Spring 2021 Project

Jeff Nguyen

01/04/2021

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 01, 2016 to December 31, 2018.

Methodology

1) Select at least five stocks from different industries.

2) Construct a portfolio of the selected stocks and graph the efficient frontier.

   a. Find the optimum weights using MPT.

   b. Using the optimum weights and monthly adjusted closing prices at the end of 2018 allocate $100.00 among the selected stocks. On 1/1/2019, the portfolio will have a value of 100 as an index.

   c. Using the daily adjusted closing prices from 1/2/ 2019 to present calculate the holding values of the portfolio. Assume fixed holdings with no re-balancing taking place over time. Calculate the CAL equation and graph the CAL and the efficient frontier.

3) Do Naive, MA(5), MA(15), ES, Holt, and Holt-Winters forecasting of your portfolio returns and do a three-period-ahead forecasting of the portfolio returns for each forecast. Estimate the accuracy statistics.

4) Start with the regression analysis and forecasting of your portfolio returns. Use the CAPM and three-factor CAPM (Fama-French) models to estimate the coefficients of the models and use them for forecasting. Do a 10-days ex-post forecasting of the portfolio risk premiums and compare the forecasted value to actual ones. Do a three-period-ahead (ex-ante) forecasting of the portfolio risk premiums and write confidence intervals.

5) Do an ARIMA model of your portfolio returns and use it for three-period ahead forecasting of the returns to portfolio. Write confidence interval. Estimate the accuracy statistics.

6) Test your ARIMA model for the stability of the ARIMA coefficients.

7) Test your ARIMA model for the existence of ARCH and GARCH and do proper corrections, if needed.

8) Find different time-series measures of volatility for your portfolio returns (see the volatility file posted on Blackboard) and do a three-period ahead forecasting of the portfolio volatility. Compare the different measures of volatility with GARCH.

9) Use the accuracy statistics of the different forecasting techniques to decide which technique fits the data best.

10) Test whether your portfolio index conforms to the efficient market hypothesis.

11) Find 1% and 3% daily and monthly VaR of your portfolio.

12) Find 1% and 3% daily and monthly equity EVaR of your portfolio.

13) Graph the security Market Line (SML) of your portfolio and test whether you would add a stock of your own choice to the portfolio or not.

14) Do an intervention function analysis of the March 15th closing of US economy due to COVID19. Did the event have any effect on return to your portfolio.

15) Do a 2-variable VAR between your portfolio index and S&P500 index. Graph the Impulse response function of the VAR and comment on the relationship.

Data Analysis

1) Select at least five stocks from different industries.

```
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 <- "2016-01-01"
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"
```

```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)
```

