Portfolio Optimization

Kushal Kharel

7/9/2021

Let us begin with some mathematical terms and concepts that is necessary to understand the analysis.

When making an investment decision, the initial capital to invest is known but the amount to be returned is uncertain. To deal with such situation we treat uncertainty by mean-variance analysis. This method uses probability theory and leads to mathematical procedures. An investment instrument can be bought and sold frequently which we call asset. Suppose investor purchases an asset at time zero and sells it one year later. If X_0 and X_1 is the amount of money invested and received and R is the total return then,

$R = \frac{X_0}{X_1}$

The rate of return r is defined as

$r = \frac{X_1 - X_0}{X_0}$

We can see that these two are related by

$R = 1 + r$

Suppose now we have n different assets. We can form a portfolio with these n assets. Suppose that this is done by apportioning an amount X_0 among n assets. We then select amounts X_{0i}, $i = 1, 2, ...n$, such that $\sum_{i=1}^{n} X_{0i} = X_0$, where X_{0i} is the amount invested in the i^{th} asset.

The amount invested can be expressed as fractions of the total investment. Thus,

$X_{0i} = \omega_i X_0$, $i = 1, 2, ..., n$ where ω_i is the weight of asset i in the portfolio. We know that $\sum_{i=1}^{n} \omega_i = 1$

Let R_i represents total return of asset i. Then $R_i X_{0i} = R_i \omega_i X_0$ is the amount generated by the i^{th} asset at the end of period. The total of this portfolio at the end of the period is $\sum_{i=1}^{n} R_i \omega_i X_0$. Hence, the total return of the portfolio is,

$R = \frac{\sum_{i=1}^{n} R_i \omega_i X_0}{X_0} = \sum_{i=1}^{n} \omega_i R_i$

Since, $\sum_{i=1}^{n} \omega_i = 1$, we have

$r = \sum_{i=1}^{n} \omega_i r_i$

We now turn towards the concept of random variable, expected value, variance, covariance, correlation coefficient.

Let us suppose that X is a random quantity that can take any values on a number line, say, $X_1, X_2, X_3, ..., X_n$. Also let us assume that p_i represents the probability of occurrence of X_i. We know from our probability class that $\sum_{i=1}^{m} p_i = 1$ and $p_i \geq 0$ for each i. Then the quantity X is known as a random variable.

The expected value of a random variable X is the average obtained when probabilities are expressed as frequencies. It is defined as,

$E(X) = \overline{X} = \sum_{i=1}^{m} p_i X_i$

Variance is the degree of deviation from the mean. For any random variable X, the variance of X is defined as,

$Var(X) = \sigma_X^2 = E[(X - \overline{X})] = E(X^2) - 2E(X)\overline{X} + \overline{X^2} = E(X^2) - \overline{X^2}$

Let X and Y be two random variables with expected values $\overline{X}$ and $\overline{Y}$. The covariance is defined as,

$Cov(X,Y) = \sigma_{XY} = \sigma_{YX} = E[(X - \overline{X})(Y - \overline{Y})]$

Alternative formula,

$Cov(X,Y) = E(XY) - \overline{XY}$

Note: The variance of a random variable is the covariance of that variable with itself.

If two variables X and Y are independent then they are uncorrelated and $\sigma_{XY} = 0$

The correlation coefficient of two random variables is defined as,

$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$

We have covered the most basic probability concepts. Now we turn our attention to portfolio mean and variance.

