Portfolio Optimization in R

Suraj Udasi

December 14, 2022

Abstract

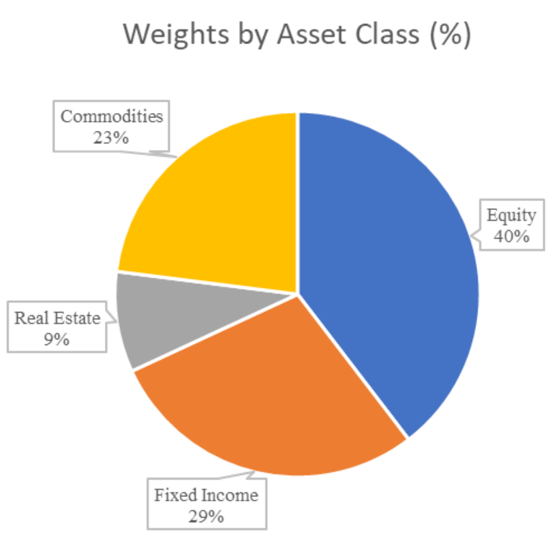
This paper attempts to provide recommendations and advice for an Ultra High Net Worth Client, based in Palo Alto, CA. The recommendation will take into consideration the risk and returns of the current portfolio and rebalance it based on various frameworks to minimize risk, maximize returns, and minimize the correlation between assets. The model will also try to predict the future returns for the stock of the rebalanced portfolio. For the scope of this assessment, S&P 500 is considered the benchmark to track the portfolio performance against. The entire analysis and the output hereof of is carried out in R and the code for the same is available as part of the appendix

Introduction

The current client portfolio is invested in various asset classes such as Equity, Fixed Income, Real Estate, and Commodities. The portfolio performance is presented below:

Initial Portfolio				Annualized Retruns (ROR)			Risk	Performance		Tracking Error
Sr. No	Tickers	Asset Class	Weights	12M Returns	18M Returns	24M Returns	12M Sigma	Sharpe Ratio	Treynor Ratio	
1	IXN	Equity	17.50%	-23.90%	-8.64%	-0.70%	0.29	-0.07	-0.02	0.07
2	QQQ	Equity	22.10%	-27.09%	-12.36%	-3.61%	0.27	-0.09	-0.02	0.08
3	IEF	Fixed Income	28.50%	-12.84%	-8.70%	-8.20%	0.10	-0.12	-0.05	0.18
4	VNQ	Real Estate	8.90%	-22.33%	-6.85%	4.48%	0.23	-0.08	-0.02	0.10
5	GLD	Commodities	23.00%	-1.43%	1.16%	-2.80%	0.13	-0.01	-0.01	0.23
			Portfolio Performance	**-16.01%**	**-6.59%**	**-2.97%**	**0.16**	**-0.09**	**-0.02**	**0.10**
			Benchmark - SP500	*-15.66%*	*-4.36%*	*3.45%*	*0.22*	*-0.06*		

The current portfolio good mix of asset classes and is invested 39.6% in Equity, 28.5% in Fixed Income, 8.9% in Real Estate, and 23.0% in Commodities



The benchmark S&P500 index is a good indicator of overall stock market performance over the years. Based on returns, the current portfolio performed worse than the benchmark in the last 12M, 18M, and 24M time periods. The current portfolio reported -2.97% returns for the 24M period as compared to the 3.45% returns of the S&P 500

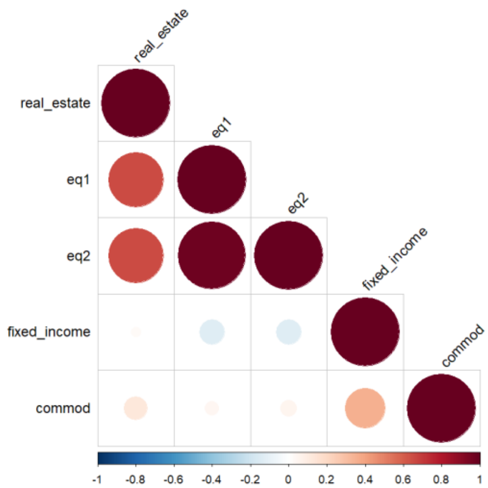
The initial risk assessment shows that equity assets (IXN and QQQ) have the highest risk of 0.28 on average, followed by real estate (VNQ) with a sigma of 0.23. Fixed Income (IEF) and commodities (GLD) have a relatively low sigma of 0.10 and 0.13, respectively as compared to the others. The overall portfolio sigma is 0.16, which is lower than the benchmark sigma of 0.22, i.e. the overall portfolio is less risky than the benchmark.

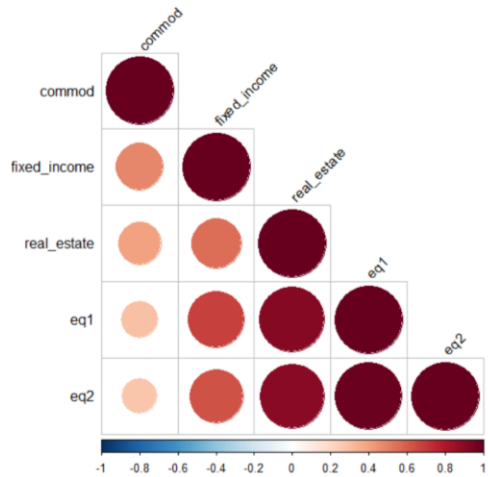
The **Sharpe Ratio** compares the returns of the assets with their risk. Currently, the 12M Sharpe ratios for all assets in the portfolio are negative as the assets are not offering positive excess returns relative to the risk. The portfolio Sharpe Ratio of -0.09 is worse than the S&P 500 Sharpe Ratio of -0.06 despite a lower sigma of 0.16 (vs sigma of 0.22 of S&P 500). This confirms that the portfolio performed worse than the market.

The **Treynor Ratio** is another metric used in this assessment to determine the risk-adjusted rate of return (AROR). The Treynor ratio helps compare the asset's excess return generated for each unit of risk as compared to the Sharpe Ratio which helps understand the asset's return compared to its risk. The portfolio has a Treynor Ratio of -0.02 vs a Sharpe Ratio of -0.09. This means that for every additional unit of risk the portfolio will generate -0.02 units of return.

Fixed Income and Commodities assets have a higher **tracking error** as compared to the other assets in the portfolio. This is mainly due to the other assets being of a similar class as the S&P 500 index set as the benchmark to calculate the tracking error. IEF & GLD also have a similarly low sigma and negative returns as compared to the other three assets in the portfolio. This shows that they have different characteristics and behave differently when compared to the equity class of assets.

To optimize the portfolio for better performance and diversification, asset correlations of the current portfolio need to be analyzed. Below are the two-correlation matrices for the current portfolio returns:



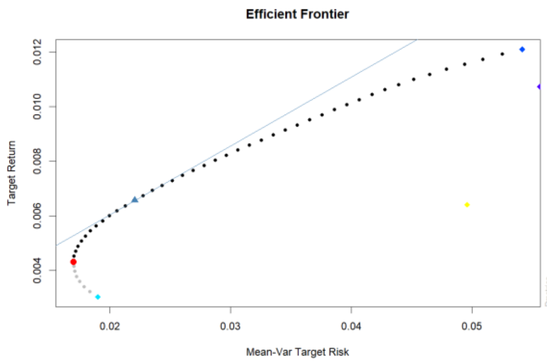


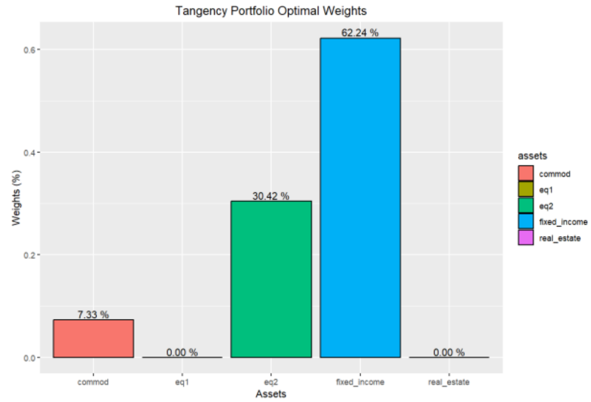
All time Correlation Matrix **Last 12M Correlation Matrix**

The matrix on the left shows that the two equity assets are highly correlated, i.e. if one asset performs well the other will also perform well. In the case of the commodities asset class (GLD), the asset returns have a very low correlation when compared to the equity assets and real estate assets, i.e. if equity and real estate assets move up, the commodities asset will move in the same direction but at a much lesser rate and vice versa in the opposite case. Whereas the Fixed income asset (IEF) is negatively correlated with the equity assets, i.e. if the returns of equity assets are down the fixed asset will move up.

The matrix on the left shows the correlations of the assets over the last 12M. They seem to have changed dramatically. All assets in the portfolio seem to have a positive correlation. Fixed income asset which was negatively correlated as seen in the matrix on the left is now showing a strong positive correlation to equity assets and the correlation between commodities asset and equity assets are stronger.