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

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

```
## 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
```

```r
# 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)
```

```r
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 

# 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 o
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 value
erm <- matrix(colMeans(EretTabl, 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	0.0190403	0.0027112	.87
GWPH	0.0183674	0.0299313	1.96

Instruments	Mean Returns	Variance of Returns	Beta (5Y Monthly)
DIS	0.0045494	0.0017214	1.08
CAT	0.0223445	0.0058996	.98
AMZN	0.0263838	0.0062955	1.3

Parameters of indices:

| Instruments	Mean Returns	Variance of Returns	Beta
S&P 500	0.0070788	0.0010008	N/A
10-Year T-bill	0.0019565	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',
    )
```
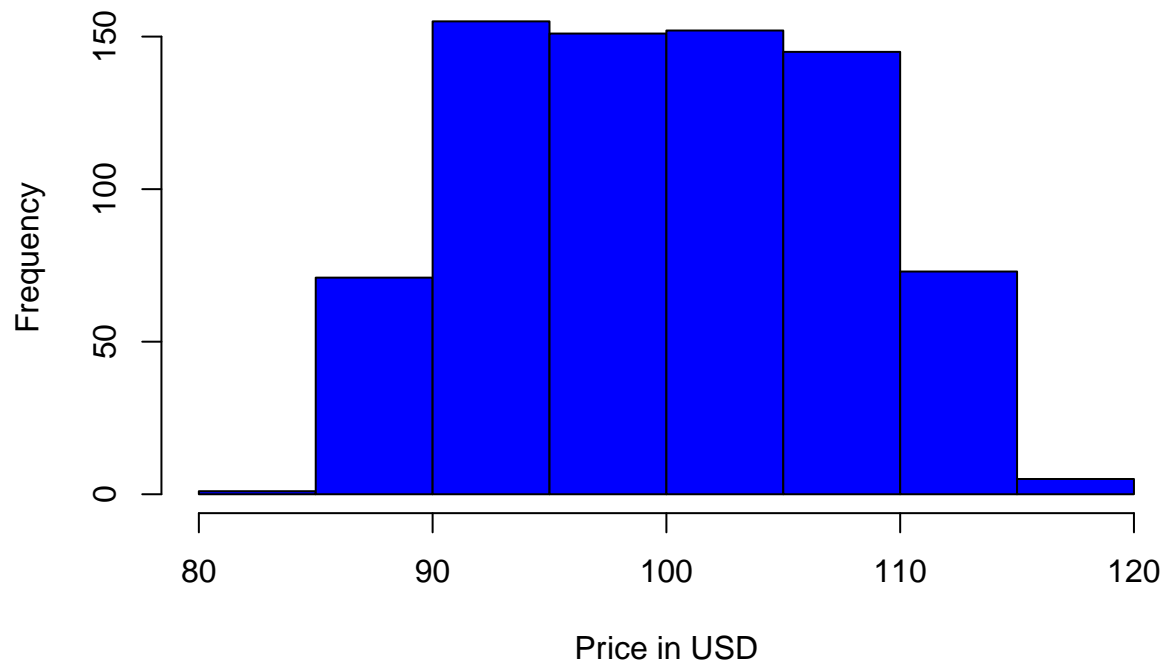
Daily Adjusted Closing Prices for MSFT
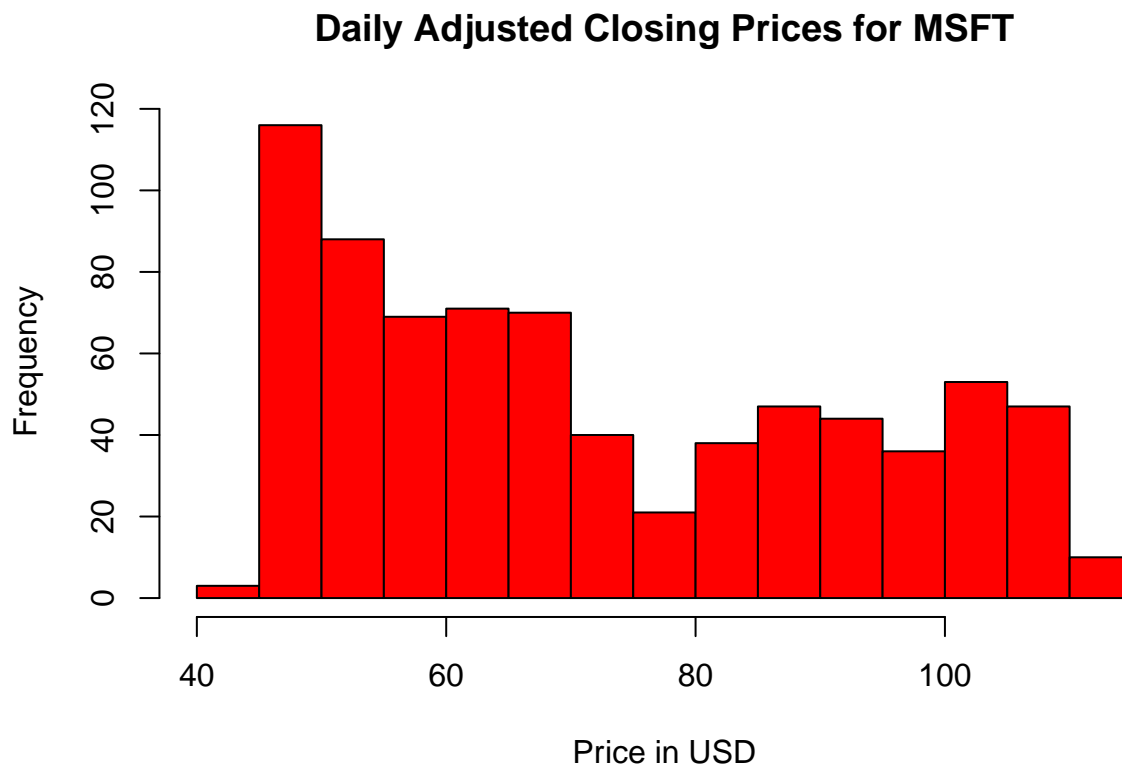

```
hist(adjGWPH,
     main='Daily Adjusted Closing Prices for GWPH',
     xlab='Price in USD',
```

```
    col='green',
    )
```

