Creating an Optimal Investment Strategy using Modern Portfolio Theory

Edric Franco

June 2020

Contents

1) Introduction

In this project we will apply Markowitz model otherwise known as Modern Portfolio Theory or Mean Variance Analysis to 30 randomly chosen stocks from the S&P 500. The model states that the optimal portfolio is the one that maximizes the Sharpe Ratio. We will optimize our strategy using the 2010-2018 dataset and we will validate it on the 2018-2020 dataset.

Sharpe Ratio is the excess Return per Unit Risk. In Mean-Variance Analysis, we measure risk as variance which we want to minimize and the mean of the Returns would be the one we want to maximize.

We will Calculate the Efficient Frontier by finding the minimium variance for a given return and find the Optimal portfolio by maximizing the Sharpe Ratio.

Finally, we will compare the Returns of our Optimal Portfolio to the Returns of S&P 500 by applying Monte Carlo Simulation to estimate the true mean return and the Max Drawdown (i.e the largest decline from a historical peak). We then compare the historical Value-at-Risk (VaR) / Conditional-Value-at-Risk (VaR) of the Optimal Portfolio returns and the S&P 500 returns.

Stock Tickers Used:

'MMM', 'AXP','AAPL','BA','CAT','CVX','CSCO', 'KO','XOM',
'GS', 'HD', 'IBM', 'INTC','JNJ','JPM','MCD','MRK','MSFT',
'NKE','PFE','PG', 'TRV','UNH','GOOGL','VZ','V','WMT','FFIV','DIS'

2) Data

2.1) Gathering Data

Let us Gather the data from January 2010 to June 2020 using get.hist.quote function in R.

Gathering the monthly prices of 30 random stocks from S&P 500:

```
ticker.symbols= c("MMM", "AXP","AAPL","BA","CAT","CVX","CSCO","KO","XOM","GS",
                  "HD","IBM","INTC","JNJ","JPM","MCD","MRK","MSFT","NKE","PFE","PG",
                  "TRV","UNH","GOOGL","VZ","V","WMT","FFIV","DIS")

N.stocks = length(ticker.symbols)

start.date=as.Date("2010-01-01")
end.date=as.Date("2020-07-01")
P=get.hist.quote(instrument = ticker.symbols[1],
                 start = start.date, end=end.date,
                 quote = "AdjClose", retclass = "zoo",
                 quiet = T, compression='m')
names(P)=ticker.symbols[1]

for (i in 2:N.stocks) {
  x=try(get.hist.quote(instrument = ticker.symbols[i],
                       start = start.date, end=end.date,
                       quote = "AdjClose", retclass = "zoo",
                       quiet = T, compression='m'))
  if( class(x)!= "try-error"){
    names(x)=ticker.symbols[i]
    P = merge(P, x, all=TRUE)
  }
}
```

Getting monthly prices of S&P 500 (ticker = ^GSPC):

```
Pm = get.hist.quote(instrument = "^GSPC",
                    start = start.date, end=end.date,
                    quote = "AdjClose", retclass = "zoo",
                    quiet = T, compression='m')
```

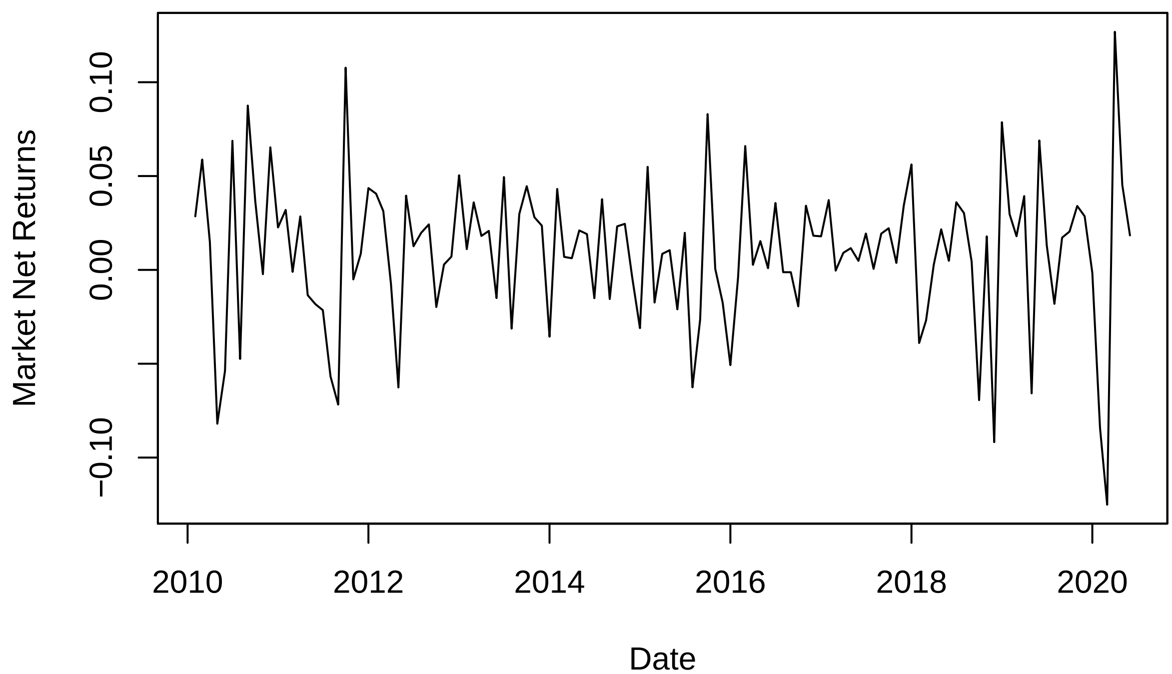
2.2) Net and Log Returns

Calculating Net Returns and Log Returns of S&P 500 and the 30 chosen stocks:

```
r = diff(log(P)) # Log returns of 30 stocks
R = exp(r)-1  # Net returns of 30 stocks
rm = diff(log(Pm)) # Log return of Market
Rm = exp(rm)-1 # Net return of Market (S&P 500)
Rf = 0.02 # Assuming Risk Free rate of 2 percent
```

The monthly market returns for the S&P 500 ranges from around -0.1 to 0.1 %. (See figure below.)