Suppose that we have n assets with random rates of return $r_1, r_2, ..., r_n$ with expected values $E(r_1) = \overline{r_1}, E(r_2) = \overline{r_2}, ..., E(r_n) = \overline{r_n}$ and corresponding weights $w_i, i = 1, 2, ..., n$. Then, the rate of return of portfolio is,

$r = \omega_1 r_1 + \omega_2 r_2 + ... + \omega_n r_n$

Taking expectation on both sides and using the property of linearity we have,

$E(r) = \omega_1 E(r_1) + \omega_2 E(r_2) + ... + \omega_n E(r_n)$

Now, let us denote the variance of the return of portfolio by σ^2, variance of return of asset i by σ_i^2 and the covariance of the return of asset i with asset j by σ_{ij}. Then, the variance of the rate of return of the portfolio is,

$\sigma^2 = E[(r - \overline{r})^2] = E[(\sum_{i=1}^n \omega_i r_i - \sum_{i=1}^n \omega_i \overline{r_i})^2] = E[(\sum_{i=1}^n \omega_i (r_i - \overline{r_i})) - (\sum_{j=1}^n \omega_j (r_j - \overline{r_j}))] = E[\sum_{i,j=1}^n \omega_i \omega_j (r_i - \overline{r_i})(r_j - \overline{r_j})] = \sum_{i,j=1}^n \omega_i \omega_j \sigma_{ij}$

Most of the basic concepts have been covered so far. Now we turn our focus on the Markowitz model. It is the optimization model which assists in the selection of the most efficient portfolio (minimum-variance portfolios) by analyzing possible portfolios of the given securities.

Assume that we have n assets with expected rates of return $E(r_1) = \overline{r_1}, E(r_2) = \overline{r_2}, ..., E(r_n) = \overline{r_n}$, the covariances σ_{ij} for $i, j = 1, 2, ..., n$ and sum of set of n weights ω_i, $i = 1, 2, ...n$ equals to 1. In order to find a minimum-variance portfolio, we fix the mean value at some arbitrary value $\overline{r}$. Then we find the feasible portfolio of minimum variance that has this mean. Mathematically,

minimize $\frac{1}{2} \sum_{i,j=1}^n \omega_i \omega_j \sigma_{ij}$

subject to $\sum_{i=1}^n \omega_i \overline{r_i} = \overline{r}$

and $\sum_{i=1}^n \omega_i = 1$

So far, we have done a lot of mathematical heavylifting. Now let us apply above concepts. Let us select FAANG stocks to build our portfolio.

Let us first load the required packages from our R library

Let's download the prices for our stocks

```
start_Date = "2015-01-01"
end_Date = "2020-12-31"
symbols = c("FB","AAPL","AMZN","NFLX","GOOG")
stock_prices = tq_get(symbols,from=start_Date,to=end_Date) %>%
  select(symbol,date,adjusted)%>% ## selecting only adjusted values
  pivot_wider(names_from = symbol,values_from = adjusted) %>%
  tk_xts(date_var=date)    ## Converting data into times series format
```

```
## Warning: Non-numeric columns being dropped: date
```