Daily Adjusted Closing Prices for GWPH
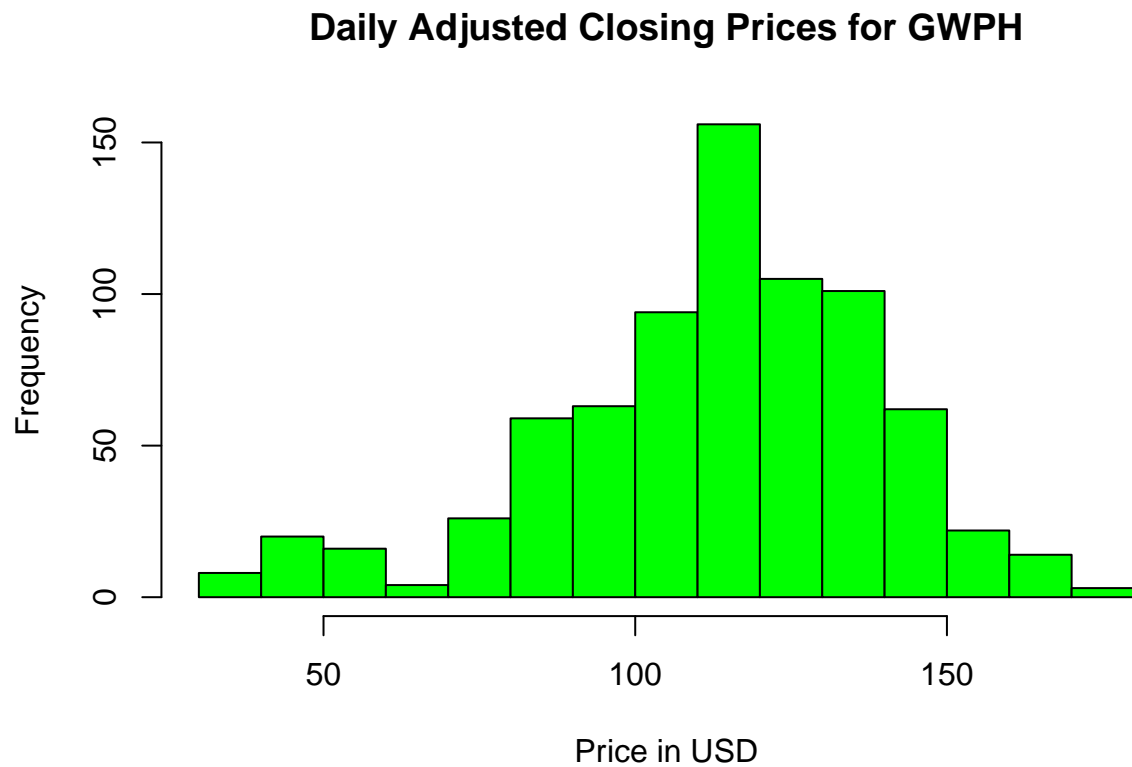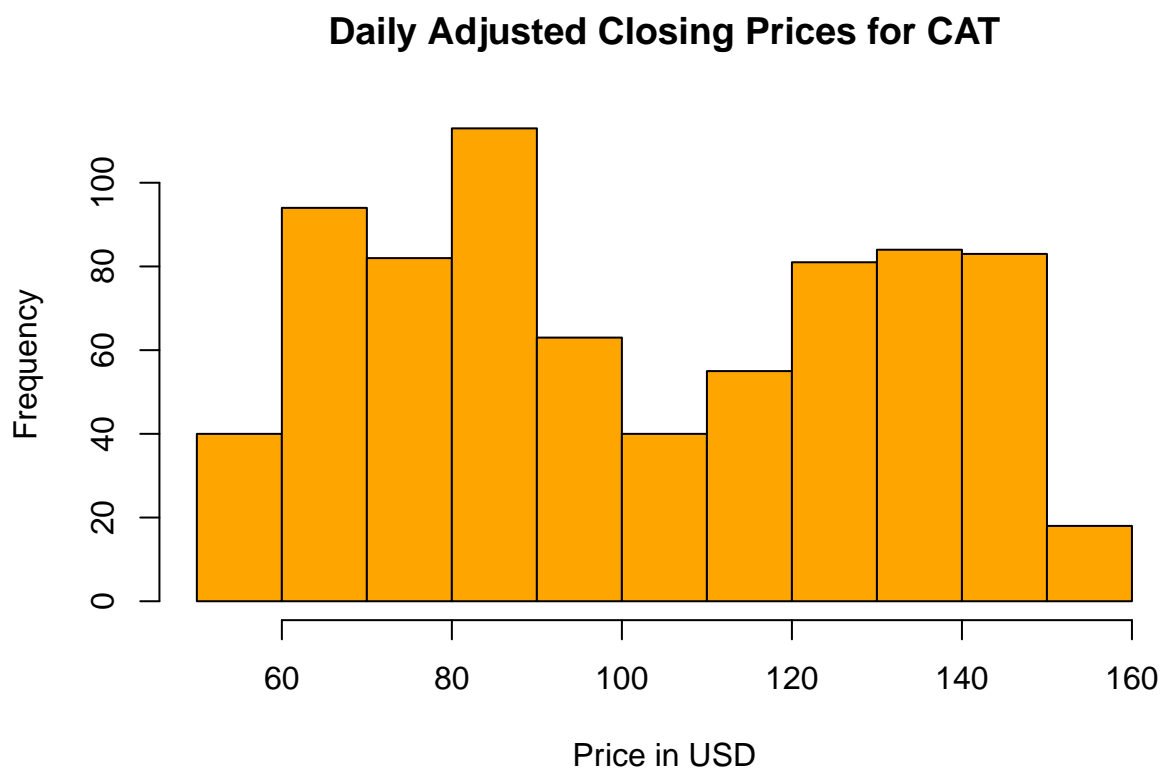


```
hist(adjDIS,