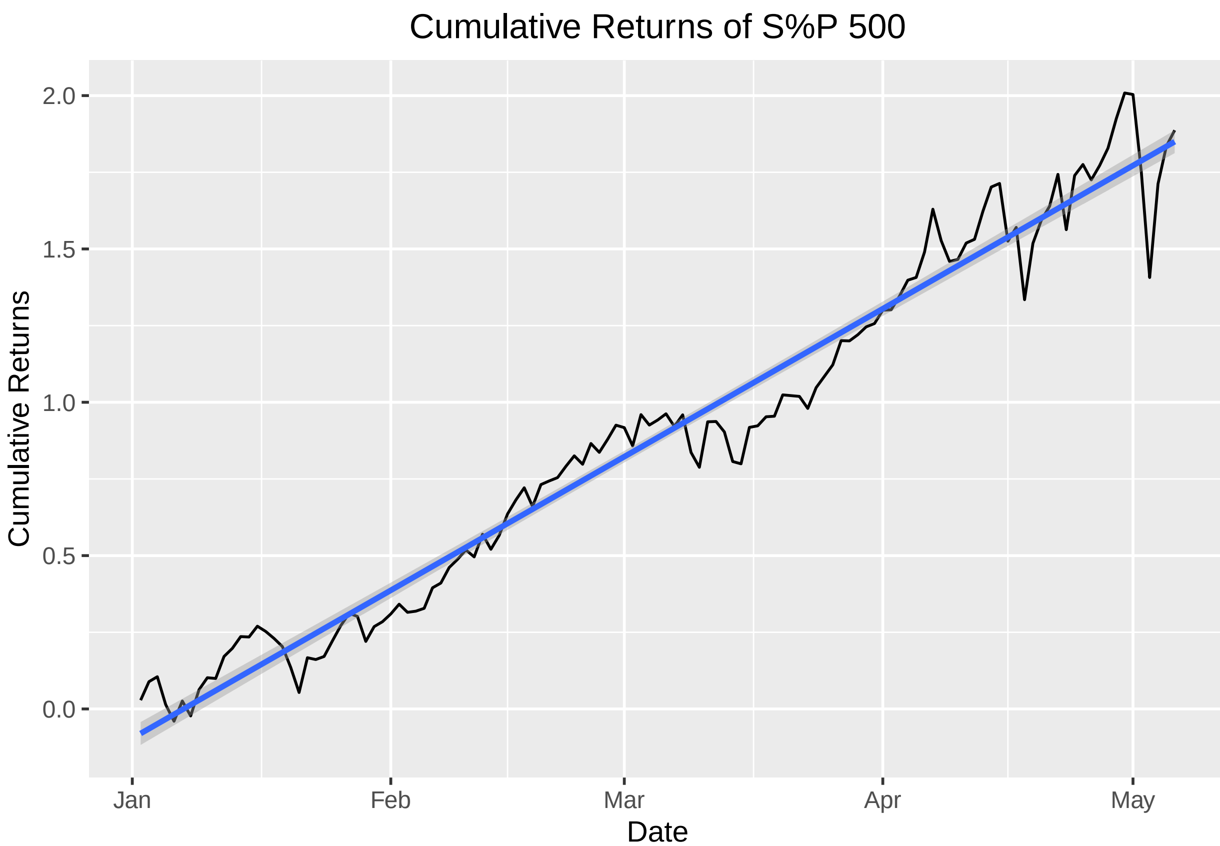
```
plot(Rm, ylab ='Market Net Returns', xlab= 'Date')
```



The returns of S&P 500 increases in a steady pace, with around a 200% net return from 2010-2020. That is, if we invested 100 dollars in 2010 at the S&P 500, it would now be valued around 300 dollars. (i.e 100*(1+R))

```
comb.ret=function(x){cumprod(1+x)-1} # function to compound net returns
cum_ret = apply(Rm, MARGIN=2, FUN=comb.ret)
data.frame(cum_ret) %>%
  mutate(Date=as.Date(index(cum_ret)))%>%
  ggplot(aes(x=Date, y=Adjusted)) +
  geom_line() +
  geom_smooth(method='lm') +
  ylab('Cumulative Returns') +
```

```
ggtitle('Cumulative Returns of S%P 500') +
theme(plot.title = element_text(hjust = 0.5))
```



2.3) Splitting Data for Out-of-Sample Testing

We split our data from Febuary 2018 onwards for out-of-sample testing later on and we would not use this when optimizing our portfolio to detect overfitting.

```
R_test = window(R, start=as.Date('2018-02-01'))
R = window(R, start=as.Date("2010-01-01"), end=as.Date("2018-01-01"))

Rm_test = window(Rm, start=as.Date('2018-02-01'))
Rm = window(Rm, start=as.Date("2010-01-01"), end=as.Date("2018-01-01"))
```

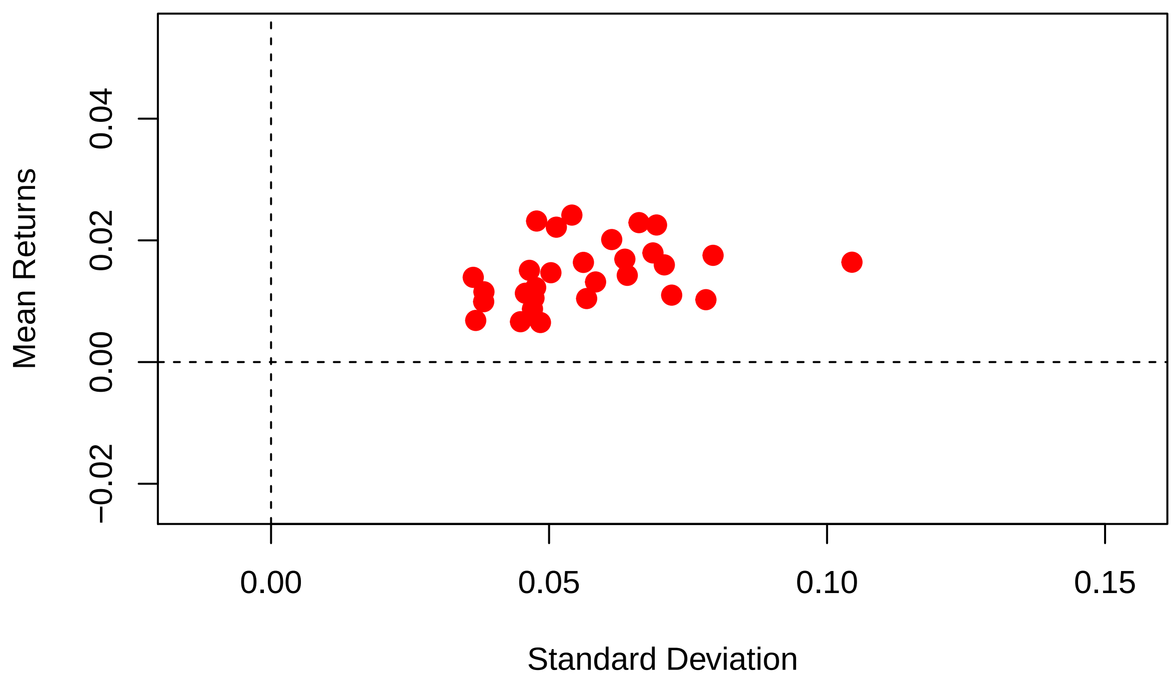
3) Methods

3.1) Optimization

To find the optimal portfolio, we will minimize variance and maximize mean return.