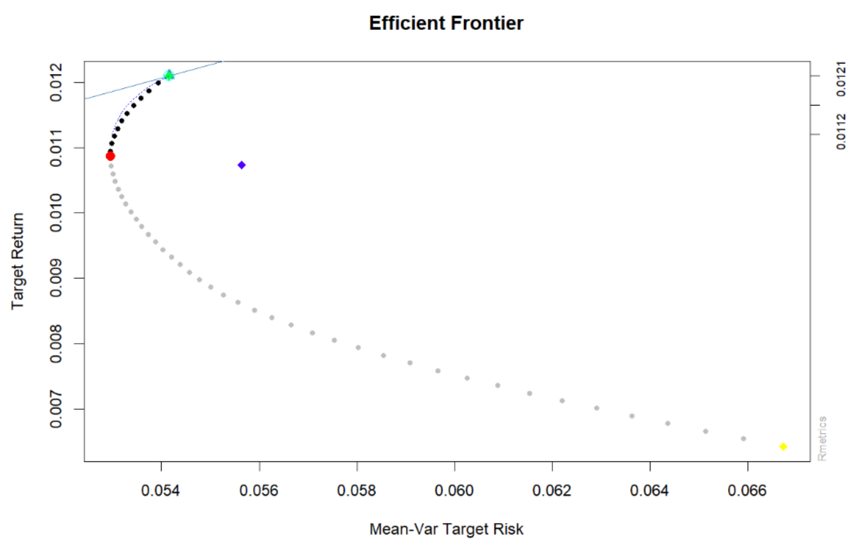
Building an efficient frontier for the initial portfolio:





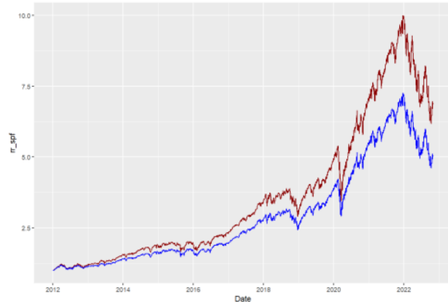
The efficient frontier shows the sets of optimal portfolios that offer the highest expected return for a target level of risk. The optimal portfolio (or tangency portfolio) weights are shown to the right of the efficient frontier. This portfolio comprises highly diversified assets offering, hence weights for eq1 (IXN) and real estate (VNQ) are 0%; in line with the correlation discussion above.

Plotting the Efficient Frontier for 3 assets with similar Sharpe Ratios (IXN, QQQ and VNQ), the light blue dotted line along the efficient frontier is the share ratio of the portfolio, almost the same as the curve of the efficient frontier. As compared to the overall portfolio frontier, the frontier for assets with a higher Sharpe Ratio shows that the return is higher for incremental units of risk. The curve as well is much rounded than flat (in case of overall portfolio frontier)



Fama French 3 Factor model

IXN



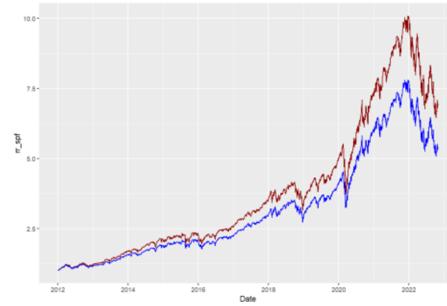

```
Call:
lm(formula = rr.v ~ subset_y$Mkt.RF + subset_y$SMB + subset_y$HML)

Residuals:
      Min        1Q    Median        3Q       Max
-0.117435 -0.005776  0.000655  0.006576  0.089482

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      0.0007932  0.0002523   3.144 0.001683 **
subset_y$Mkt.RF -0.0020829  0.0002331  -8.935  < 2e-16 ***
subset_y$SMB     0.0015816  0.0004316   3.665 0.000252 ***
subset_y$HML    -0.0009592  0.0003087  -3.107 0.001908 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01316 on 2722 degrees of freedom
  (28 observations deleted due to missingness)
Multiple R-squared:  0.03291,   Adjusted R-squared:  0.03185
F-statistic: 30.88 on 3 and 2722 DF,  p-value: < 2.2e-16
```

QQQ



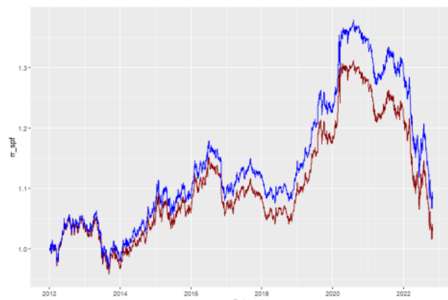

```
Call:
lm(formula = rr.v ~ subset_y$Mkt.RF + subset_y$SMB + subset_y$HML)

Residuals:
      Min        1Q    Median        3Q       Max
-0.100617 -0.005006  0.000709  0.006400  0.068388

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      0.0007893  0.0002440   3.235  0.00123 **
subset_y$Mkt.RF -0.0016915  0.0002255  -7.502 8.43e-14 ***
subset_y$SMB     0.0013405  0.0004174   3.211  0.00134 **
subset_y$HML    -0.0008732  0.0002986  -2.925  0.00348 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01272 on 2722 degrees of freedom
  (28 observations deleted due to missingness)
Multiple R-squared:  0.0242,    Adjusted R-squared:  0.02312
F-statistic:  22.5 on 3 and 2722 DF,  p-value: 2.184e-14
```

IEF




```
Call:
lm(formula = rr.v ~ subset_y$Mkt.RF + subset_y$SMB + subset_y$HML)

Residuals:
      Min        1Q    Median        3Q       Max
-0.100617 -0.005006  0.000709  0.006400  0.068388

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      0.0007893  0.0002440   3.235  0.00123 **
subset_y$Mkt.RF -0.0016915  0.0002255  -7.502 8.43e-14 ***
subset_y$SMB     0.0013405  0.0004174   3.211  0.00134 **
subset_y$HML    -0.0008732  0.0002986  -2.925  0.00348 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01272 on 2722 degrees of freedom
  (28 observations deleted due to missingness)
Multiple R-squared:  0.0242,    Adjusted R-squared:  0.02312
F-statistic:  22.5 on 3 and 2722 DF,  p-value: 2.184e-14
```

VNQ

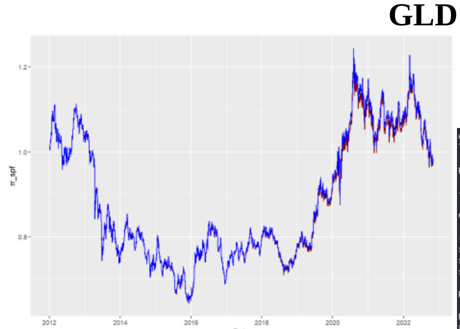

```
Call:
lm(formula = rr.v ~ subset_y$Mkt.RF + subset_y$SMB + subset_y$HML)

Residuals:
      Min        1Q    Median        3Q       Max
-0.163432 -0.005451  0.000649  0.006183  0.082731

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      0.0004172  0.0002335   1.787   0.0741 .
subset_y$Mkt.RF -0.0011078  0.0002158  -5.134 3.04e-07 ***
subset_y$SMB     0.0010085  0.0003995   2.524   0.0116 *
subset_y$HML    -0.0005316  0.0002858  -1.860   0.0630 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01218 on 2722 degrees of freedom
  (28 observations deleted due to missingness)
Multiple R-squared:  0.01164,   Adjusted R-squared:  0.01055
F-statistic: 10.68 on 3 and 2722 DF,  p-value: 5.594e-07
```

GLD




```
Call:
lm(formula = rr.v ~ subset_y$Mkt.RF + subset_y$SMB + subset_y$HML)

Residuals:
      Min        1Q    Median        3Q       Max
-0.087943 -0.004914  0.000243  0.004896  0.049058

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.700e-05  1.817e-04   0.204    0.839
subset_y$Mkt.RF 4.288e-05  1.679e-04   0.255    0.798
subset_y$SMB   -1.013e-05  3.109e-04  -0.033    0.974
subset_y$HML   -2.419e-04  2.223e-04  -1.088    0.277

Residual standard error: 0.009476 on 2722 degrees of freedom
  (30 observations deleted due to missingness)
Multiple R-squared:  0.0004645, Adjusted R-squared:  -0.0006371
F-statistic: 0.4217 on 3 and 2722 DF,  p-value: 0.7375
```

The Fama French 3 factor model aims to describe stock returns through three factors: market risk, the outperformance of small-cap companies relative to large-cap companies, and the outperformance of high book-to-market value companies versus low book-to-market value companies. It appears that most stocks in the portfolio are trending like small-cap companies that see higher returns than large-cap companies (beta > 0 and most significant p-value < 0.5), except for GLD. This trend could be due to the market realigning to the after-effects of the Covid-19 pandemic. The blue lines on the chart represent the trend for stock returns and the red lines represent the standard error.

Optimizing the portfolio:

The analysis above sets a starting point for optimizing the current portfolio for the client. The aim is to maximize returns and reduce the risk of the overall portfolio. This can be achieved by diversifying the asset mix and rebalancing the weights of the portfolio. The optimal portfolio on the efficient frontier suggests that we should sell IXN & VNQ and allot maximum weights to IEF (fixed income). But instead, we will make the following changes to the current portfolio:

a. **SELL the NASDAQ 100 (QQQ) and BUY Vanguard Dividend Appreciation ETF (VIG)**

   The initial portfolio is invested in Ishares Global Tech ETF (IXN) which has exposure to electronics, computer software, hardware, and Information Technology companies. ~50% of QQQ holdings are in the Technology sector and replicate most of the stocks already invested in through IXN. VIG invests in stocks of companies operating across diversified sectors (eg: financial, healthcare). The fund invests in growth and value stocks of companies across diversified market capitalization and in dividend-paying stocks of companies.

b. **SELL the iShares 7-10 Year Treasury Bond ETF (IEF) and BUY iShares Core U.S. Aggregate Bond ETF (AGG)**

   AGG offers broad exposure to U.S. investment-grade bonds as compared to IEF which provides exposure to intermediate-term U.S. Treasury bonds and specific segments of the U.S. Treasury market. It is also important to retain fixed income as a way to diversify the portfolio as it is are comparatively less risky asset class than equity.