    main='Daily Adjusted Closing Prices for DIS',
    xlab='Price in USD',
    col='blue',
    )
```
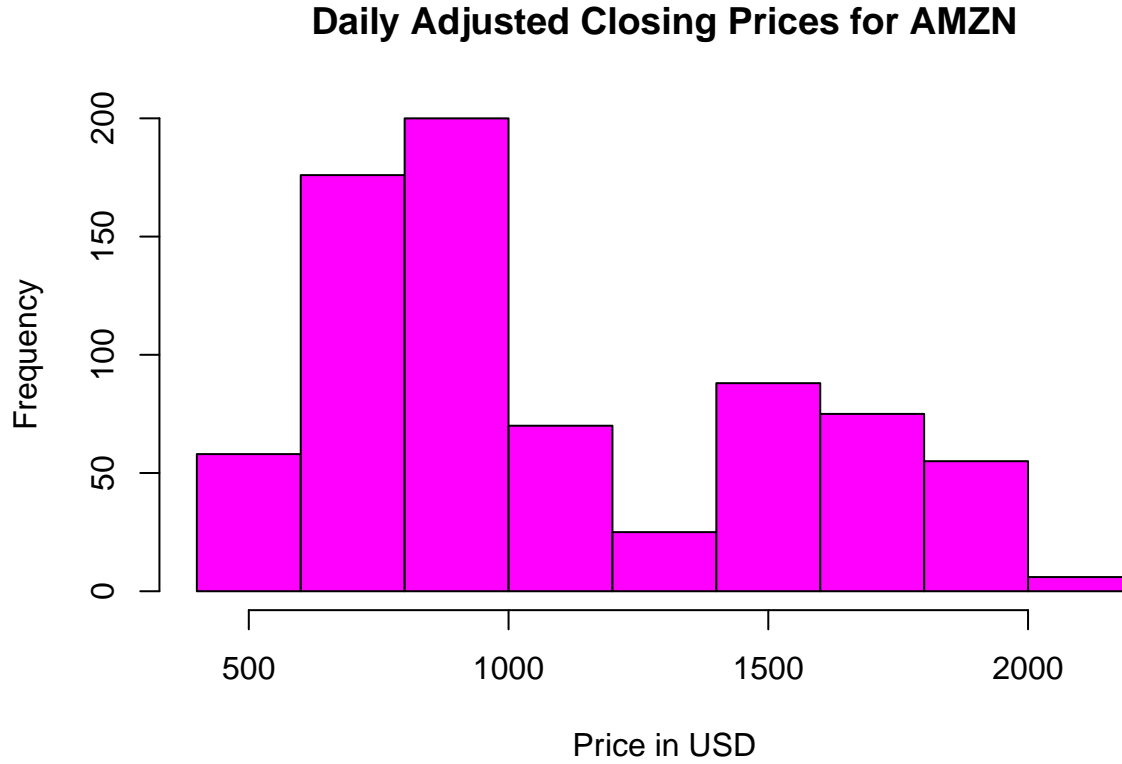
Daily Adjusted Closing Prices for DIS