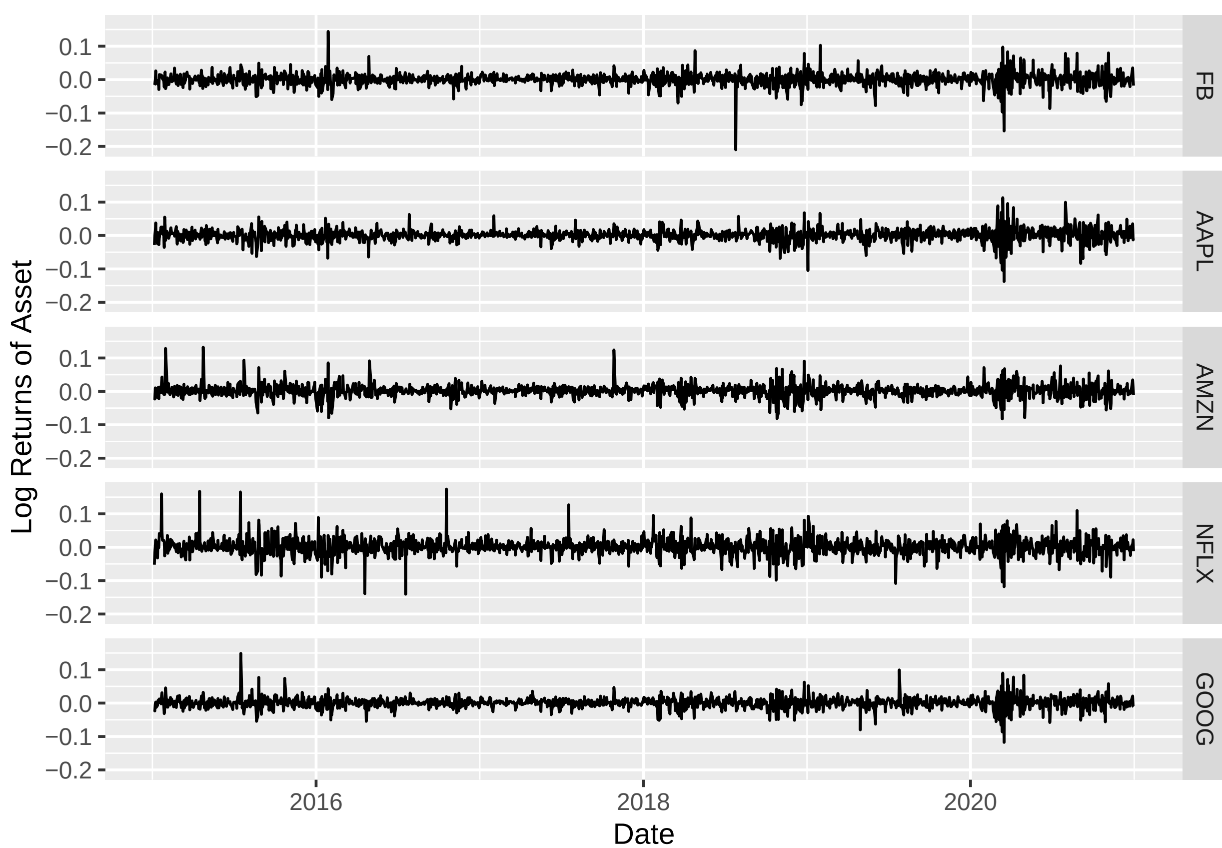
Calculating FAANG daily log returns

```
FAANG_Returns = diff(log(stock_prices),na.pad = FALSE)
head(FAANG_Returns)
```

```
##                      FB          AAPL        AMZN         NFLX         GOOG
## 2015-01-05 -0.016191500 -2.857596e-02 -0.02073067 -0.052237977 -0.021066071
## 2015-01-06 -0.013564835  9.431544e-05 -0.02309801 -0.017268787 -0.023449862
## 2015-01-07  0.000000000  1.392464e-02  0.01054398  0.005178409 -0.001714733
## 2015-01-08  0.026308755  3.770266e-02  0.00681267  0.021945626  0.003148121
## 2015-01-09 -0.005643961  1.071732e-03 -0.01181821 -0.015578461 -0.013035197
## 2015-01-12 -0.013207456 -2.494925e-02 -0.01876517 -0.032280783 -0.007322608
```

Plotting log returns data for all assets in our portfolio

```
autoplot(FAANG_Returns) +
  xlab("Date")+
  ylab("Log Returns of Asset") +
  theme_get()
```



Calculating Mean daily returns for each asset

```
Mean_Return = colMeans(FAANG_Returns)
print(round(Mean_Return,5))
```

```
##      FB    AAPL    AMZN    NFLX    GOOG
## 0.00082 0.00111 0.00157 0.00156 0.00080
```