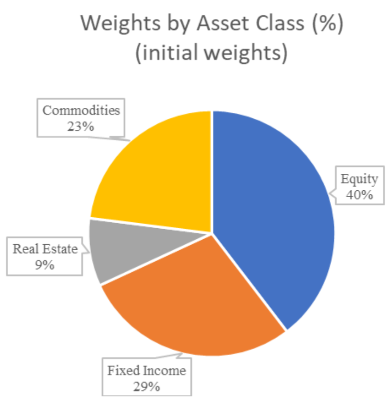
c. SELL Vanguard Real Estate ETF (VNQ) and BUY ProShares UltraShort Real Estate ETF (SRS)

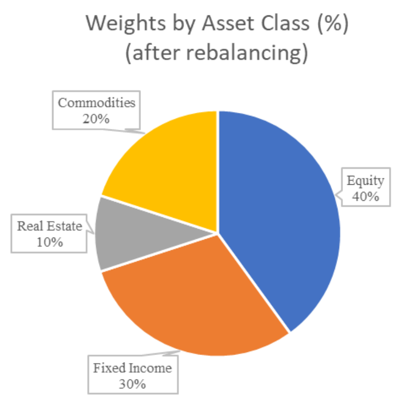
VNQ is another ETF in the current portfolio that is in stocks of companies operating across real estate sectors whereas SRS invests through derivatives in stocks of companies operating across mortgage real estate investment trusts (REITs, equity real estate investment trusts, financials, diversified financials, real estate sectors. The fund employs a short strategy and uses derivatives such as swaps to create its portfolio. SRS is a type of alternative investment to further diversify the portfolio.

d. SELL SPDR Gold Shares (GLD) and BUY Invesco DB Optimum Yield Diversified Commodity Strategy No K-1 ETF (PDBC)

PDBC primarily invests in energy, precious metals, industrial metals, and agriculture commodities and is more diversified than GLD which invests in only Gold.

e. Changed the amounts invested in asset classes





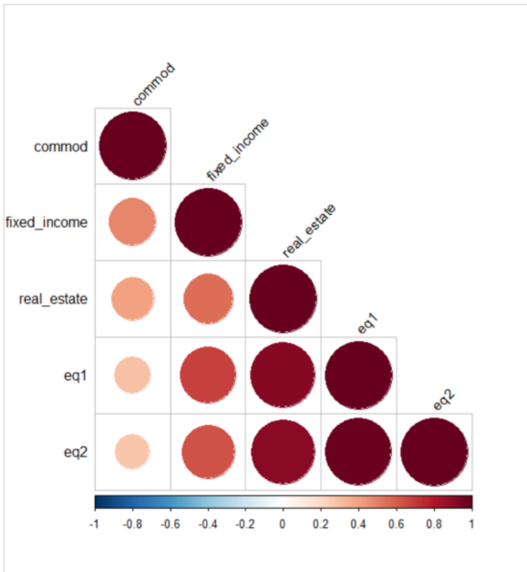
Summary of the portfolio performance after incorporating these changes:

Rebalanced Portfolio				Annualized Retruns (ROR)			Risk	Performance		Tracking Error
Sr. No	Tickers	Asset Class	Weights	12M Returns	18M Returns	24M Returns	12M Sigma	Sharpe Ratio	Treynor Ratio	
1	IXN	Equity	18.00%	-23.90%	-8.64%	-0.70%	0.29	-0.07	-0.02	0.07
2	VIG	Equity	22.00%	-6.71%	2.95%	7.45%	0.20	-0.03	-0.01	0.05
3	AGG	Fixed Income	30.00%	-10.79%	-7.40%	-6.39%	0.09	-0.11	-0.05	0.18
4	SRS	Alternative - Real Estate	10.00%	40.03%	0.40%	-18.20%	0.49	0.07	-0.02	0.69
5	PDBC	Commodities	20.00%	18.07%	17.33%	29.35%	0.20	0.07	0.05	0.26
Rebalanced Portfolio Performance				**-0.76%**	**2.09%**	**4.56%**	**0.09**	**-0.02**	**-0.01**	**0.15**
Initial Portfolio Performance				**-16.01%**	**-6.59%**	**-2.97%**	**0.16**	**-0.09**	**-0.02**	**0.10**
Benchmark - SP500				*-15.66%*	*-4.36%*	*3.45%*	*0.22*	*-0.06*		

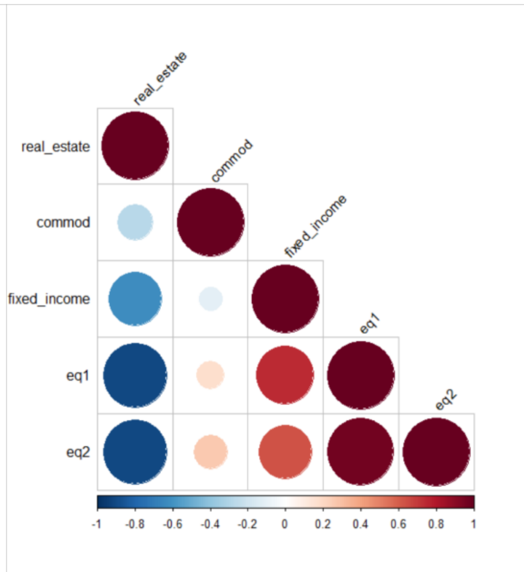
The performance of the rebalanced portfolio performance is significantly better than the initial portfolio in terms of the returns (*12M returns -0.76% vs -16.01%; 24M returns +4.56% vs -2.97%*) as well as the sigma which drastically reduced from 0.16 to 0.09.

The improvement in returns is mainly due to the high returns from SRS and PDBC which make up only 30% of the portfolio. The returns from VIG were also better than QQQ (-6.71% vs -27.09%) considering a negligible change in weights, this helped reduce the overall negative returns of the portfolio.

The rebalanced portfolio has also performed better than the S&P 500 index across all performance measures



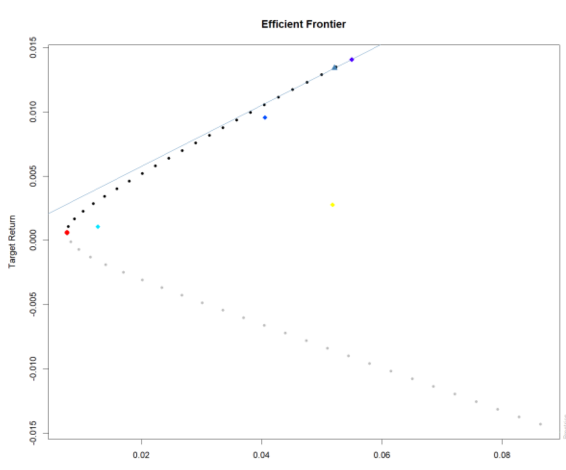
Initial Portfolio Correlation Matrix

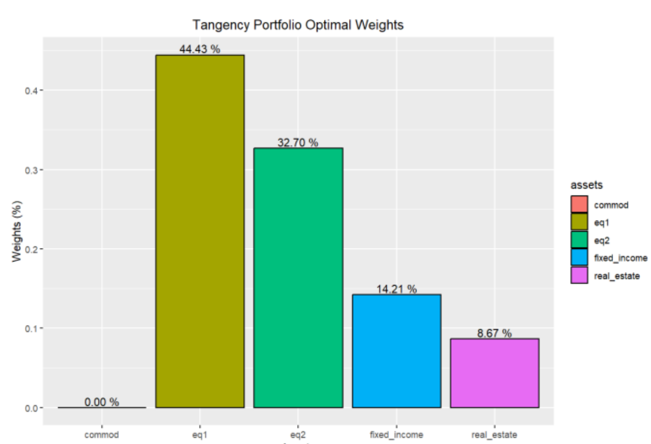


Rebalanced Portfolio Correlation Matrix

The assets in the rebalanced portfolio display higher negative correlation for the 12M as compared to the initial portfolio. This is also an indicator of higher diversification of the overall portfolio which has translated to better overall performance.

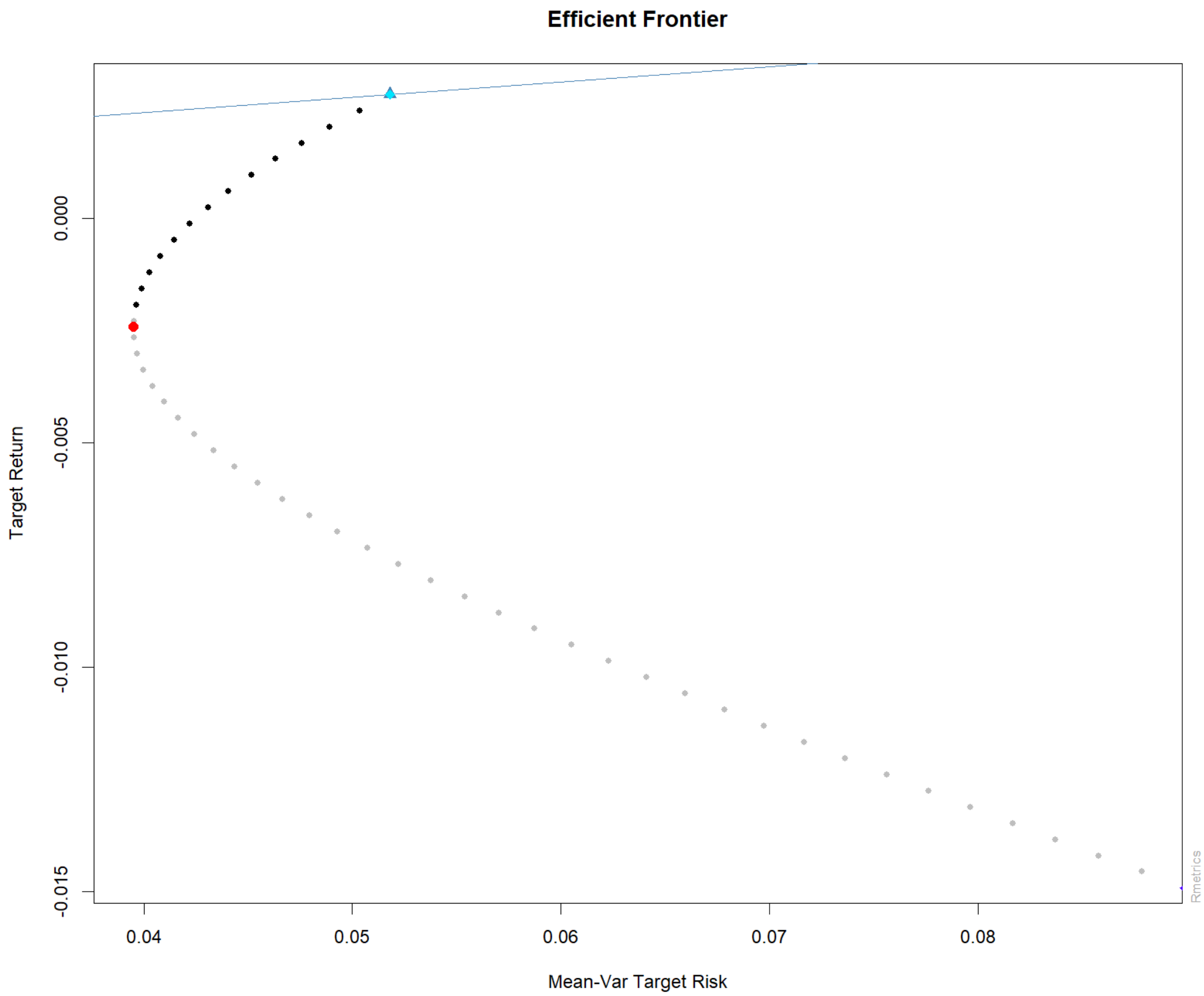
Plotting the efficient frontier of rebalanced portfolio and the tangency portfolio weights:





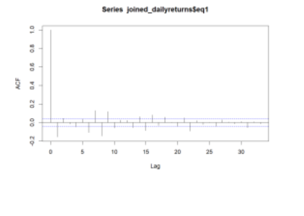
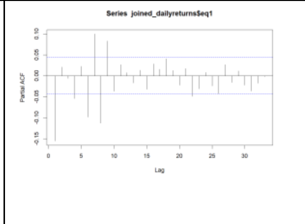
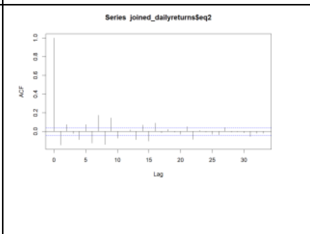
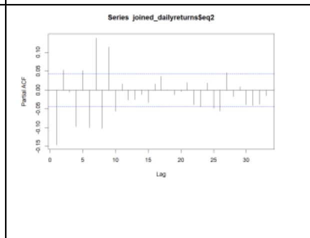
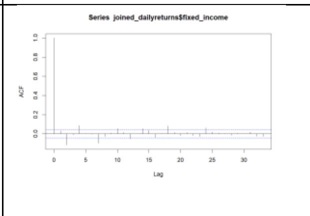
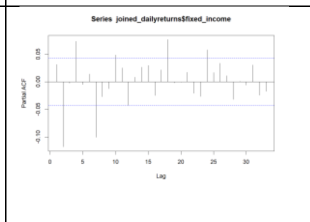
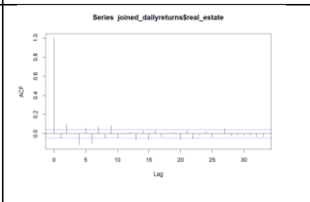
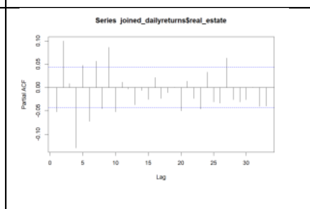
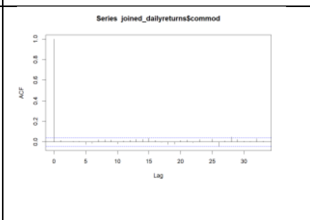
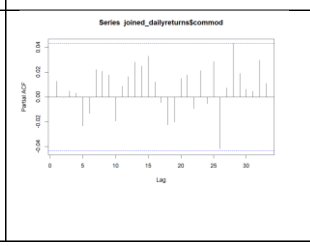
The tangency portfolio weights of the rebalanced portfolio suggest that higher returns can be achieved by completely selling the PDBC and increasing allocation in Equity assets. However, I would not recommend that to my client as it may demand a higher risk appetite (increasing equity allocation from 40% to 77%)

Efficient frontier for two assets with the highest Sharpe Ratio SRS and PDBC:



Like the initial portfolio frontier analysis, plotting the Efficient Frontier for 2 assets with similar high Sharpe Ratios of 0.07 (SRS and PDBC), shows that the return is higher for incremental units of risk and the curve is more rounded.

Applying Predictive models to see how the rebalanced portfolio will perform:

Assets	Moving Average (Trending of errors)	Lags (dependency on prior returns)	Predictions of returns for next 3 days
IXN			$pred Time Series: Start = 2040 End = 2042 Frequency = 1 [1] -0.0007231043 0.0011896456 0.0006279269 $se Time Series: Start = 2040 End = 2042 Frequency = 1 [1] 0.01467497 0.01484358 0.01485712
VIG			$pred Time Series: Start = 2040 End = 2042 Frequency = 1 [1] -0.002147078 0.003728356 -0.002911235 $se Time Series: Start = 2040 End = 2042 Frequency = 1 [1] 0.01025450 0.01029765 0.01030259
AGG			$pred Time Series: Start = 2040 End = 2042 Frequency = 1 [1] 2.493551e-04 -5.576058e-04 5.352404e-05 $se Time Series: Start = 2040 End = 2042 Frequency = 1 [1] 0.003081394 0.003083000 0.003098334
SRS			$pred Time Series: Start = 2040 End = 2042 Frequency = 1 [1] 0.0008563195 -0.0056302463 0.0014548020 $se Time Series: Start = 2040 End = 2042 Frequency = 1 [1] 0.02579322 0.02583534 0.02600978
PDBC			$pred Time Series: Start = 2040 End = 2042 Frequency = 1 [1] 0.0001377274 0.0001377274 0.0001377274 $se Time Series: Start = 2040 End = 2042 Frequency = 1 [1] 0.01177176 0.01177176 0.01177176

Appendix:

A. Code for analysis of initial portfolio:

```
#install.packages("corrplot")
library(quantmod)
library(tseries)
library(timeSeries)
library(fPortfolio)
library(caTools)
library(dplyr)
library(ggplot2)
library(PerformanceAnalytics)
library(corrplot)
source("http://www.sthda.com/upload/rquery_cormat.r")



eq_1 <- getSymbols("IXN", auto.assign = FALSE)[,6]
eq_2 <- getSymbols("QQQ", auto.assign = FALSE)[,6]
fixed_income <- getSymbols("IEF", auto.assign = FALSE)[,6]
real_estate <- getSymbols("VNQ", auto.assign = FALSE)[,6]
commod <- getSymbols("GLD", auto.assign = FALSE)[,6]

joined_dailyprices <- merge.xts(eq_1, eq_2, fixed_income, real_estate, commod)

colnames(joined_dailyprices) <- c("eq1", "eq2", "fixed_income","real_estate",
                                  "commod")

eq1_dreturns <- dailyReturn(eq_1)
eq2_dreturns <- dailyReturn(eq_2)
fixed_income_dreturns <- dailyReturn(fixed_income)
real_estate_dreturns <- dailyReturn(real_estate)
commod_dreturns <- dailyReturn(commod)

joined_dailyreturns <- merge.xts(eq1_dreturns,
                                 eq2_dreturns,
                                 fixed_income_dreturns,
                                 real_estate_dreturns,
                                 commod_dreturns)

colnames(joined_dailyreturns) <- c("eq1", "eq2", "fixed_income","real_estate",
                                   "commod")

eq1_returns <- monthlyReturn(eq_1)
eq2_returns <- monthlyReturn(eq_2)
fixed_income_returns <- monthlyReturn(fixed_income)
real_estate_returns <- monthlyReturn(real_estate)
commod_returns <- monthlyReturn(commod)

joined_monthlyreturns <- merge.xts(eq1_returns,
                                   eq2_returns,
                                   fixed_income_returns,
                                   real_estate_returns,
```

```
                                   commod_returns)

benchmark_returns <- monthlyReturn(getSymbols("^GSPC", auto.assign = FALSE))
joined_monthlyreturns <- merge.xts(joined_monthlyreturns, benchmark_returns)

colnames(joined_monthlyreturns) <- c("eq1", "eq2", "fixed_income","real_estate",
                                     "commod","SP500")

IXN_alloc <- 0.175
QQQ_alloc <- 0.221
IEF_alloc <- 0.285
VNQ_alloc <- 0.089
GLD_alloc <- 0.230

initial_weights <- c(0.175, 0.221, 0.285, 0.089, 0.230)
initial_assets  <- c("eq1", "eq2", "fixed_income","real_estate",
                     "commod")

intialweights <- rbind(initial_assets, initial_weights)

barplot(initial_weights, main="Initial Portfolio", xlab="Assets", ylab = "Weights (%)",
col=cm.colors(ncol(intialweights)))

#creating our portfolio returns viable:
joined_monthlyreturns <- as.data.frame(joined_monthlyreturns) %>%
  mutate(portfolio = IXN_alloc*eq1 +
           QQQ_alloc*eq2 +
           IEF_alloc*fixed_income +
           VNQ_alloc*real_estate +
           GLD_alloc*commod)

time_index <- nrow(joined_monthlyreturns)

timeindex_12M <- (time_index-11) : time_index
timeindex_18M <- (time_index-17) : time_index
timeindex_24M <- (time_index-23) : time_index


Returns_12M <- Return.annualized(joined_monthlyreturns[timeindex_12M,])
Returns_18M <- Return.annualized(joined_monthlyreturns[timeindex_18M,])
Returns_24M <- Return.annualized(joined_monthlyreturns[timeindex_24M,])

#Sigma

eq1_sigma <- sd(joined_monthlyreturns$eq1[timeindex_12M]) * sqrt(12)
eq2_sigma <- sd(joined_monthlyreturns$eq2[timeindex_12M]) * sqrt(12)
fixed_income_sigma <- sd(joined_monthlyreturns$fixed_income[timeindex_12M]) * sqrt(12)
real_estate_sigma <- sd(joined_monthlyreturns$real_estate[timeindex_12M]) * sqrt(12)
commod_sigma <- sd(joined_monthlyreturns$commod[timeindex_12M]) * sqrt(12)
portfolio_sigma <- sd(joined_monthlyreturns$portfolio[timeindex_12M]) * sqrt(12)
SP500_sigma <- sd(joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)

tickers <- c("eq1", "eq2", "fixed_income", "real_estate", "commod", "portfolio",
             "SP500")
sigmas <- c(eq1_sigma, eq2_sigma, fixed_income_sigma, real_estate_sigma, commod_sigma, portfolio_sigma,
            SP500_sigma)
```

```
sigma_summary <- cbind(tickers, sigmas)


# Tracking Error
eq1_te <- sd(joined_monthlyreturns$eq1[timeindex_12M] - joined_monthlyreturns$SP500[timeindex_12M]) *
sqrt(12)
eq2_te <- sd(joined_monthlyreturns$eq2[timeindex_12M] - joined_monthlyreturns$SP500[timeindex_12M]) *
sqrt(12)
fixed_income_te <- sd(joined_monthlyreturns$fixed_income[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
real_estate_te <- sd(joined_monthlyreturns$real_estate[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
commod_te <- sd(joined_monthlyreturns$commod[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
portfolio_te <- sd(joined_monthlyreturns$portfolio[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
SP500_te <- sd(joined_monthlyreturns$SP500[timeindex_12M] - joined_monthlyreturns$SP500[timeindex_12M])
* sqrt(12)

tracking_errors <- c(eq1_te, eq2_te, fixed_income_te, real_estate_te, commod_te, portfolio_te,
                     SP500_te)

te_summary <- cbind(tickers, tracking_errors)

# Sharpe Ratio

riskfree <- 0.001

eq1_sharpe <- (mean(joined_monthlyreturns$eq1[timeindex_12M]) - riskfree ) / eq1_sigma
eq2_sharpe <- (mean(joined_monthlyreturns$eq2[timeindex_12M]) - riskfree ) / eq2_sigma
fixed_income_sharpe <- (mean(joined_monthlyreturns$fixed_income[timeindex_12M]) - riskfree ) /
fixed_income_sigma
real_estate_sharpe <- (mean(joined_monthlyreturns$real_estate[timeindex_12M]) - riskfree ) / real_estate_sigma
commod_sharpe <- (mean(joined_monthlyreturns$commod[timeindex_12M]) - riskfree ) / commod_sigma
portfolio_sharpe <- (mean(joined_monthlyreturns$portfolio[timeindex_12M]) - riskfree ) / portfolio_sigma
SP500_sharpe <- (mean(joined_monthlyreturns$SP500[timeindex_12M]) - riskfree ) / SP500_sigma

sharpe_ratios <- c(eq1_sharpe, eq2_sharpe, fixed_income_sharpe, real_estate_sharpe, commod_sharpe,
portfolio_sharpe,
                   SP500_sharpe)

SR_summary <- cbind(tickers, sharpe_ratios)

# Covariance matrix and Correlation

cov(joined_monthlyreturns, use='complete.obs')
portfolio_cor <- cor(joined_monthlyreturns, use='complete.obs')
portfolio_cor_12M <- cor(joined_monthlyreturns[timeindex_12M,], use='complete.obs')

rquery.cormat(joined_monthlyreturns[,1:5])
rquery.cormat(joined_monthlyreturns[timeindex_12M,1:5])

# CAPM for Beta
#let's assume that the risk free rate is 0 (zero)
```

```
last_12_months <- joined_monthlyreturns[(time_index-11) : time_index, ]

eq1_reg <- lm(eq1 ~ SP500 ,data=last_12_months)
summary(eq1_reg)

eq2_reg <- lm(eq2 ~ SP500 ,data=last_12_months)
summary(eq2_reg)

fixed_income_reg <- lm(fixed_income ~ SP500 ,data=last_12_months)
summary(fixed_income_reg)

real_estate_reg <- lm(real_estate ~ SP500 ,data=last_12_months)
summary(real_estate_reg)

commod_reg <- lm(commod ~ SP500 ,data=last_12_months)
summary(commod_reg)

portfolio_reg <- lm(portfolio ~ SP500 ,data=last_12_months)
summary(portfolio_reg)

#How do our residuals look like in this model? are these models good?
#we want to see residuals(standardized) that are linear
plot(eq1_reg, which=2, col=c("red"))
plot(eq2_reg, which=2, col=c("blue"))
plot(fixed_income_reg, which=2, col=c("green4"))
plot(real_estate_reg, which=2, col=c("black"))
plot(commod_reg, which=2, col=c("orange"))
plot(portfolio_reg, which=2, col=c("pink"))

#we can first create a random sample out of the entire data:
testing_sample_indx <- sample(1:nrow(joined_monthlyreturns), size=5) #I'll be taking 5 dates
#we will now subset the data from just those observations from the random sample
testing_sample_data <- joined_monthlyreturns[testing_sample_indx,]
#now we'll use the predict() function to estimate the mu using the CAPM model
predict(eq1_reg, testing_sample_data)

# Treynor's ratio: this will give us the beta
eq1_treynor <- (mean(joined_monthlyreturns$eq1[timeindex_12M])-riskfree ) / eq1_reg$coefficients[2]
eq2_treynor <- (mean(joined_monthlyreturns$eq2[timeindex_12M])-riskfree ) / eq2_reg$coefficients[2]
fixed_income_treynor <- (mean(joined_monthlyreturns$fixed_income[timeindex_12M])-riskfree ) /
fixed_income_reg$coefficients[2]
real_estate_treynor <- (mean(joined_monthlyreturns$real_estate[timeindex_12M])-riskfree ) /