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

Daily Adjusted Closing Prices for CAT

Price in USD

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

Daily Adjusted Closing Prices for AMZN

CAPM Portfolio Construction

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

The variance of the portfolio returns is:
$$\sigma_{p,x}^2 = var(R_{p,x})$$

Formulating the Markowitz portfolio problem:

The investor's objective is:

$$max \quad \mu_p = w' * \mu \quad \text{s.t.}$$

$$\sigma_p^2 = w' * \left(\sum\right) * w \quad \text{and} \quad w' * I = 1$$

where:

$$w = \quad \text{matrix of asset weights in the portfolio}$$

$$w' = \quad \text{transpose matrix of asset weights in the portfolio}$$

$$\mu = \quad \text{matrix of mean returns of asset in the portfolio}$$

$$\sum = \quad \text{Variance-covariance matrix of asset returns in the the portfolio}$$

$$w' * I = \sum_{i=1}^{n} w_n \quad \text{or the sum weights of the asset in the portfolio, I is notation for identity matrix}$$

$$(2)$$

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

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

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:

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

$$\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_1} = 0 \tag{5}$$

$$\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_2} = 0$$

Simplify, we have:

$$\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial w} = 2 * \sum *w + \lambda_1 * \mu + \lambda_2 * I = 0$$

$$\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_1} = w' * \mu - \mu_{p,0} = 0 \tag{6}$$

$$\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_2} = w' * I - 1 = 0$$

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

$$A * z_w = b_0$$

$$\text{where}$$

$$A = \begin{pmatrix} 2 * \sum & \mu & I \\ \mu' & 0 & 0 \\ I' & 0 & 0 \end{pmatrix}$$

$$z_w = \begin{pmatrix} w \\ \lambda_1 \\ \lambda_2 \end{pmatrix}$$

$$b_0 = \begin{pmatrix} 0 \\ \mu_{p,0} \\ I \end{pmatrix}$$

(8)

The solution for z_w is:

$$z_w = A^{-1} * b_0$$

(9)

The variance-covariance matrix is as follow:

VCV

```
##                MSFT          GWPH           DIS           CAT          AMZN
## MSFT 0.0027112031 0.003255723 0.0004124035 0.0014986841 0.0025785122
## GWPH 0.0032557231 0.029931273 0.0020363050 0.0058326666 0.0064760106
## DIS  0.0004124035 0.002036305 0.0017214342 0.0009435592 0.0005948209
## CAT  0.0014986841 0.005832667 0.0009435592 0.0058996292 0.0023212055
## AMZN 0.0025785122 0.006476011 0.0005948209 0.0023212055 0.0062955209
```

The monthly risk-free rate is: 0.0019664

```
# 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:

WOPT

```
##         Weights
## MSFT  0.53181762
## GWPH -0.11209776
## DIS  -0.08535046
## CAT   0.34599850
## AMZN  0.31963210
```

The statistics of the optimal portfolio is:

PTBL

```
##          Opt. Portfolio
## Return     0.023842999
## Variance   0.003055134
## Std Dev    0.055273268
## Sharpe     0.395790195
```

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

```r
# 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"
```

```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
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
```

```r
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	0.5318176	0.4876924
GWPH	-0.1120978	-0.0887903
DIS	-0.0853505	-0.0752235
CAT	0.3459985	0.2651849
AMZN	0.3196321	0.0180343