The plot below indicates the mean returns and standard deviations of our 30 stocks:

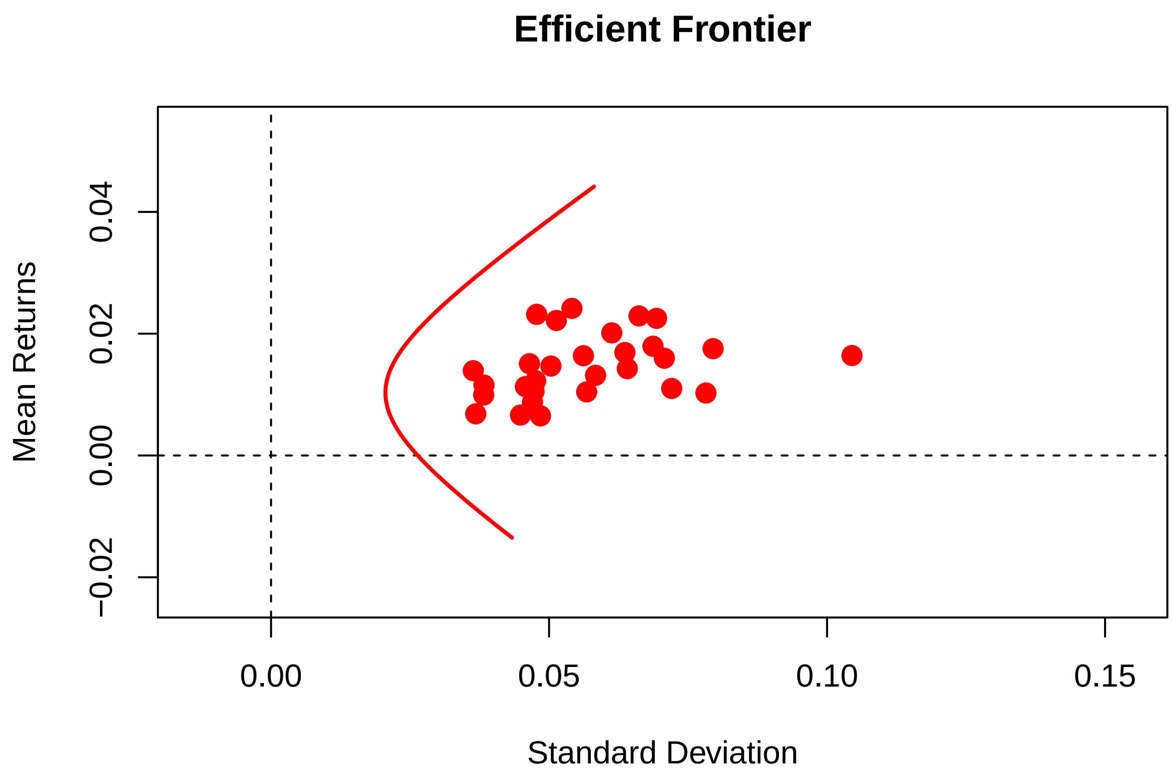
```r
COV = cov(R) # Variance - Covariance Matrix
MU = colMeans(R) # Mean Returns
SD = sqrt(diag(COV)) # Standard Deviation of Returns
N=dim(R)[2] # Number of tickers
plot(SD,MU,pch=20,cex=2, col=2, xlim=c(min(SD)-0.05,max(SD)+0.05),
     ylim=c(min(MU)-0.03, max(MU)+0.03),
     xlab='Standard Deviation', ylab='Mean Returns')
abline(v=0, lty=2); abline(h=0, lty=2)
```



Let us now calculate the efficient frontier. That is, we calculate the minimum variance for a given mean via Quadratic Optimization. From the figure below, the red curve indicates the efficient frontier (i.e the minimum standard deviation for a given mean return)

```r
Amat = cbind(rep(1, N), MU)
mu.p = sd.p = seq( min(MU)-0.02, max(MU)+0.02,length=100)

pd_D_mat <- nearPD(COV) # Convert to a positive definite matrix
```

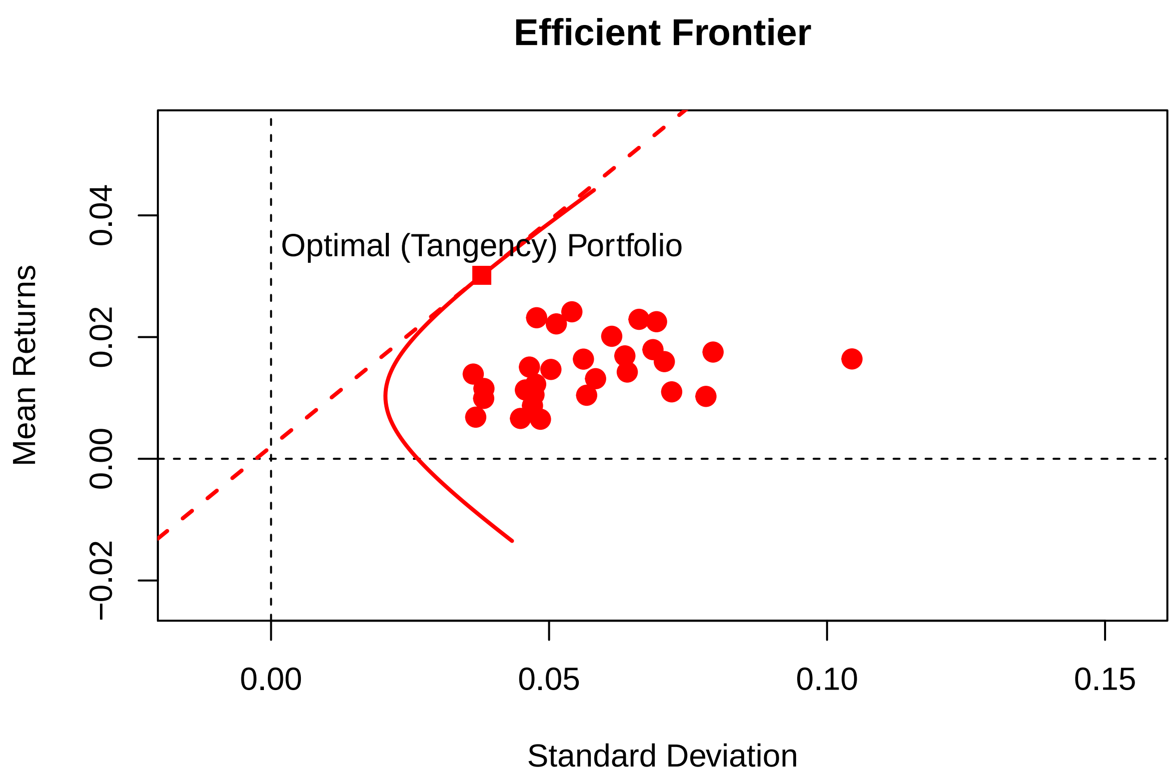
```r
for (i in 1:length(mu.p)){
  bvec=c(1,mu.p[i])

  out=solve.QP(Dmat=2*as.matrix(pd_D_mat$mat),
               dvec=rep(0, N),
               Amat=Amat,
               bvec=bvec,
               meq=2)

  sd.p[i] = sqrt(out$value)
}

plot(SD,MU,pch=20,cex=2, col=2, xlim=c(min(SD)-0.05,max(SD)+0.05),
     ylim=c(min(MU)-0.03, max(MU)+0.03),
     xlab='Standard Deviation', ylab='Mean Returns', main='Efficient Frontier')
abline(v=0, lty=2); abline(h=0, lty=2)
lines(sd.p,mu.p,type="l", lwd=2, col=2) # plot least variance portfolios
```