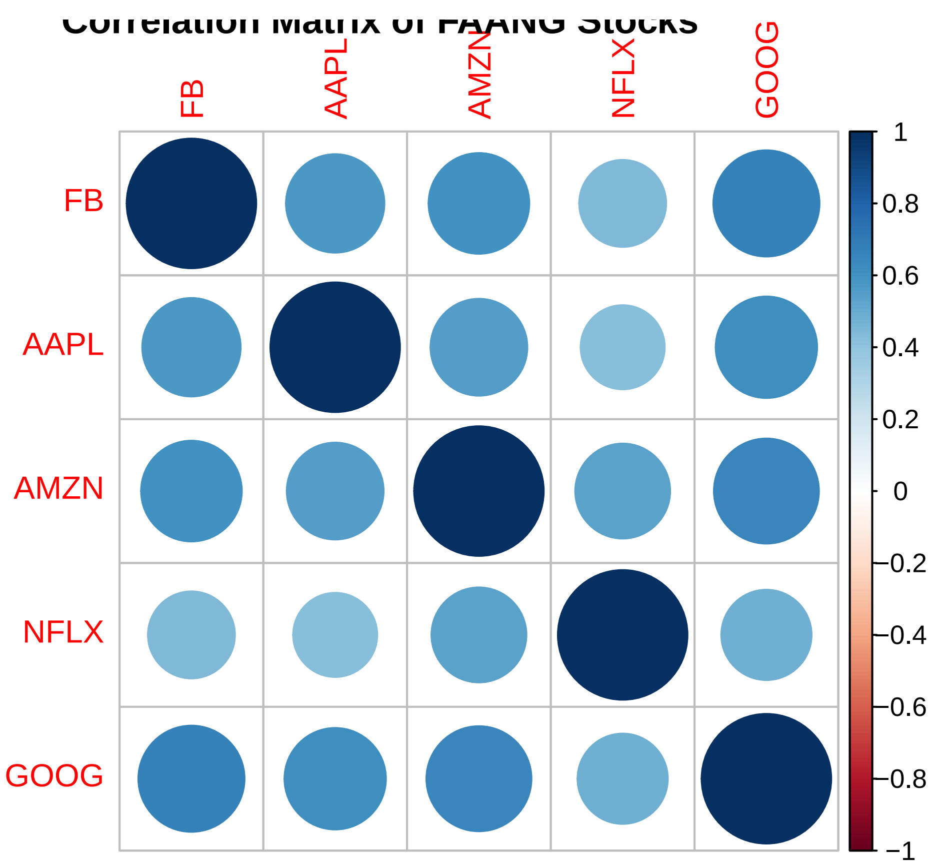
Now we will calculate the covariance matrix for our FAANG stocks and annualize it

```
Cov_Matrix = cov(FAANG_Returns) * 252   ## Annualizing
print(round(Cov_Matrix,5))
```

```
##           FB    AAPL    AMZN    NFLX    GOOG
## FB   0.10259 0.05477 0.05945 0.06069 0.05762
## AAPL 0.05477 0.08802 0.05093 0.05255 0.04874
## AMZN 0.05945 0.05093 0.09462 0.06944 0.05417
## NFLX 0.06069 0.05255 0.06944 0.17736 0.05477
## GOOG 0.05762 0.04874 0.05417 0.05477 0.07200
```

Visualizing the correlation matrix

```
corrplot(cor(FAANG_Returns),main="Correlation Matrix of FAANG Stocks")
```



Now let us assign random weights to each of our stocks in our portfolio.

```
Weights = runif(n=length(symbols))
Weights = Weights/sum(Weights)
print(Weights)
```

```
## [1] 0.22160913 0.12689430 0.23503586 0.39350852 0.02295219
```

```
print(sum(Weights))
```

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

Calculating annualized portfolio returns

```
FAANG_Portfolio_Returns = (sum(Weights*Mean_Return)+1)^252-1
FAANG_Portfolio_Returns
```

```
## [1] 0.3958726
```

Calculating annualized standard deviation

```
FAANG_Portfolio_Risk = sqrt(t(Weights) %*% (Cov_Matrix %*% Weights))
print(FAANG_Portfolio_Risk)
```

```
##             [,1]
## [1,] 0.2874325
```