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"))
```
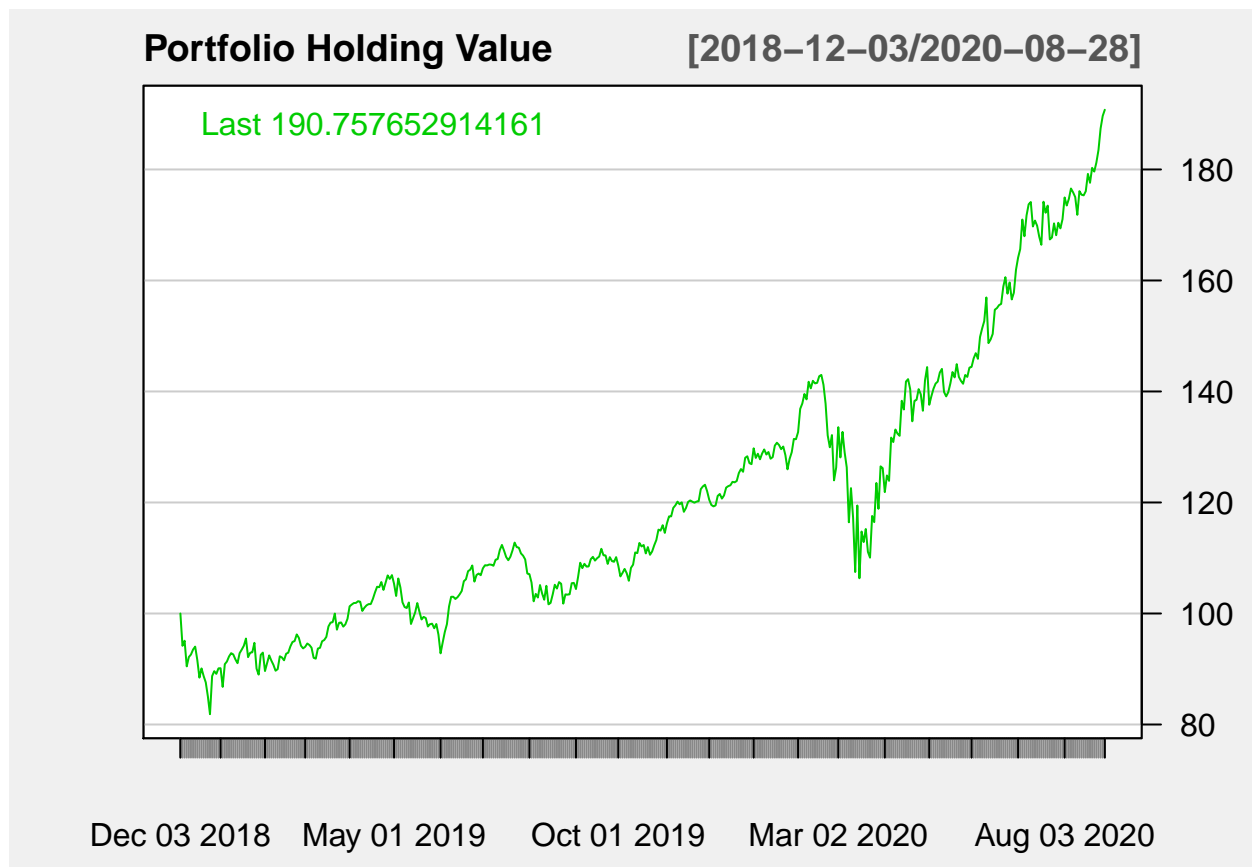

By inspection we can see the portfolio experience a sharp sell off of almost 20% in December 2018, coincide with the broad U.S.market selloff due to a combination of the FED hiking the federal funds rate by 25 basis points to a targeted range of 2.25% to 2.5% (JeffCoxCNBCcom) and corporations followed 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.

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 (σ_p, r_p) which is $(0.0552733, 0.023843)$.

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

The efficient frontier is the portfolio possibility curve represented by the equation: $CAL = 0.0019664 + 0.3957902 * \sigma_p$

```
# 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 Retur

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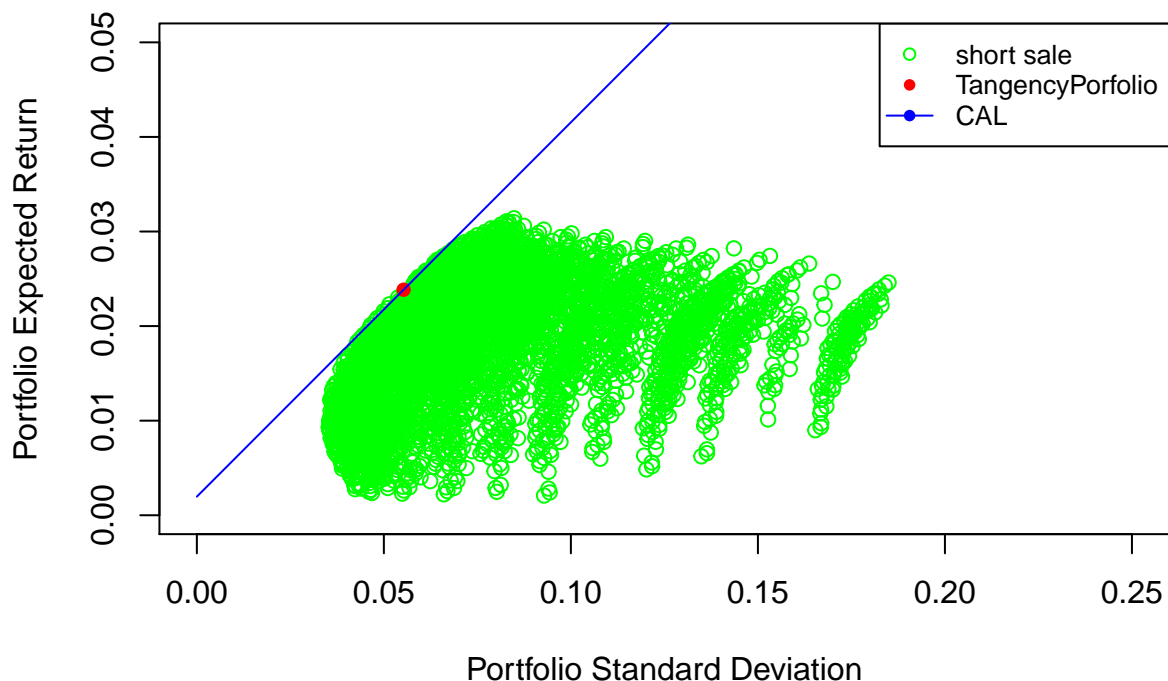
Estimate CAPM for your portfolio and graph the estimated β of the CAPM and the average return of your portfolio as a point relative to SML.