Assuming a risk-free Monthly interest rate of 0.2%, Let us calculate the tangency portfolio weights and plot the sharpe ratio. The Sharpe Ratio indicates the excess return per unit risk (or the change in mean return per unit increase in variance).

```r
mu.f = 0.002

COV.i=solve(COV)
W.tang=COV.i%*%(MU-mu.f) / sum( COV.i%*%(MU-mu.f) )
```

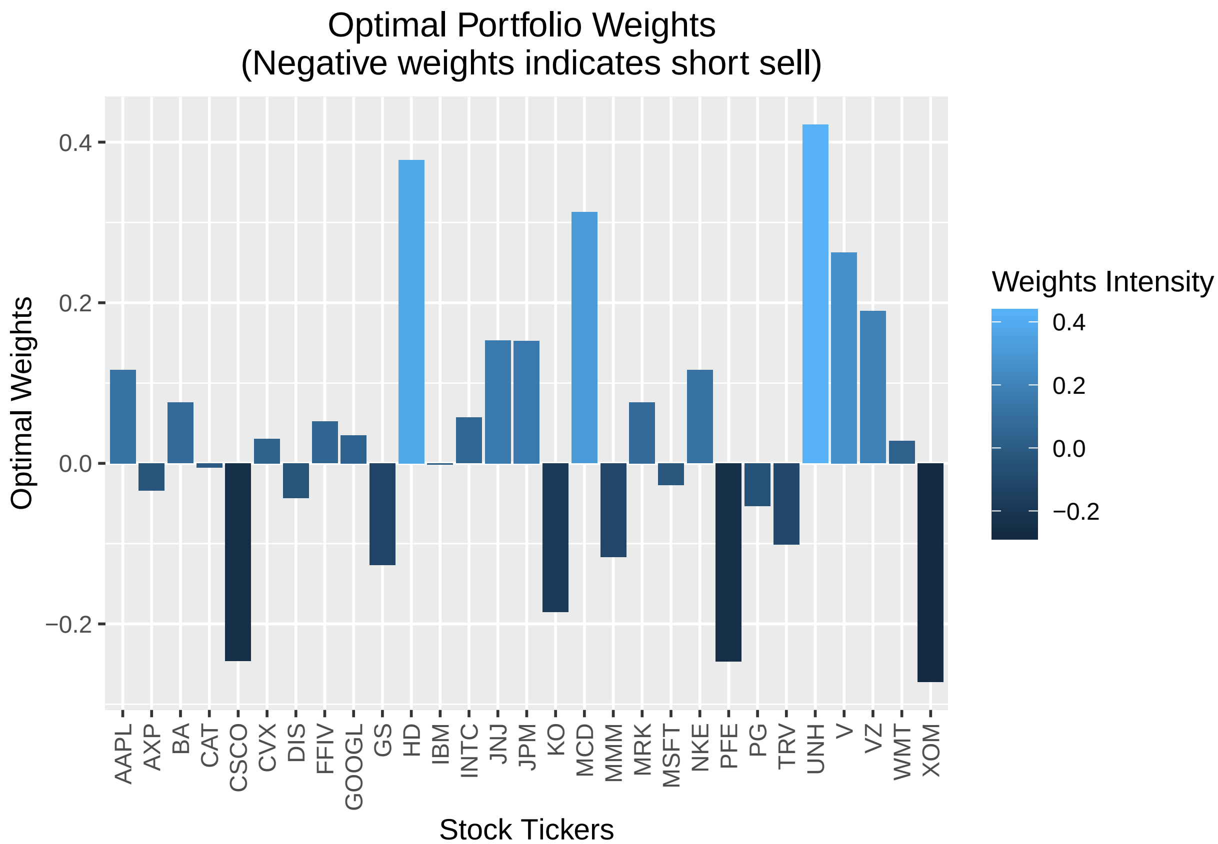
```r
mu.tang=sum(W.tang*MU)
sd.tang=sqrt(sum( (COV %*% W.tang) * W.tang ) )

plot(SD,MU,pch=20,cex=2, col=2, xlim=c(min(SD)-0.05,max(SD)+0.05),
     ylim=c(min(MU)-0.03, max(MU)+0.03),
     xlab='Standard Deviation', ylab='Mean Returns', main='Efficient Frontier')
abline(v=0, lty=2); abline(h=0, lty=2)
lines(sd.p,mu.p,type="l", lwd=2, col=2) # plot least variance portfolios
points( sd.tang, mu.tang, pch=15, cex=1.3, col=2)
sharpe=(mu.tang-mu.f)/sd.tang # Sharpe Ratio
abline(mu.f,sharpe,lwd=2,lty=2,col=2)
text(sd.tang, mu.tang, c('Optimal (Tangency) Portfolio'), pos=3)
```



The optimal portfolio weights would be the tangency portfolio or the one that maximizes the Sharpe Ratio. We can see the optimal weights below, negative weights indicate that we should short sell that stock. The sum of these portfolio weights should equal to 1.

```r
data.frame(W.tang) %>%
  mutate(tickers=ticker.symbols) %>%
  ggplot(aes(x=tickers, y=W.tang, fill=W.tang)) +
  geom_bar(stat='identity') +
  ylab('Optimal Weights') +
  xlab('Stock Tickers') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_fill_continuous(name = "Weights Intensity") +
```

```r
ggtitle('Optimal Portfolio Weights \n (Negative weights indicates short sell)') +
theme(plot.title = element_text(hjust = 0.5))
```



