qquad \sigma_p^2 = w' * (\sum) * w \quad \text{and} \quad w' * I = 1 \\
&\text{where:} \\
&\qquad w = \quad \text{matrix of asset weights in the portfolio} \\
&\qquad w' = \quad \text{transpose matrix of asset weights in the portfolio} \\
&\qquad \mu = \quad \text{matrix of mean returns of asset in the portfolio} \\
&\qquad \sum = \quad \text{Variance-covariance matrix of asset returns in the the portfolio} \\
&w' * I = \sum_{i=1}^{n} w_n \quad \text{or the sum weights of the asset in the portfolio, I is notation for identity matrix}
\end{aligned} \tag{2}$$

Let $\mu_{p,0}$ denotes a target expected return level. Formulate the problem:

$$\begin{aligned}
min \quad \sigma_{p,w}^2 &= w' * (\sum) * w \quad \text{s.t.} \\
\mu_p &= w' * \mu = \mu_{p,0}, \quad \text{and} \quad w' * I = 1
\end{aligned} \tag{3}$$

To solve this, form the Lagrangian function:

$$L(w, \lambda_1, \lambda_2) = w' * \sum *w + \lambda_1 * (w' * \mu - \mu_{p,0}) + \lambda_2 * (w' * I - 1) \tag{4}$$

Because there are two constraints ($w' * \mu = \mu_{p,0}$ and $w'1 = 1$) there are two Langrange multipliers λ_1 and λ_2. The first order condition for a minimum are the linear equations:

$$\begin{aligned}
\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial w} &= \frac{\partial(\sum *w^2)}{\partial w} + \frac{\partial(\lambda_1 * (w' * \mu - \mu_{p,0}))}{\partial w} + \frac{\lambda_2 * (w' * I - 1)}{\partial w} = 0 \\
\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_1} &= 0 \\
\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_2} &= 0
\end{aligned} \tag{5}$$

Simplify, we have:

$$\begin{aligned}
\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial w} &= 2 * \sum *w + \lambda_1 * \mu + \lambda_2 * I = 0 \\
\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_1} &= w' * \mu - \mu_{p,0} = 0 \\
\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_2} &= w' * I - 1 = 0
\end{aligned} \tag{6}$$

Rewrite in matrix form:

$$\begin{pmatrix} 2 * \sum & \mu & I \\ \mu' & 0 & 0 \\ I' & 0 & 0 \end{pmatrix} * \begin{pmatrix} w \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \mu_{p,0} \\ I \end{pmatrix} \tag{7}$$

or

$$
\begin{aligned}
A * z_w &= b_0 \\
&\text{where} \\
A &= \begin{pmatrix} 2*\sum & \mu & I \\ \mu' & 0 & 0 \\ I' & 0 & 0 \end{pmatrix} \\
z_w &= \begin{pmatrix} w \\ \lambda_1 \\ \lambda_2 \end{pmatrix} \\
b_0 &= \begin{pmatrix} 0 \\ \mu_{p,0} \\ I \end{pmatrix}
\end{aligned} \tag{8}
$$

The solution for z_w is:

$$z_w = A^{-1} * b_0 \tag{9}$$

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: 0.001933

```
# 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,
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:

```r
WOPT
```

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

The statistics of the optimal portfolio is:

```r
PTBL
```

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

```r
# 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
```

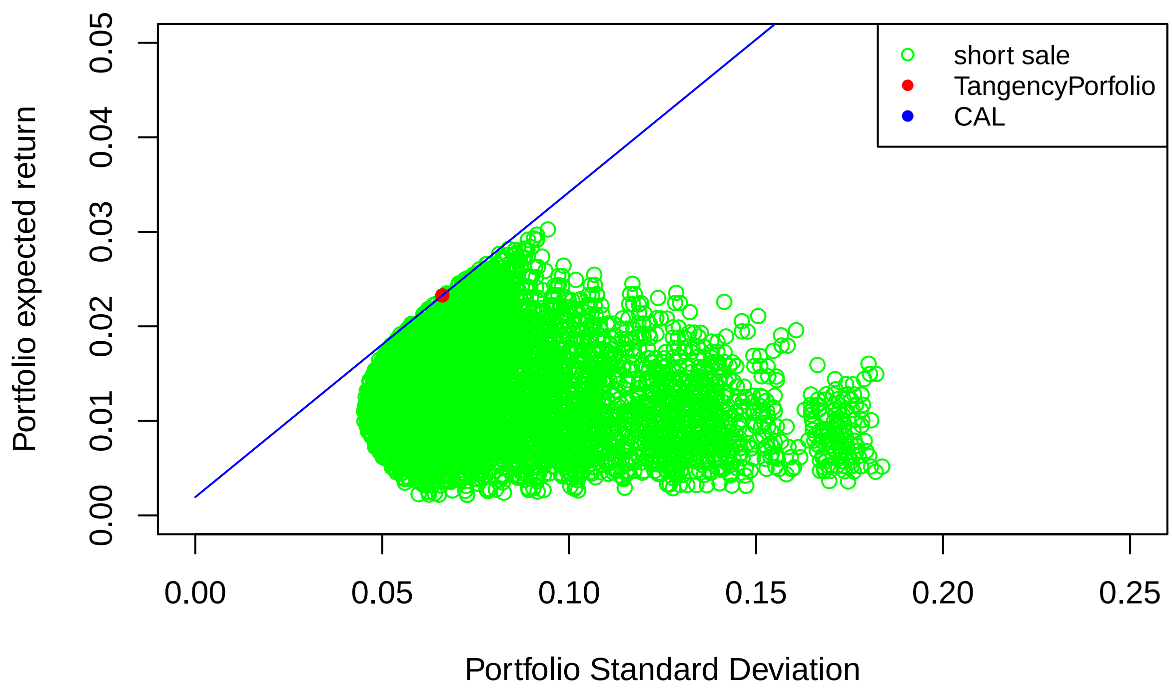
```
## Warning in SRatio * f: Recycling array of length 1 in array-vector arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
#Plot the portfolio possibilities curve:
plot(StdDev_p, Rport, col="green1", xlab="Portfolio Standard Deviation", ylab= "Portfolio expected retu

#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 =
```



The tangency point of the Capital Allocation Line is the point where the weights of the portfolio is optimal, represented by the point (σ_p, r_p) which is $(0.0660904, 0.0232685)$.

The efficient frontier is the portfolio possibility curve represented by the equation: $CAL = 0.001933 + 0.3228223 * \sigma_p$

2) Allocate $100–equal weight–among selected securities.(3)

```r
# Set start date and end date of data
start_date1 <- "2018-12-01"
end_date1 <- "2020-08-31"

# Get data for JPM, FB and the 10 year T-bill (TNX)
getSymbols("MSFT", src = "yahoo", from = start_date1, to = end_date1)
```

```
## [1] "MSFT"
```

```r
getSymbols("GWPH", src = "yahoo", , from = start_date1, to = end_date1)
```

```
## [1] "GWPH"
```

```r
getSymbols("DIS", src = "yahoo", , from = start_date1, to = end_date1)
```

```
## [1] "DIS"
```

```r
getSymbols("CAT", src = "yahoo", , from = start_date1, to = end_date1)
```

```
## [1] "CAT"
```

```r
getSymbols("AMZN", src = "yahoo", , from = start_date1, to = end_date1)
```

```
## [1] "AMZN"
```

```r
getSymbols("^GSPC", src = "yahoo", , from = start_date1, to = end_date1) # S&P 500
```

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