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

$$E(R_a - R_f) = \beta * (R_m - R_f)$$
$$\text{or}$$
$$R_a = R_f + \beta * (R_m - R_f)$$
$$R_a - R_f = \alpha_{Jensen} + \beta * (R_m - R_f)$$
$$\text{or}$$
$$Y = \alpha_{Jensen} + \beta * X + epsilon \tag{10}$$
$$\text{with}$$
$$Y = R_a - R_f$$
$$X = R_m - R_f$$
$$\beta = \text{Market risk or systematic risk}$$
$$\epsilon = \text{stochastic error term}$$

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:

$$H_0 : \alpha = 0$$
$$H_a : \alpha \neq 0$$
$$\text{and} \tag{11}$$
$$H_0 : \beta = 0$$
$$H_a : \beta \neq 0$$

```r
# 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] 0.02665436
```

```r
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)
```
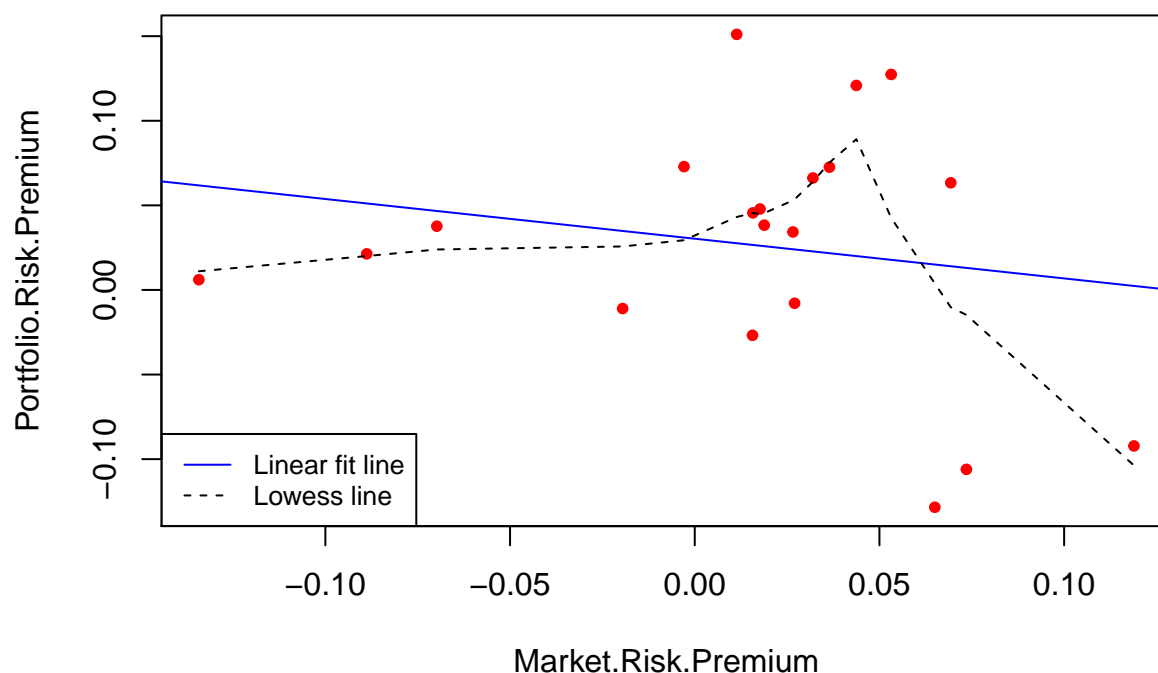
Relationship Between Market & Portfolio Risk Premium

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 X variable on the x-axis) and the CAPM Portfolio Risk Premium (the dependent Y variable on the y-axis).