3.2) Alpha and Beta

Let us now calculate the Returns of our optimal portfolio. Assuming a risk-free monthly rate of 0.2%, Let us calculate the Alpha and Beta of our trading strategy. Alpha indicates how much our trading portfolio outperforms the market (S&P 500) and Beta indicates the strength of the correlation between our trading portfolio and the market (S&P 500). (i.e: Returns of our portfolio = Alpha + Beta * (Returns of market))

```r
# Returns using optimal weights
R.p = R %*% W.tang
Y = R.p - mu.f
X = Rm - mu.f
out = lm(Y~X)
(beta = out$coef[2])
```

```
##         X
## 0.5357715
```

```r
(alpha = out$coef[1])
```

```
## (Intercept)
##  0.02347774
```

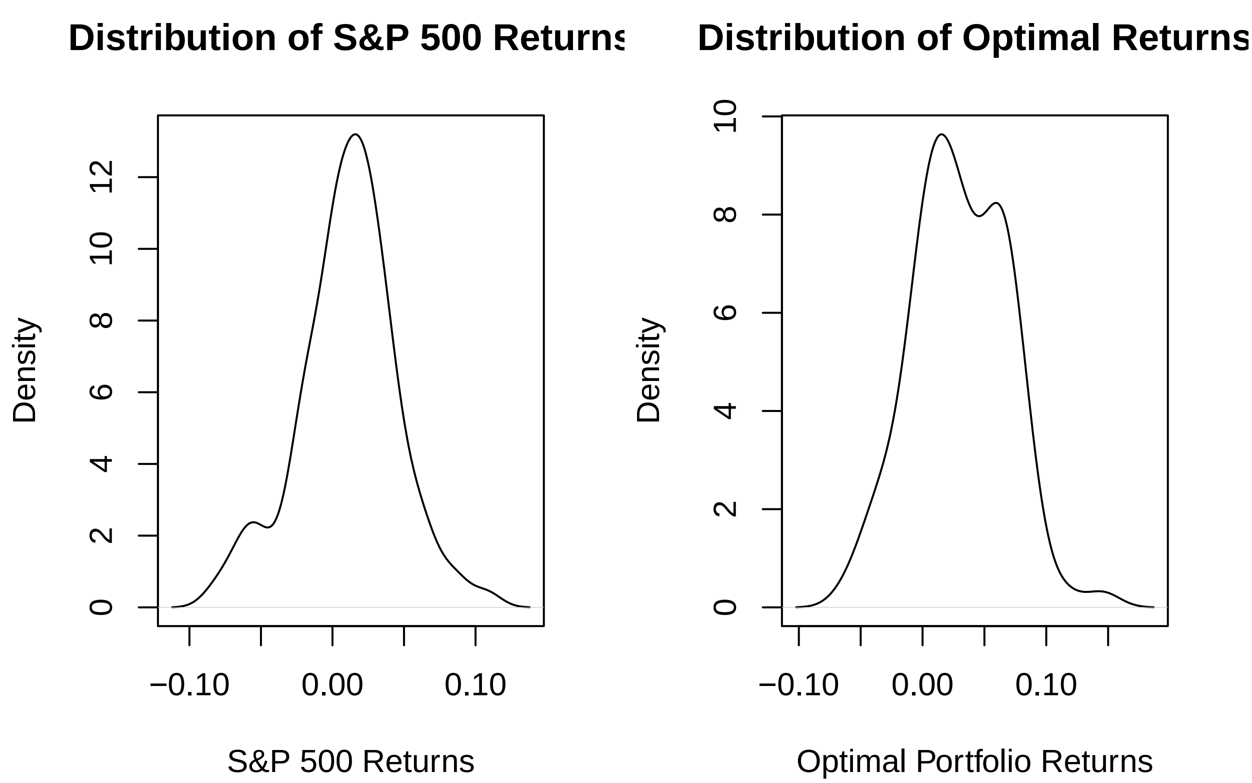
We can see that the Alpha of our strategy is positive (0.0235) which means our strategy outperforms the market and the Beta of our strategy is 0.535 which means our strategy does not have a high correlation with the market.

4) Evaluation

4.1) Monte Carlo Estimation of Returns

We can see from the plot below that the distribution of both the S&P 500 and Optimal Returns have approximately heavy tails (extreme values). It appears good to fit a student-t distribution and conduct a Monte Carlo simulation to estimate the true mean returns.

```
par(mfrow=c(1,2))
plot(density(Rm), xlab='S&P 500 Returns', ylab='Density', main='Distribution of S&P 500 Returns')
plot(density(R.p), xlab = 'Optimal Portfolio Returns', ylab='Density', main='Distribution of Optimal Ret
```



Let us fit a Student-t distribution to the Returns:

```
params=matrix(0,2,3) # matrix of parameters
params[1,1:3] = fitdistr(R.p, 't')$est # parameters for optimal portfolio returns
params[2,1:3] = fitdistr(Rm, 't')$est # parameters for S&P 500 returns
colnames(params) = c("mu","sigma","df")
rownames(params) = c('opt', 's&p')
params
```

```
##              mu      sigma        df
## opt 0.03005265 0.03627518 22.684173
## s&p 0.01159961 0.02917960  6.737764
```

Let us now conduct Monte Carlo Simluation to estimate the true mean Returns and maxdrawdown of both Returns. We can see from below that the estimated monthly mean return of our optimal portfolio is 0.03 with a 95% Confidence interval of [-0.04, 0.104]. The estimated mean return of S&P 500 is 0.0115 with a 95% Confidence interval of [-0.0567, 0.0798]. We can see that the optimal portfolio has a larger estimate of the mean than the S&P 500.

```r
N = 5000 # Number of simulations

comb.ret=function(x){cumprod(1+x)-1} # function to compound net returns
myMDD=function(x)(maxdrawdown(x)$maxdrawdown) # function returning only maximum drawdown value

## Optimal Portfolio Returns
Rsim.opt=params[1,1]+params[1,2]*rt(N*252,params[1,3]) # generate daily iid t returns
Rsim.opt=matrix(Rsim.opt, nrow=252, ncol=N )
cum.Rsim.opt=apply(X=Rsim.opt, MARGIN=2, FUN=comb.ret)
MDD.opt=apply(cum.Rsim.opt, 2, myMDD)
(mean_opt = mean(Rsim.opt))
```

```
## [1] 0.03009538
```

```r
se_opt = sd(Rsim.opt)
(ci_opt = mean_opt + c(-1, 1) * qnorm(0.975) * se_opt)
```

```
## [1] -0.04447436  0.10466512
```

```r
## S&P 500 Returns
Rsim.snp=params[2,1]+params[2,2]*rt(N*252,params[2,3]) # generate daily iid t returns
Rsim.snp=matrix(Rsim.snp, nrow=252, ncol=N )
cum.Rsim.snp=apply(X=Rsim.snp, MARGIN=2, FUN=comb.ret)
MDD.snp=apply(cum.Rsim.snp, 2, myMDD)
(mean_snp = mean(Rsim.snp))
```