Calculating sharpe ratio assuming risk free rate of interest of 0%

```
Sharpe_Ratio = FAANG_Portfolio_Returns/FAANG_Portfolio_Risk
print(Sharpe_Ratio)
```

```
##            [,1]
## [1,] 1.377272
```

We have calculated most of the basic requirements to perform our optimization. Lets do the simulation for 5000 random portfolios.

```
nSim = 5000
Weights_Stocks = matrix(nrow=nSim,ncol=length(symbols))   ## matrix for weights
FAANG_Portfolio_Returns = c()   ## empty vector to store returns
FAANG_Portfolio_Risk = c()   ## empty vector to store SD
Sharpe_Ratio = vector('numeric',length=nSim)

for(i in 1:5000){
  Weights = runif(length(symbols))
  Weights = Weights/sum(Weights)
  Weights_Stocks[i,] = Weights

  FAANG_Port_Returns = (sum(Weights*Mean_Return)+1)^252-1
  FAANG_Portfolio_Returns[i] = FAANG_Port_Returns
```

```
  FAANG_Port_Risk = sqrt(t(Weights) %*% (Cov_Matrix %*% Weights))
  FAANG_Portfolio_Risk[i] = FAANG_Port_Risk

  sr = FAANG_Port_Returns/FAANG_Port_Risk
  Sharpe_Ratio[i] = sr
}
```

Storing the values in a table and changing variable names

Combining all the values together

```
Portfolio_Values = tk_tbl(cbind(Weights_Stocks,Portfolio_Values))
```

```
## Warning in tk_tbl.data.frame(cbind(Weights_Stocks, Portfolio_Values)): Warning:
## No index to preserve. Object otherwise converted to tibble successfully.
```

```
head(Portfolio_Values)
```

```
## # A tibble: 6 x 8
##       FB    AAPL    AMZN   NFLX   GOOG Return  Risk SharpeRatio
##    <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <dbl>       <dbl>
## 1 0.141  0.110   0.361   0.0480 0.341   0.336 0.253        1.33
## 2 0.379  0.163   0.00724 0.319  0.131   0.322 0.277        1.16
## 3 0.495  0.00152 0.118   0.0505 0.336   0.267 0.267        1.00
## 4 0.225  0.176   0.136   0.122  0.341   0.305 0.252        1.21
## 5 0.0536 0.305   0.449   0.150  0.0425  0.407 0.264        1.54
## 6 0.208  0.163   0.189   0.383  0.0572  0.384 0.283        1.36
```