real_estate_reg$coefficients[2]
commod_treynor <- (mean(joined_monthlyreturns$commod[timeindex_12M])-riskfree ) /
commod_reg$coefficients[2]
portfolio_treynor <- (mean(joined_monthlyreturns$portfolio[timeindex_12M])-riskfree ) /
portfolio_reg$coefficients[2]

TRs <- c("eq1", "eq2", "fixed_income", "real_estate", "commod", "portfolio")
T_ratios <- c(eq1_treynor, eq2_treynor, fixed_income_treynor, real_estate_treynor, commod_treynor,
portfolio_treynor)

TR_summary <- cbind(TRs, T_ratios)


## Efficient Frontier using fportfolio
```

```
eq1_returns <- monthlyReturn(eq_1)
eq2_returns <- monthlyReturn(eq_2)
fixed_income_returns <- monthlyReturn(fixed_income)
real_estate_returns <- monthlyReturn(real_estate)
commod_returns <- monthlyReturn(commod)

monthlyret <- merge.xts(eq1_returns,
                        eq2_returns,
                        fixed_income_returns,
                        real_estate_returns,
                        commod_returns)

asset_tickers <- c("eq1", "eq2", "fixed income", "real estate", "commod")

colnames(monthlyret) <- asset_tickers

monthlyret_ts <- as.timeSeries(monthlyret)


effFrontier <- portfolioFrontier(
  monthlyret_ts, `setRiskFreeRate<-`(portfolioSpec(), 0.001),
  constraints = "LongOnly"
)

effFrontier_sameSR <- portfolioFrontier(
  monthlyret_ts[,c(1,2,4)], `setRiskFreeRate<-`(portfolioSpec(), 0.001),
  constraints = "LongOnly"
)

#1: Efficient Frontier
#2: Global Minimum Vairnace Portfolio
#3: Tangent (Optimal) Porfolio
#4: Risk/Return for each Asset
#5: Equal Weights Portfolio
#6: Two Assets Frontier
#7: Monte Carlo Portfolio
#8: Sharpe Ratio

plot(effFrontier, c(1,2,3,4))
plot(effFrontier_sameSR, c(1,3,8,4))
frontierweights <- getWeights(effFrontier_sameSR)
frontierweights <- getWeights(effFrontier)
colnames(frontierweights) <- c("eq1", "eq2", "fixed_income","real_estate",
                               "commod")
risk_return <- frontierPoints(effFrontier)

annualisedpoints <- data.frame(targetRisk=risk_return[, "targetRisk"] * sqrt(12),
                               targetRetrun=risk_return[,"targetReturn"] *12)

plot(annualisedpoints)

#frotnier weights
barplot(t(frontierweights), main="Frontier Weights", col=cm.colors(ncol(frontierweights)+2),
legend=colnames(frontierweights))
```

```
#minimum variance portfolio
mvp <- minvariancePortfolio(monthlyret_ts, spec = portfolioSpec(), constraints = "LongOnly")
mvp_weights <- getWeights(mvp)


tangencyport <- tangencyPortfolio(monthlyret_ts, spec = portfolioSpec(), constraints = "LongOnly")
tangencyport_weights <- getWeights(tangencyport)

barplot(tangencyport_weights, main="Tangency Portfolio", xlab="Assets", ylab = "Weights (%)",
col=cm.colors(ncol(frontierweights)))

df <- data.frame(tangencyport_weights)
assets <- colnames(frontierweights)
ggplot(data=df, aes(x=assets, y=tangencyport_weights, fill=assets))+
  geom_bar(stat="identity", position=position_dodge(), color="black")+
  geom_text(aes(label=sprintf("%.02f %%", tangencyport_weights*100)),
            position = position_dodge(width=0.9), vjust=-0.25, check_overlap = T) +
  ggtitle("Tangency Portfolio Optimal Weights") + theme(plot.title = element_text(hjust = 0.5)) +
  labs(x="Assets", y="Weights (%)")



########################################################
### Using Fama French models and interpreting results ###
########################################################


source("/Users/suraj/OneDrive/Desktop/Hult/0. Modules/4. Business Insights Through Data/Codes and Templates/8. 
Fama French multi-factor models source this file before running the next one.R")



########################################################
### Using Fama French 3 Factor model  ##################
########################################################

#calling the Fama French 3F model UDF for IXN
eq1_FF3F <- fama_french_3F(ticker="IXN", from_date='2012-01-01', to_date='2022-12-13')
summary(eq1_FF3F[[2]])#looking at factor loading - are any statisitcally significant
#now let's visualize the model error and the cumulative stock returns
ggplot(data=eq1_FF3F[[1]])+
  geom_line(aes(x=Date, y=rr_spf), color="red4")+
  geom_line(aes(x=Date, y=tr_cum), color="blue") #red is the error and blue is the stock return

#calling the Fama French 3F model UDF for QQQ
eq2_FF3F <- fama_french_3F(ticker="QQQ", from_date='2012-01-01', to_date='2022-12-13')
summary(eq2_FF3F[[2]])

ggplot(data=eq2_FF3F[[1]])+
  geom_line(aes(x=Date, y=rr_spf), color="red4")+
  geom_line(aes(x=Date, y=tr_cum), color="blue")

#calling the Fama French 3F model UDF for IEF
fixedincome_FF3F <- fama_french_3F(ticker="IEF", from_date='2012-01-01', to_date='2022-12-13')
summary(eq2_FF3F[[2]])
```

```
ggplot(data=fixedincome_FF3F[[1]])+
  geom_line(aes(x=Date, y=rr_spf), color="red4")+
  geom_line(aes(x=Date, y=tr_cum), color="blue")

#calling the Fama French 3F model UDF for VNQ
realestate_FF3F <- fama_french_3F(ticker="VNQ", from_date='2012-01-01', to_date='2022-12-13')
summary(realestate_FF3F[[2]])

ggplot(data=realestate_FF3F[[1]])+
  geom_line(aes(x=Date, y=rr_spf), color="red4")+
  geom_line(aes(x=Date, y=tr_cum), color="blue")

#calling the Fama French 3F model UDF for GLD
commod_FF3F <- fama_french_3F(ticker="GLD", from_date='2012-01-01', to_date='2022-12-31')
summary(commod_FF3F[[2]])

ggplot(data=commod_FF3F[[1]])+
  geom_line(aes(x=Date, y=rr_spf), color="red4")+
  geom_line(aes(x=Date, y=tr_cum), color="blue")



####################################################
#### ACF and pACF - daily timeseries analysis #######
####################################################
#WE WILL USE DAILY RETURNS - ROR for this because it is stationary
#ACF - will show us the moving average (MA) component - TRENDING of errors
acf(joined_dailyreturns$eq1) #eq1 - we might see 1 lags here MA(1)
acf(joined_dailyreturns$eq2) #eq2 - only 1 lags MA(1)
acf(joined_dailyreturns$fixed_income) #fixed_income - see 2 lags MA(2)
acf(joined_dailyreturns$real_estate) #real_estate - see 1 lags MA(1)
acf(joined_dailyreturns$commod) #commod - see 0 lags MA(0)

#pACF - will show us the lag - "if today depends on yesterday" - autocorrelation - AR
pacf(joined_dailyreturns$eq1) #eq1 - we might see 1 lags here AR(1)
pacf(joined_dailyreturns$eq2) #eq2 - only 1 lags AR(1)
pacf(joined_dailyreturns$fixed_income) #fixed_income - see 2 lags AR(2)
pacf(joined_dailyreturns$real_estate) #real_estate - see 1 lags AR(1)
pacf(joined_dailyreturns$commod) #commod - see 0 lags AR(0)


#######################################################################
#### Fitting ARMA on daily ROR stationary time series      ############
#######################################################################
#fitting an ARMA(1,1) on IXN:
eq1_arma <- arma(joined_dailyreturns$eq1, order=c(1,1))
summary(eq1_arma)
#however, to use the predict() function we need to use the arima function
eq1_arima <- arima(joined_dailyreturns$eq1,
                   order=c(1,0,1))
predict(eq1_arima, n.ahead =3)

#fitting an ARMA(1,1) on QQQ:
eq2_arma <- arma(joined_dailyreturns$eq2, order=c(1,1))
summary(eq2_arma)
```

```
#however, to use the predict() function we need to use the arima function
eq2_arima <- arima(joined_dailyreturns$eq2,
                   order=c(1,0,1))
predict(eq2_arima, n.ahead =3)

#fitting an ARMA(2,2) on IEF:
fixed_income_arma <- arma(joined_dailyreturns$fixed_income, order=c(2,2))
summary(fixed_income_arma)
#however, to use the predict() function we need to use the arima function
fixed_income_arima <- arima(joined_dailyreturns$fixed_income,
                            order=c(2,0,2))
predict(fixed_income_arima, n.ahead =3) #want to get forecasted values for 3 days out

#fitting an ARMA(1,1) on VNQ:
real_estate_arma <- arma(joined_dailyreturns$real_estate, order=c(1,1))
summary(real_estate_arma)

real_estate_arima <- arima(joined_dailyreturns$real_estate,
                           order=c(1,0,1))
predict(real_estate_arima, n.ahead =3) #want to get forecasted values for 3 days out

#fitting an ARMA(0,0) on GLD:
commod_arma <- arma(joined_dailyreturns$commod, order=c(0,0))
summary(commod_arma)

commod_arima <- arima(joined_dailyreturns$commod,
                      order=c(0,0,0))
predict(commod_arima, n.ahead =3)
```