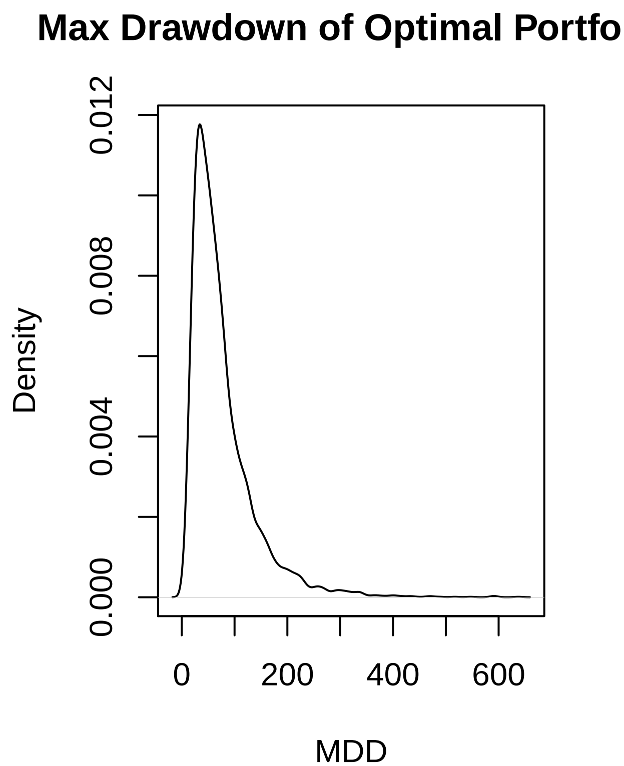
```
## [1] 0.01155559
```

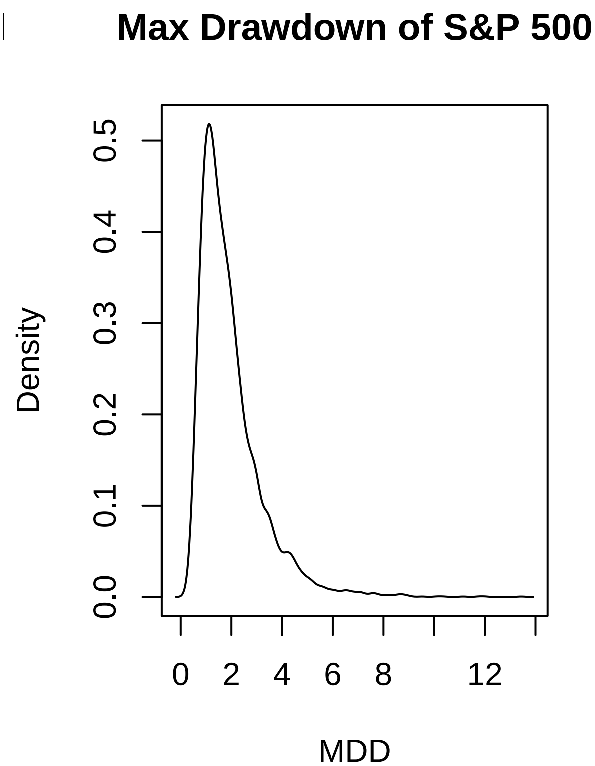
```r
se_snp = sd(Rsim.snp)
(ci_snp = mean_snp + c(-1, 1) * qnorm(0.975) * se_snp)
```

```
## [1] -0.05660588  0.07971705
```

Max Drawdown is defined as largest decline from a historical peak, we can see that both Returns are stable with majority of the Max Drawdowns in the simulation are in the lower tail.

```r
par(mfrow=c(1, 2))
plot(density(MDD.opt), xlab='MDD', main='Max Drawdown of Optimal Portfolio')
plot(density(MDD.snp), xlab='MDD', main='Max Drawdown of S&P 500')
```





4.2) Value-at-Risk (VaR) / Conditional-Value-at-Risk (CVaR)

Value at Risk is defined as the value such that the probability we lose more than that value is less than the significance level. (I.e Worst Case returns of our Investment with a given probability)

Meanwhile, Conditional Value at Risk is simply the average worst case returns assuming our Investment is worse than the Value at Risk (VaR)

Let us calculate the historical VaR and CVaR of S&P 500 and our Optimal Portfolio Returns at a 5% significance level.

Optimal Portfolio

- The Optimal Portfolio has a VaR of 0.033 and CVaR of 0.045

```
sig = 0.05
VaR.opt = -quantile(R.p, sig) # 0.0339
CVaR.opt = -mean(R.p[which(R.p<(-VaR.opt))]) # 0.045
```