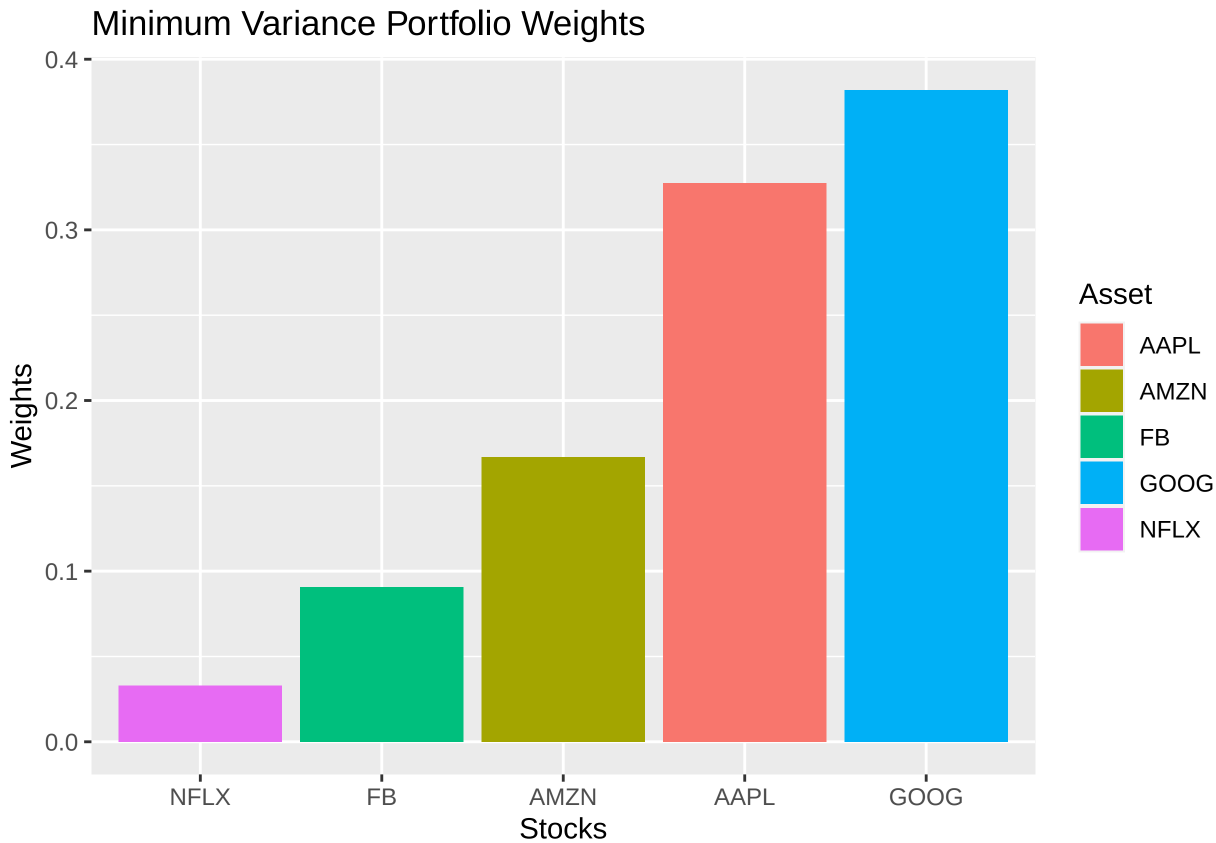
Now we can find our minimum variance portfolio

```
Minimum_Variance_Portfolio = Portfolio_Values[which.min(Portfolio_Values$Risk),]
Minimum_Variance_Portfolio = as.data.frame(Minimum_Variance_Portfolio)
Minimum_Variance_Portfolio
```

```
##           FB      AAPL      AMZN       NFLX      GOOG    Return      Risk
## 1 0.09073354 0.3274899 0.1668781 0.03282961 0.3820689 0.3048463 0.2469239
##   SharpeRatio
## 1    1.234576
```

Let’s visualize the weights allocated to each asset according to minimum portfolio variance

```
X = Minimum_Variance_Portfolio %>%
  gather(FB:GOOG, key = Asset, value = Weights) %>%
  mutate(Asset = as.factor(Asset)) %>%
  ggplot(aes(x=fct_reorder(Asset,Weights),y=Weights,fill=Asset))+
  geom_bar(stat = 'identity') +
  theme()+
  labs(x="Stocks", y= "Weights", title = "Minimum Variance Portfolio Weights")
plot((X))
```