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

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

```
##
## Call:
## lm(formula = Y ~ X, data = data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14347 -0.04775  0.01132  0.04489  0.12346
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03029    0.01735   1.746   0.0979 .
## X           -0.23470    0.29421  -0.798   0.4354
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07488 on 18 degrees of freedom
## Multiple R-squared:  0.03415,    Adjusted R-squared:  -0.01951
## F-statistic: 0.6364 on 1 and 18 DF,  p-value: 0.4354
```

The estimated equation is $Y = .04050 - .34504 * X$, where the p_{value} for the intercept $.0308 < .05$. Therefore, we reject the null hypothesis at 95% confidence level that the intercept α_{Jensen} statistically is no different from zero. Thus, we reject the null hypothesis $H_0 : \alpha = 0$ and accept the null hypothesis $H_a : \alpha \neq 0$.

The coefficient $\beta = 1.07468$ represents the increase in portfolio risk premium relative to increase in the market portfolio risk premium. The p_{value} for β is $.2544 > .05$, implying that the coefficient β statistically is insignificant at 95% or more, and we accept the null hypothesis $H_0 : \beta = 0$ and reject the alternative hypothesis $H_a : \beta \neq 0$.

Goodness of Fit:
Through inspection, we observe the $R^2 = .07151$ value to not be close to 1 at all. $R^2 = .07151$ implies that 7.15 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 $S.E. = .07458$.
From this, we can calculate the forecasting efficiency statistic to be:

$$\frac{S.E.}{\overline{Y}} = \frac{.05618}{0.0266544} \tag{12}$$
$$= 279.8\% > 10\%$$

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:

```r
# 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 Lin

# 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))
```
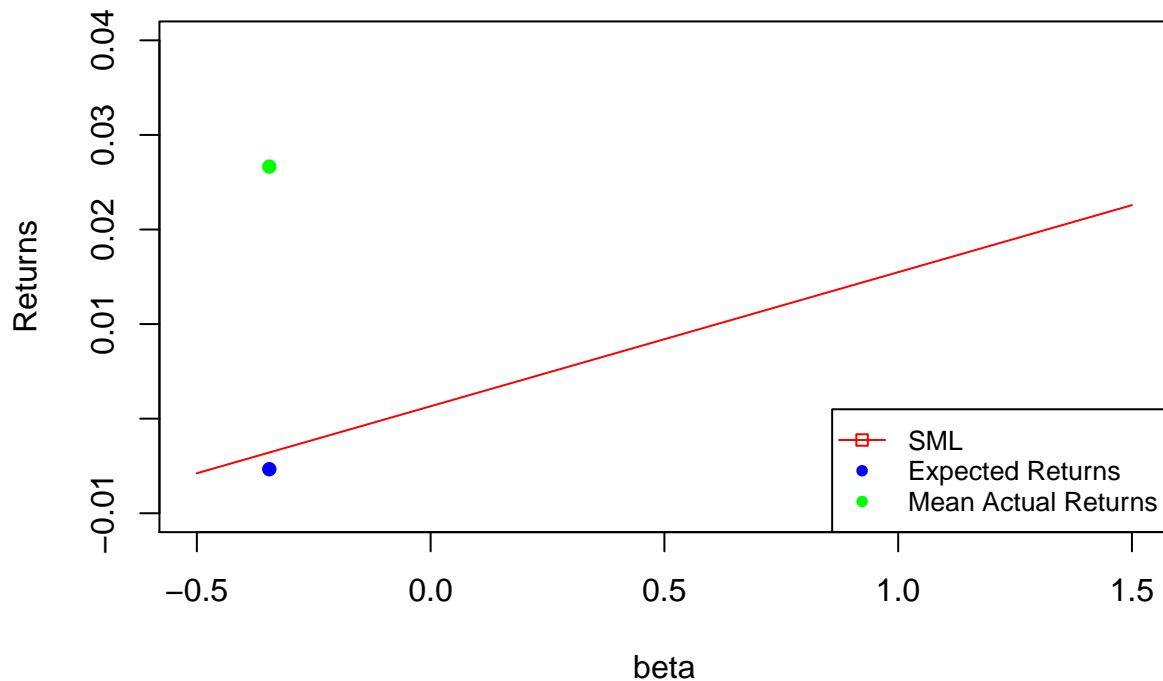
CAPM portfolio beta relative to the Security Market Line

The Security Market Line pass through the point $(0, \overline{R_f})$ and $(1, \overline{X})$, which are $(0, 0.0013135)$ and $(1, 0.0154876)$.

Relative to its market risk of $beta = -.34504$, the expected return is -0.0053439 and the average return is 0.0266544. We can observe that at this estimated β, the expected return is below the security market line and the actual average return is above the security market line.

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.