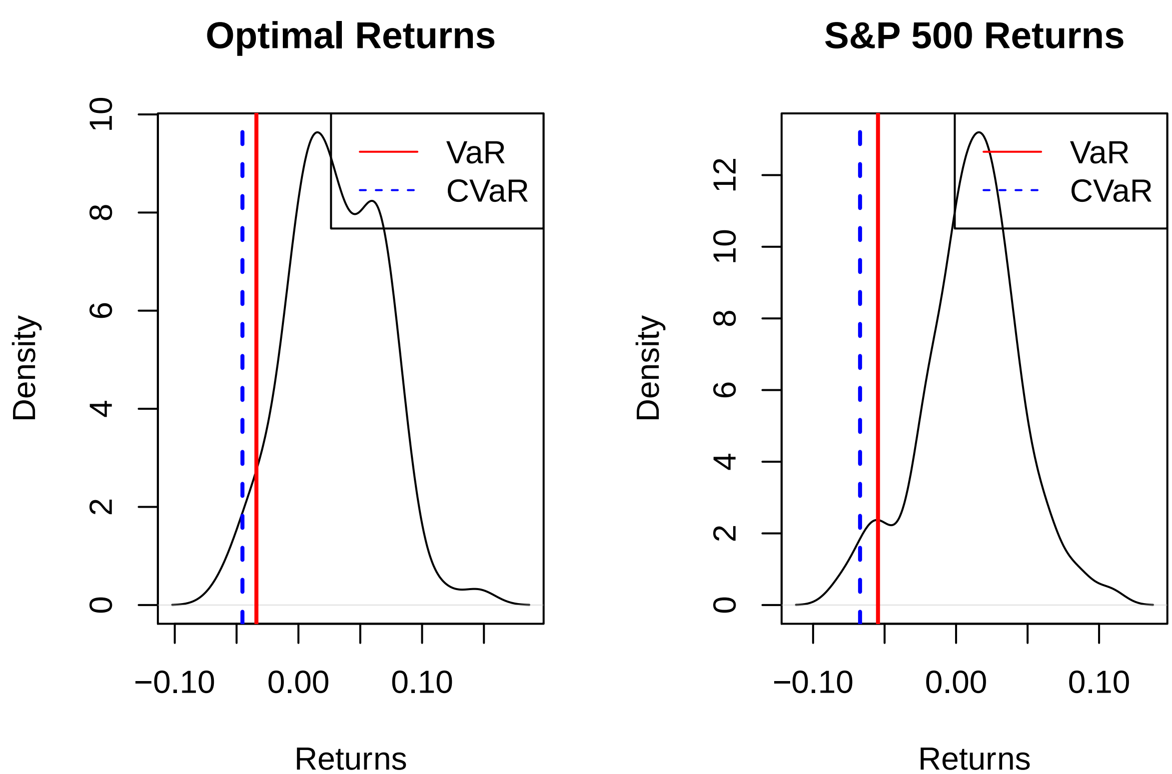
S&P 500

- The S&P 500 Investment has a VaR of 0.054 and CVaR of 0.067

```r
VaR.snp = -quantile(Rm, sig) # 0.0546
CVaR.snp = -mean(Rm[which(Rm<(-VaR.snp))]) # 0.067
```

We can see that the VaR and CVaR are both lower for our Optimal Portfolio which indicates that the worst case behaviour of the Optimal Portfolio is still better than the S&P 500

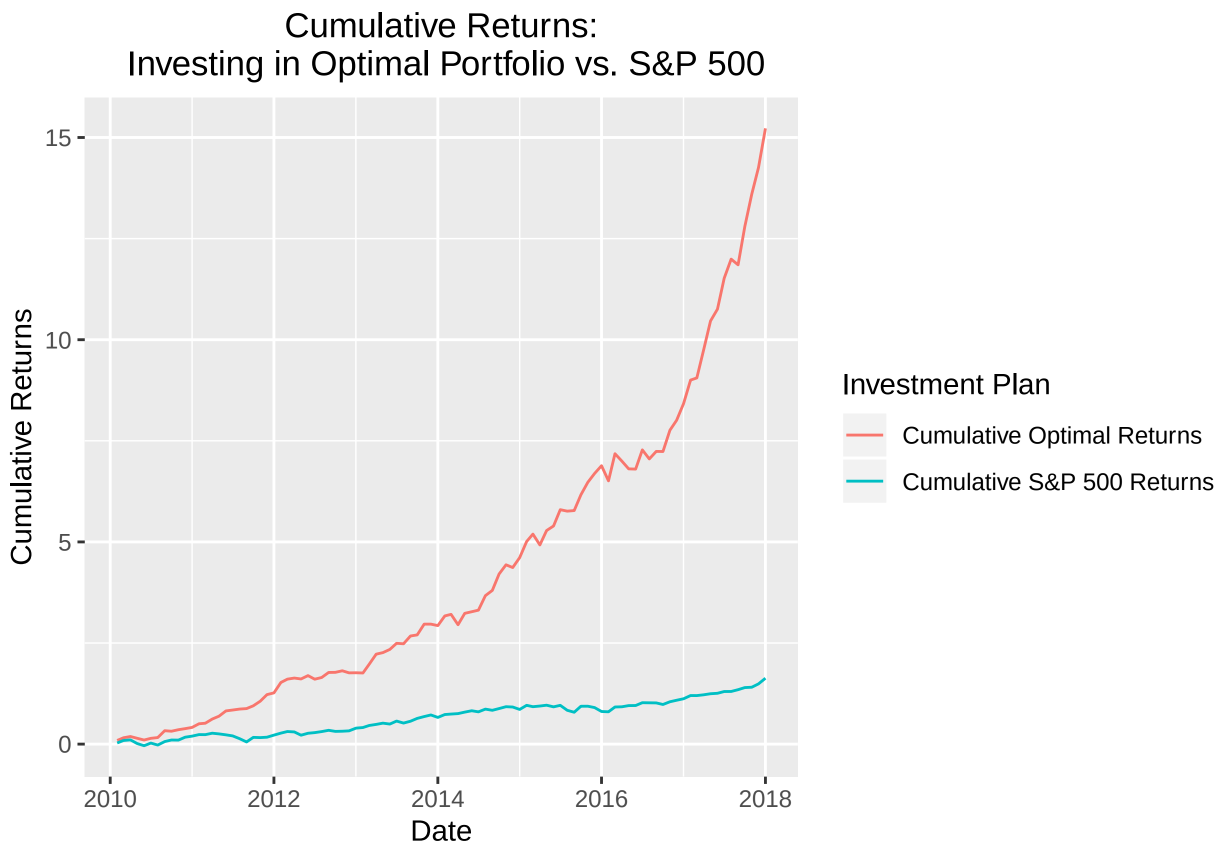
```r
par(mfrow=c(1,2))
plot(density(R.p), xlab='Returns', ylab='Density', main='Optimal Returns')
abline(v=-VaR.opt, col='red', lty=1, lwd=2)
abline(v=-CVaR.opt, col='blue', lty=2, lwd=2)
legend(x='topright',legend=c('VaR','CVaR'), col=c('red','blue'), lty=1:2)

plot(density(Rm), xlab='Returns', ylab='Density', main='S&P 500 Returns')
abline(v=-VaR.snp, col='red', lty=1, lwd=2)
abline(v=-CVaR.snp, col='blue', lty=2, lwd=2)
legend(x='topright',legend=c('VaR','CVaR'), col=c('red','blue'), lty=1:2)
```



4.3) Comparison of Cumulative Returns

By using the Optimal Trading Strategy, the cumulative Returns from 2010-2020 would be 6700 %. That is, if we invest 100 dollars in this Trading Strategy in 2010, we would have 6800 dollars by year 2020 compared to only 300 dollars if we invested in S&P 500. (i.e: 100 * (1+67))

```r
cum_ret_snp = apply(Rm, MARGIN=2, FUN=comb.ret)
cumulative_opt_returns = apply(R.p, MARGIN=2, FUN=comb.ret)
colnames(cum_ret_snp) = 'cumulative_SnP_Returns'
colnames(cumulative_opt_returns) = 'cumulative_Opt_Returns'

data.frame(cumulative_opt_returns) %>%
  mutate(Date = as.Date(index(Rm)), Cumulative_SnP_Returns = cum_ret_snp) %>%
  pivot_longer(c(1,3), names_to='Strategy', values_to='Cumulative_Returns') %>%
  ggplot(aes(x=Date, y=Cumulative_Returns, color=Strategy)) +
  ylab('Cumulative Returns') +
  geom_line() +
  scale_color_discrete(name = 'Investment Plan', labels=c('Cumulative Optimal Returns', 'Cumulative S&P
  ggtitle('Cumulative Returns:\n Investing in Optimal Portfolio vs. S&P 500') +
  theme(plot.title = element_text(hjust = 0.5))
```