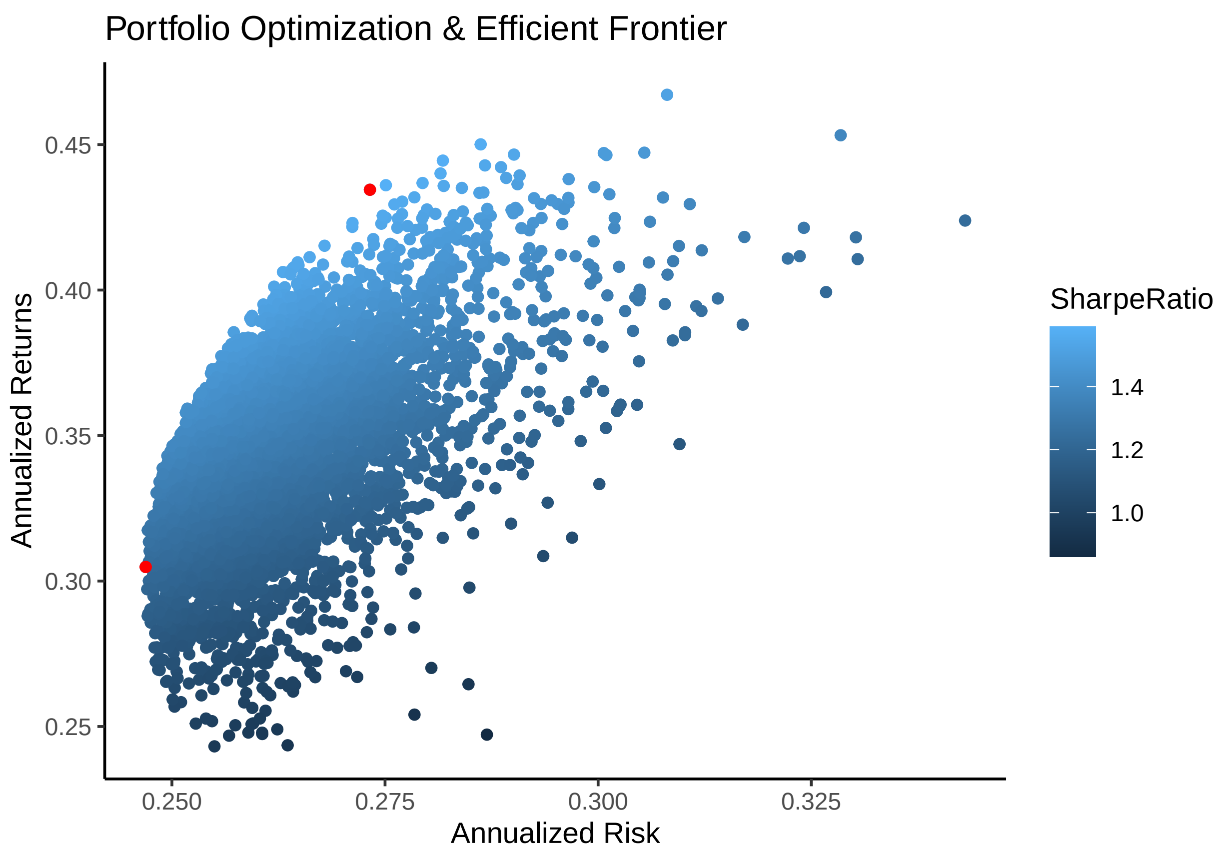


Finding the portfolio with highest sharpe ratio also known as tangent portfolio

```
Max_SharpeRatio = Portfolio_Values[which.max(Portfolio_Values$SharpeRatio),]
```

Plotting all the random portfolios and visualizing the efficient frontier

```
p <- Portfolio_Values %>%
  ggplot(aes(x = Risk, y = Return, color = SharpeRatio)) +
  geom_point() +
  theme_classic() +
  scale_y_continuous() +
  scale_x_continuous() +
  labs(x = 'Annualized Risk',
       y = 'Annualized Returns',
       title = "Portfolio Optimization & Efficient Frontier") +
  geom_point(aes(x = Risk,
                 y = Return), data = Minimum_Variance_Portfolio, color = 'red') +
  geom_point(aes(x = Risk,
                 y = Return), data = Max_SharpeRatio, color = 'red')
plot(p)
```


Portfolio Optimization & Efficient Frontier
Annualized Returns
0.45
0.40
0.35
0.30
0.25
0.250
0.275
0.300
0.325
Annualized Risk
SharpeRatio
1.4
1.2
1.0