```r
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"
```

```r
# 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(#"MSFT"=sharesMSFT*adjMSFT1,
                       #"GWPH"=sharesGWPH*adjGWPH1,
                       #"DIS"=sharesDIS*adjDIS1,
                       #"CAT"=sharesCAT*adjCAT1,
                       #"AMZN"=sharesAMZN*adjAMZN1,
                       "Holding Value"=sharesMSFT*adjMSFT1 +
                                       sharesGWPH*adjGWPH1 +
                                       sharesDIS*adjDIS1 +
                                       sharesCAT*adjCAT1 +
                                       sharesAMZN*adjAMZN1)
names(holdings)[1] <- "Port. Holdings Val"  # rename column
```

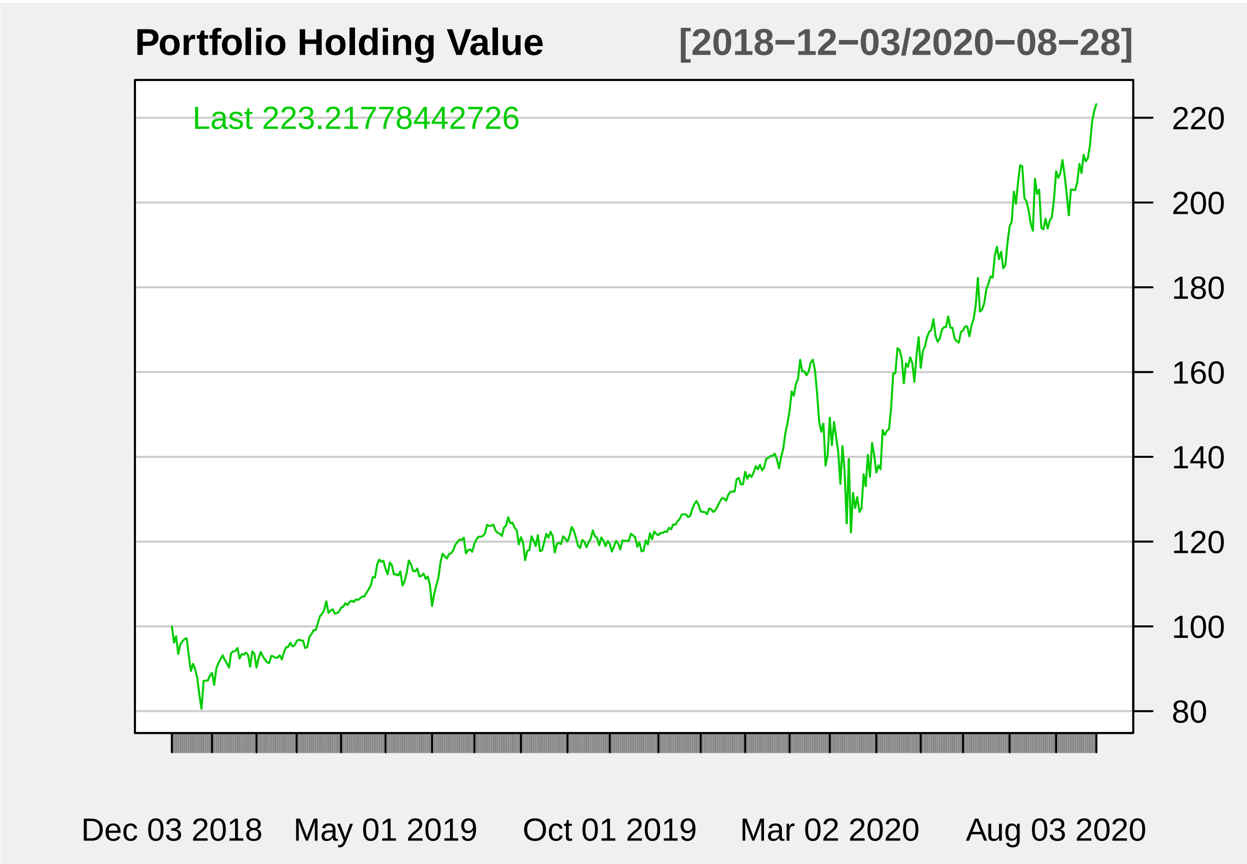
Based on the optimal weighting, to allocate $100 to the portfolio, we would be purchase the following amount of each security:

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

3) Observe the holding value of the portfolio from December 2018 to August 2020.(3)

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 start going into effects in the U.S.

4) Estimate CAPM for the portfolio and observe its β and returns relative to the Security Market Line.

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

$$
\begin{aligned}
E(R_a - R_f) &= \beta * (R_m - R_f) \\
&\text{or} \\
R_a &= R_f + \beta * (R_m - R_f) \\
R_a - R_f &= \alpha_{Jensen} + \beta * (R_m - R_f) \\
&\text{or} \\
Y &= \alpha_{Jensen} + \beta * X \\
&\text{with} \\
Y &= R_a - R_f \\
X &= R_m - R_f \\
\beta &= \text{Market risk or systematic risk}
\end{aligned}
\tag{10}
$$

Impact of COVID-19 on a CAPM Portfolio (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)

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

Methodology:

1) Compare the statistics of the CAPM portfolio to the State Street's SPDR S&P 500 Trust ETF portfolio. (7,8)

2) Observe the differences between the CAPM portfolio and the State Street's SPDR S&P 500 Trust ETF portfolio. (9)

Forecast the returns of the portfolio CATsed on the CAPM model (10,11,12)

1) Do two periods ex-post forecasting of returns and compare to actual returns. (10)
2) Do two periods ex-ante forecasting of returns and compare to actual returns. (11)
3) Forecasting the portfolio returns for the period of January 2014 to August 2018. (12)

****

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.