B. Code for rebalancing portfolio:

```
library(quantmod)
library(tseries)
library(timeSeries)
library(fPortfolio)
library(caTools)
library(dplyr)
library(ggplot2)
library(PerformanceAnalytics)
library(corrplot)
source("http://www.sthda.com/upload/rquery_cormat.r")



eq_1 <- getSymbols("IXN", auto.assign = FALSE)[,6]
eq_2 <- getSymbols("VIG", auto.assign = FALSE)[,6]
fixed_income <- getSymbols("AGG", auto.assign = FALSE)[,6]
real_estate <- getSymbols("SRS", auto.assign = FALSE)[,6]
commod <- getSymbols("PDBC", auto.assign = FALSE)[,6]

joined_dailyprices <- merge.xts(eq_1, eq_2, fixed_income, real_estate, commod)

colnames(joined_dailyprices) <- c("eq1", "eq2", "fixed_income","real_estate",
                                  "commod")

eq1_dreturns <- dailyReturn(eq_1)
eq2_dreturns <- dailyReturn(eq_2)
fixed_income_dreturns <- dailyReturn(fixed_income)
real_estate_dreturns <- dailyReturn(real_estate)
commod_dreturns <- dailyReturn(commod)

joined_dailyreturns <- merge.xts(eq1_dreturns,
                                 eq2_dreturns,
                                 fixed_income_dreturns,
                                 real_estate_dreturns,
                                 commod_dreturns)

colnames(joined_dailyreturns) <- c("eq1", "eq2", "fixed_income","real_estate",
                                   "commod")

joined_dailyreturns <- na.omit(joined_dailyreturns, "extend")

eq1_returns <- monthlyReturn(eq_1)
eq2_returns <- monthlyReturn(eq_2)
fixed_income_returns <- monthlyReturn(fixed_income)
real_estate_returns <- monthlyReturn(real_estate)
commod_returns <- monthlyReturn(commod)

joined_monthlyreturns <- merge.xts(eq1_returns,
                                   eq2_returns,
                                   fixed_income_returns,
                                   real_estate_returns,
                                   commod_returns)

benchmark_returns <- monthlyReturn(getSymbols("^GSPC", auto.assign = FALSE))
```

```
joined_monthlyreturns <- merge.xts(joined_monthlyreturns, benchmark_returns)

colnames(joined_monthlyreturns) <- c("eq1", "eq2", "fixed_income","real_estate",
                                     "commod","SP500")

IXN_alloc <- 0.18
VIG_alloc <- 0.22
AGG_alloc <- 0.30
SRS_alloc <- 0.10
PDBC_alloc <- 0.20

#creating our portfolio returns viable:
joined_monthlyreturns <- as.data.frame(joined_monthlyreturns) %>%
  mutate(portfolio = IXN_alloc*eq1 +
           VIG_alloc*eq2 +
           AGG_alloc*fixed_income +
           SRS_alloc*real_estate +
           PDBC_alloc*commod)

time_index <- nrow(joined_monthlyreturns)

timeindex_12M <- (time_index-11) : time_index
timeindex_18M <- (time_index-17) : time_index
timeindex_24M <- (time_index-23) : time_index


Returns_12M <- Return.annualized(joined_monthlyreturns[timeindex_12M,])
Returns_18M <- Return.annualized(joined_monthlyreturns[timeindex_18M,])
Returns_24M <- Return.annualized(joined_monthlyreturns[timeindex_24M,])

#Sigma

eq1_sigma <- sd(joined_monthlyreturns$eq1[timeindex_12M]) * sqrt(12)
eq2_sigma <- sd(joined_monthlyreturns$eq2[timeindex_12M]) * sqrt(12)
fixed_income_sigma <- sd(joined_monthlyreturns$fixed_income[timeindex_12M]) * sqrt(12)
real_estate_sigma <- sd(joined_monthlyreturns$real_estate[timeindex_12M]) * sqrt(12)
commod_sigma <- sd(joined_monthlyreturns$commod[timeindex_12M]) * sqrt(12)
portfolio_sigma <- sd(joined_monthlyreturns$portfolio[timeindex_12M]) * sqrt(12)
SP500_sigma <- sd(joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)

tickers <- c("eq1", "eq2", "fixed_income", "real_estate", "commod", "portfolio",
             "SP500")
sigmas <- c(eq1_sigma, eq2_sigma, fixed_income_sigma, real_estate_sigma, commod_sigma, portfolio_sigma,
            SP500_sigma)

sigma_summary <- cbind(tickers, sigmas)


# Tracking Error
eq1_te <- sd(joined_monthlyreturns$eq1[timeindex_12M] - joined_monthlyreturns$SP500[timeindex_12M]) *
sqrt(12)
eq2_te <- sd(joined_monthlyreturns$eq2[timeindex_12M] - joined_monthlyreturns$SP500[timeindex_12M]) *
sqrt(12)
fixed_income_te <- sd(joined_monthlyreturns$fixed_income[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
```

```
real_estate_te <- sd(joined_monthlyreturns$real_estate[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
commod_te <- sd(joined_monthlyreturns$commod[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
portfolio_te <- sd(joined_monthlyreturns$portfolio[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
SP500_te <- sd(joined_monthlyreturns$SP500[timeindex_12M] - joined_monthlyreturns$SP500[timeindex_12M])
* sqrt(12)

tracking_errors <- c(eq1_te, eq2_te, fixed_income_te, real_estate_te, commod_te, portfolio_te,
                     SP500_te)

te_summary <- cbind(tickers, tracking_errors)

# Sharpe Ratio

riskfree <- 0.003

eq1_sharpe <- (mean(joined_monthlyreturns$eq1[timeindex_12M]) - riskfree ) / eq1_sigma
eq2_sharpe <- (mean(joined_monthlyreturns$eq2[timeindex_12M]) - riskfree ) / eq2_sigma
fixed_income_sharpe <- (mean(joined_monthlyreturns$fixed_income[timeindex_12M]) - riskfree ) /
fixed_income_sigma
real_estate_sharpe <- (mean(joined_monthlyreturns$real_estate[timeindex_12M]) - riskfree ) / real_estate_sigma
commod_sharpe <- (mean(joined_monthlyreturns$commod[timeindex_12M]) - riskfree ) / commod_sigma
portfolio_sharpe <- (mean(joined_monthlyreturns$portfolio[timeindex_12M]) - riskfree ) / portfolio_sigma
SP500_sharpe <- (mean(joined_monthlyreturns$SP500[timeindex_12M]) - riskfree ) / SP500_sigma

sharpe_ratios <- c(eq1_sharpe, eq2_sharpe, fixed_income_sharpe, real_estate_sharpe, commod_sharpe,
portfolio_sharpe,
                   SP500_sharpe)

SR_summary <- cbind(tickers, sharpe_ratios)

# Covariance matrix and Correlation

cov(joined_monthlyreturns, use='complete.obs')
portfolio_cor <- cor(joined_monthlyreturns, use='complete.obs')
portfolio_cor_12M <- cor(joined_monthlyreturns[timeindex_12M,], use='complete.obs')

rquery.cormat(joined_monthlyreturns[,1:5])
rquery.cormat(joined_monthlyreturns[timeindex_12M,1:5])

# CAPM for Beta
#let's assume that the risk free rate is 0 (zero)

last_12_months <- joined_monthlyreturns[(time_index-11) : time_index, ]

eq1_reg <- lm(eq1 ~ SP500 ,data=last_12_months)
summary(eq1_reg)

eq2_reg <- lm(eq2 ~ SP500 ,data=last_12_months)
summary(eq2_reg)

fixed_income_reg <- lm(fixed_income ~ SP500 ,data=last_12_months)
summary(fixed_income_reg)
```

```
real_estate_reg <- lm(real_estate ~ SP500 ,data=last_12_months)
summary(real_estate_reg)

commod_reg <- lm(commod ~ SP500 ,data=last_12_months)
summary(commod_reg)

portfolio_reg <- lm(portfolio ~ SP500 ,data=last_12_months)
summary(portfolio_reg)

#create a random sample out of the entire data:
testing_sample_indx <- sample(1:nrow(joined_monthlyreturns), size=5) #I'll be taking 5 dates
#subset the data from just those observations from the random sample
testing_sample_data <- joined_monthlyreturns[testing_sample_indx,]
#predict() function to estimate the mu using the CAPM model
predict(eq1_reg, testing_sample_data)

# Treynor's ratio: this will give us the beta
eq1_treynor <- (mean(joined_monthlyreturns$eq1[timeindex_12M])-riskfree ) / eq1_reg$coefficients[2]
eq2_treynor <- (mean(joined_monthlyreturns$eq2[timeindex_12M])-riskfree ) / eq2_reg$coefficients[2]
fixed_income_treynor <- (mean(joined_monthlyreturns$fixed_income[timeindex_12M])-riskfree ) /
fixed_income_reg$coefficients[2]
real_estate_treynor <- (mean(joined_monthlyreturns$real_estate[timeindex_12M])-riskfree ) /
real_estate_reg$coefficients[2]
commod_treynor <- (mean(joined_monthlyreturns$commod[timeindex_12M])-riskfree ) /
commod_reg$coefficients[2]
portfolio_treynor <- (mean(joined_monthlyreturns$portfolio[timeindex_12M])-riskfree ) /
portfolio_reg$coefficients[2]

TRs <- c("eq1", "eq2", "fixed_income", "real_estate", "commod", "portfolio")
T_ratios <- c(eq1_treynor, eq2_treynor, fixed_income_treynor, real_estate_treynor, commod_treynor,
portfolio_treynor)

TR_summary <- cbind(TRs, T_ratios)

####################################################
#### ACF and pACF - daily timeseries analysis #######
####################################################

acf(joined_dailyreturns$eq1) #eq1 - we might see 2 lags here MA(2)
acf(joined_dailyreturns$eq2) #eq2 - only 2 lags MA(2)
acf(joined_dailyreturns$fixed_income) #fixed_income - see 2 lags MA(2)
acf(joined_dailyreturns$real_estate) #real_estate - see 2 lags MA(2)
acf(joined_dailyreturns$commod) #commod - see 0 lags MA(0)



#pACF - will show us the lag - "if today depends on yesterday" - autocorrelation - AR
pacf(joined_dailyreturns$eq1) #eq1 - we might see 1 lags here AR(1)
pacf(joined_dailyreturns$eq2) #eq2 - only 2 lags AR(2)
pacf(joined_dailyreturns$fixed_income) #fixed_income - see 0 lags AR(0)
pacf(joined_dailyreturns$real_estate) #real_estate - see 2 lags AR(2)
pacf(joined_dailyreturns$commod) #commod - see 0 lags AR(0)


##################################################################
#### Fitting ARMA on daily ROR stationary time series      ###########
```

```
#### NOTE : if you wanted to forecast non-stationary, use ARIMA #######
#########################################################################
#fitting an ARMA(1,2) on IXN:
eq1_arma <- arma(joined_dailyreturns$eq1, order=c(1,2))
summary(eq1_arma)
#however, to use the predict() function we need to use the arima function
eq1_arima <- arima(joined_dailyreturns$eq1,
            order=c(1,0,2))
predict(eq1_arima, n.ahead =3) #want to get forecasted values for 3 days out

#fitting an ARMA(2,2) on VIG:
eq2_arma <- arma(joined_dailyreturns$eq2, order=c(2,2))
summary(eq2_arma)
#however, to use the predict() function we need to use the arima function
eq2_arima <- arima(joined_dailyreturns$eq2,
            order=c(2,0,2))
predict(eq2_arima, n.ahead =3) #want to get forecasted values for 3 days out

#fitting an ARMA(0,2) on AGG:
fixed_income_arma <- arma(joined_dailyreturns$fixed_income, order=c(0,2))
summary(fixed_income_arma)
#however, to use the predict() function we need to use the arima function
fixed_income_arima <- arima(joined_dailyreturns$fixed_income,
                   order=c(0,0,2))
predict(fixed_income_arima, n.ahead =3) #want to get forecasted values for 3 days out

#fitting an ARMA(2,2) on SRS:
real_estate_arma <- arma(joined_dailyreturns$real_estate, order=c(2,2))
summary(real_estate_arma)

real_estate_arima <- arima(joined_dailyreturns$real_estate,
                  order=c(2,0,2))
predict(real_estate_arima, n.ahead =3) #want to get forecasted values for 3 days out

#fitting an ARMA(0,0) on PDBC:
commod_arma <- arma(joined_dailyreturns$commod, order=c(0,0))
summary(commod_arma)

commod_arima <- arima(joined_dailyreturns$commod,
              order=c(0,0,0))
predict(commod_arima, n.ahead =3)
```