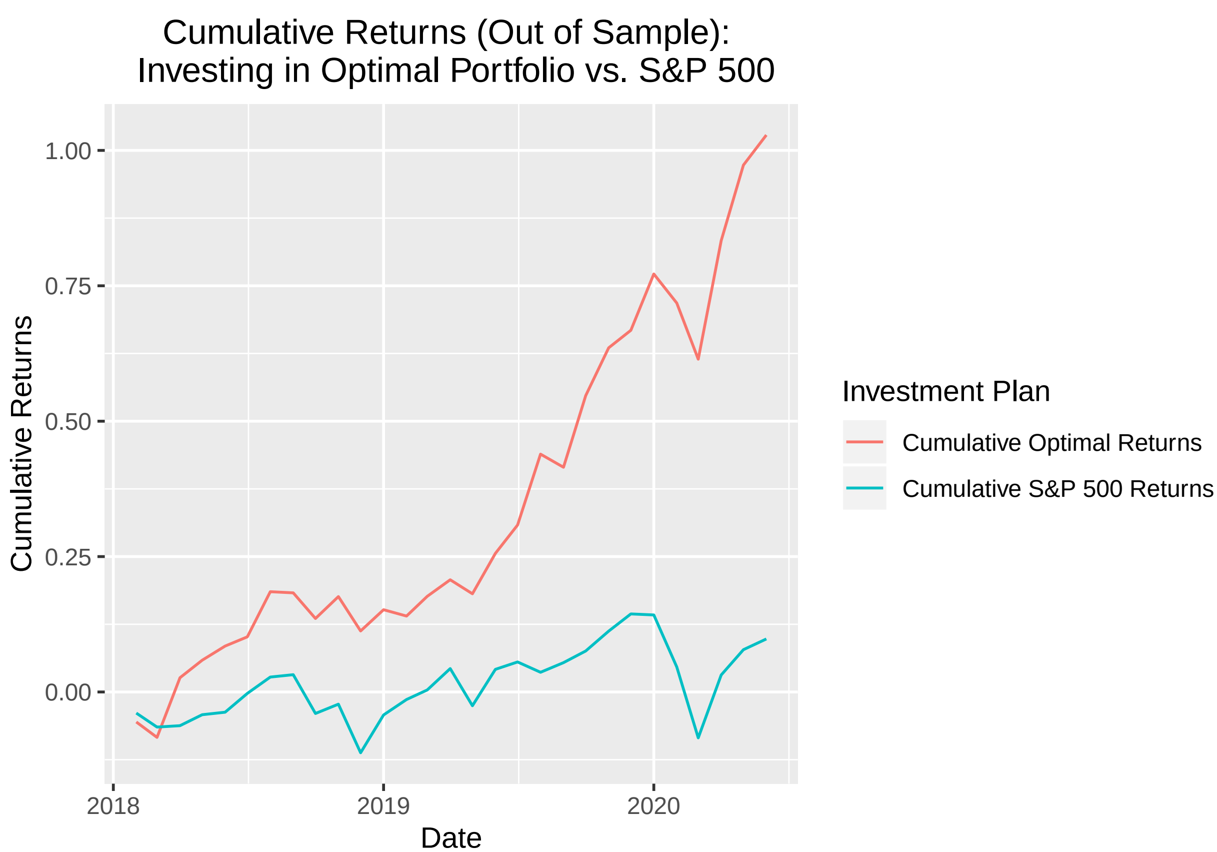


4.4) Out of Sample Validation

We now test our trading strategy to the 2018-2020 dataset (Note: We got the trading strategy by optimizing 2010-2018 dataset)

- We can see that our optimal trading strategy still outperforms the market by a lot
- Note that our trading portfolio was still able to improve returns even on the 2020 pandemic recession

```
Rp.test = R_test %*% W.tang

cum_ret_snp = apply(Rm_test, MARGIN=2, FUN=comb.ret)
cumulative_opt_returns = apply(Rp.test, MARGIN=2, FUN=comb.ret)
colnames(cum_ret_snp) = 'cumulative_SnP_Returns'
colnames(cumulative_opt_returns) = 'cumulative_Opt_Returns'

data.frame(cumulative_opt_returns) %>%
  mutate(Date = as.Date(index(Rm_test)), Cumulative_SnP_Returns = cum_ret_snp) %>%
  pivot_longer(c(1,3), names_to='Strategy', values_to='Cumulative_Returns') %>%
  ggplot(aes(x=Date, y=Cumulative_Returns, color=Strategy)) +
  ylab('Cumulative Returns') +
  geom_line() +
  scale_color_discrete(name = 'Investment Plan', labels=c('Cumulative Optimal Returns', 'Cumulative S&P
  ggtitle('Cumulative Returns (Out of Sample): \n Investing in Optimal Portfolio vs. S&P 500') +
  theme(plot.title = element_text(hjust = 0.5))
```



A 100 dollar investment on Feb 2018 would be worth 202.8459 dollars by June 2020 following the trading strategy. Note however, this does not take into account the transaction cost of buying and short selling 30 stocks.

```
100 * prod(1 + Rp.test)
```

```
## [1] 202.8459
```

5) Conclusion

By applying Markowitz model / Modern Portfolio Theory to the 30 randomly chosen stocks, we were able to create a strategy that has a higher return than the S&P 500. By choosing the portfolio weights that minimize the variance for a given mean and choosing the tangency portfolio that maximizes the Sharpe Ratio, we were able to get an optimal trading strategy for the 30 randomly chosen stocks.

In this paper, we were able to show that the trading strategy using Modern Portfolio Theory has a positive Alpha which indicates positive outperformance to the market, the Returns of the Optimal Portfolio has a Higher Estimated Mean than the S&P 500 Returns via Monte Carlo Simulation and it has a lower Value at Risk (VaR) and Conditional Value at Risk (CVaR).

For future studies, it would be useful to take into account transaction costs and figure out a new metric of risk that we want to minimize other than variance. Moreover, we could explore other approaches for the calculation of VaR and CVaR by Parametric modeling.

6) References

Markowitz, H. (1952), Portfolio Selection, *The Journal of Finance*, Retrieved from https://www.math.ust.hk/~maykwok/courses/ma362/07F/markowitz_JF.pdf