C. Code of Efficient Frontier of Rebalanced Portfolio:

```
library(quantmod)
library(tseries)
library(timeSeries)
library(fPortfolio)
library(caTools)
library(dplyr)
library(ggplot2)
library(PerformanceAnalytics)
library(corrplot)
source("http://www.sthda.com/upload/rquery_cormat.r")



eq_1 <- getSymbols("IXN", auto.assign = FALSE)[,6]
eq_2 <- getSymbols("VIG", auto.assign = FALSE)[,6]
fixed_income <- getSymbols("AGG", auto.assign = FALSE)[,6]
real_estate <- getSymbols("SRS", auto.assign = FALSE)[,6]
commod <- getSymbols("PDBC", auto.assign = FALSE)[,6]

joined_dailyprices <- merge.xts(eq_1, eq_2, fixed_income, real_estate, commod)

colnames(joined_dailyprices) <- c("eq1", "eq2", "fixed_income","real_estate",
                                  "commod")

eq1_dreturns <- dailyReturn(eq_1)
eq2_dreturns <- dailyReturn(eq_2)
fixed_income_dreturns <- dailyReturn(fixed_income)
real_estate_dreturns <- dailyReturn(real_estate)
commod_dreturns <- dailyReturn(commod)

joined_dailyreturns <- merge.xts(eq1_dreturns,
                                 eq2_dreturns,
                                 fixed_income_dreturns,
                                 real_estate_dreturns,
                                 commod_dreturns)

colnames(joined_dailyreturns) <- c("eq1", "eq2", "fixed_income","real_estate",
                                   "commod")

eq1_returns <- monthlyReturn(eq_1)
eq2_returns <- monthlyReturn(eq_2)
fixed_income_returns <- monthlyReturn(fixed_income)
real_estate_returns <- monthlyReturn(real_estate)
commod_returns <- monthlyReturn(commod)

joined_monthlyreturns <- merge.xts(eq1_returns,
                                   eq2_returns,
                                   fixed_income_returns,
                                   real_estate_returns,
                                   commod_returns)

benchmark_returns <- monthlyReturn(getSymbols("^GSPC", auto.assign = FALSE))
joined_monthlyreturns <- merge.xts(joined_monthlyreturns, benchmark_returns)
```

```
colnames(joined_monthlyreturns) <- c("eq1", "eq2", "fixed_income","real_estate",
                                    "commod","SP500")

IXN_alloc <- 0.18
VIG_alloc <- 0.22
AGG_alloc <- 0.30
SRS_alloc <- 0.10
PDBC_alloc <- 0.20

#creating our portfolio returns viable:
joined_monthlyreturns <- as.data.frame(joined_monthlyreturns) %>%
  mutate(portfolio = IXN_alloc*eq1 +
         VIG_alloc*eq2 +
         AGG_alloc*fixed_income +
         SRS_alloc*real_estate +
         PDBC_alloc*commod)

time_index <- nrow(joined_monthlyreturns)

timeindex_12M <- (time_index-11) : time_index
timeindex_18M <- (time_index-17) : time_index
timeindex_24M <- (time_index-23) : time_index


Returns_12M <- Return.annualized(joined_monthlyreturns[timeindex_12M,])
Returns_18M <- Return.annualized(joined_monthlyreturns[timeindex_18M,])
Returns_24M <- Return.annualized(joined_monthlyreturns[timeindex_24M,])

#Sigma

eq1_sigma <- sd(joined_monthlyreturns$eq1[timeindex_12M]) * sqrt(12)
eq2_sigma <- sd(joined_monthlyreturns$eq2[timeindex_12M]) * sqrt(12)
fixed_income_sigma <- sd(joined_monthlyreturns$fixed_income[timeindex_12M]) * sqrt(12)
real_estate_sigma <- sd(joined_monthlyreturns$real_estate[timeindex_12M]) * sqrt(12)
commod_sigma <- sd(joined_monthlyreturns$commod[timeindex_12M]) * sqrt(12)
portfolio_sigma <- sd(joined_monthlyreturns$portfolio[timeindex_12M]) * sqrt(12)
SP500_sigma <- sd(joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)

tickers <- c("eq1", "eq2", "fixed_income", "real_estate", "commod", "portfolio",
             "SP500")
sigmas <- c(eq1_sigma, eq2_sigma, fixed_income_sigma, real_estate_sigma, commod_sigma, portfolio_sigma,
            SP500_sigma)

sigma_summary <- cbind(tickers, sigmas)


# Tracking Error
eq1_te <- sd(joined_monthlyreturns$eq1[timeindex_12M] - joined_monthlyreturns$SP500[timeindex_12M]) *
sqrt(12)
eq2_te <- sd(joined_monthlyreturns$eq2[timeindex_12M] - joined_monthlyreturns$SP500[timeindex_12M]) *
sqrt(12)
fixed_income_te <- sd(joined_monthlyreturns$fixed_income[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
real_estate_te <- sd(joined_monthlyreturns$real_estate[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
```

```
commod_te <- sd(joined_monthlyreturns$commod[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
portfolio_te <- sd(joined_monthlyreturns$portfolio[timeindex_12M] -
joined_monthlyreturns$SP500[timeindex_12M]) * sqrt(12)
SP500_te <- sd(joined_monthlyreturns$SP500[timeindex_12M] - joined_monthlyreturns$SP500[timeindex_12M])
* sqrt(12)

tracking_errors <- c(eq1_te, eq2_te, fixed_income_te, real_estate_te, commod_te, portfolio_te,
                 SP500_te)

te_summary <- cbind(tickers, tracking_errors)

# Sharpe Ratio

riskfree <- 0.003

eq1_sharpe <- (mean(joined_monthlyreturns$eq1[timeindex_12M]) - riskfree ) / eq1_sigma
eq2_sharpe <- (mean(joined_monthlyreturns$eq2[timeindex_12M]) - riskfree ) / eq2_sigma
fixed_income_sharpe <- (mean(joined_monthlyreturns$fixed_income[timeindex_12M]) - riskfree ) /
fixed_income_sigma
real_estate_sharpe <- (mean(joined_monthlyreturns$real_estate[timeindex_12M]) - riskfree ) / real_estate_sigma
commod_sharpe <- (mean(joined_monthlyreturns$commod[timeindex_12M]) - riskfree ) / commod_sigma
portfolio_sharpe <- (mean(joined_monthlyreturns$portfolio[timeindex_12M]) - riskfree ) / portfolio_sigma
SP500_sharpe <- (mean(joined_monthlyreturns$SP500[timeindex_12M]) - riskfree ) / SP500_sigma

sharpe_ratios <- c(eq1_sharpe, eq2_sharpe, fixed_income_sharpe, real_estate_sharpe, commod_sharpe,
portfolio_sharpe,
                SP500_sharpe)

SR_summary <- cbind(tickers, sharpe_ratios)

# Covariance matrix and Correlation

cov(joined_monthlyreturns, use='complete.obs')
portfolio_cor <- cor(joined_monthlyreturns, use='complete.obs')
portfolio_cor_12M <- cor(joined_monthlyreturns[timeindex_12M,], use='complete.obs')

rquery.cormat(joined_monthlyreturns[,1:5])
rquery.cormat(joined_monthlyreturns[timeindex_12M,1:5])

# CAPM for Beta
#let's assume that the risk free rate is 0 (zero)

last_12_months <- joined_monthlyreturns[(time_index-11) : time_index, ]

eq1_reg <- lm(eq1 ~ SP500 ,data=last_12_months)
summary(eq1_reg)

eq2_reg <- lm(eq2 ~ SP500 ,data=last_12_months)
summary(eq2_reg)

fixed_income_reg <- lm(fixed_income ~ SP500 ,data=last_12_months)
summary(fixed_income_reg)

real_estate_reg <- lm(real_estate ~ SP500 ,data=last_12_months)
summary(real_estate_reg)
```

```
commod_reg <- lm(commod ~ SP500 ,data=last_12_months)
summary(commod_reg)

portfolio_reg <- lm(portfolio ~ SP500 ,data=last_12_months)
summary(portfolio_reg)

#How do our residuals look like in this model? are these models good?
#we want to see residuals(standardized) that are linear
plot(eq1_reg, which=2, col=c("red"))
plot(eq2_reg, which=2, col=c("blue"))
plot(fixed_income_reg, which=2, col=c("green4"))
plot(real_estate_reg, which=2, col=c("black"))
plot(commod_reg, which=2, col=c("orange"))
plot(portfolio_reg, which=2, col=c("pink"))

#we can first create a random sample out of the entire data:
testing_sample_indx <- sample(1:nrow(joined_monthlyreturns), size=5) #I'll be taking 5 dates
#we will now subset the data from just those observations from the random sample
testing_sample_data <- joined_monthlyreturns[testing_sample_indx,]
#now we'll use the predict() function to estimate the mu using the CAPM model
predict(eq1_reg, testing_sample_data)

# Treynor's ratio: this will give us the beta
eq1_treynor <- (mean(joined_monthlyreturns$eq1[timeindex_12M])-riskfree ) / eq1_reg$coefficients[2]
eq2_treynor <- (mean(joined_monthlyreturns$eq2[timeindex_12M])-riskfree ) / eq2_reg$coefficients[2]
fixed_income_treynor <- (mean(joined_monthlyreturns$fixed_income[timeindex_12M])-riskfree ) /
fixed_income_reg$coefficients[2]
real_estate_treynor <- (mean(joined_monthlyreturns$real_estate[timeindex_12M])-riskfree ) /
real_estate_reg$coefficients[2]
commod_treynor <- (mean(joined_monthlyreturns$commod[timeindex_12M])-riskfree ) /
commod_reg$coefficients[2]
portfolio_treynor <- (mean(joined_monthlyreturns$portfolio[timeindex_12M])-riskfree ) /
portfolio_reg$coefficients[2]

TRs <- c("eq1", "eq2", "fixed_income", "real_estate", "commod", "portfolio")
T_ratios <- c(eq1_treynor, eq2_treynor, fixed_income_treynor, real_estate_treynor, commod_treynor,
portfolio_treynor)

TR_summary <- cbind(TRs, T_ratios)

## Efficient Frontier using fportfolio

eq1_returns <- monthlyReturn(eq_1)
eq2_returns <- monthlyReturn(eq_2)
fixed_income_returns <- monthlyReturn(fixed_income)
real_estate_returns <- monthlyReturn(real_estate)
commod_returns <- monthlyReturn(commod)

monthlyret <- merge.xts(eq1_returns,
                        eq2_returns,
                        fixed_income_returns,
                        real_estate_returns,
                        commod_returns)

monthlyret <- na.omit(monthlyret, "extend")
```

```
asset_tickers <- c("eq1", "eq2", "fixed income", "real estate", "commod")

colnames(monthlyret) <- asset_tickers

monthlyret_ts <- as.timeSeries(monthlyret)


effFrontier <- portfolioFrontier(
  monthlyret_ts, `setRiskFreeRate<-`(portfolioSpec(), 0.001),
  constraints = "LongOnly"
)



#1: Efficient Frontier
#2: Global Minimum Vairnace Portfolio
#3: Tangent (Optimal) Porfolio
#4: Risk/Return for each Asset
#5: Equal Weights Portfolio
#6: Two Assets Frontier
#7: Monte Carlo Portfolio
#8: Sharpe Ratio

plot(effFrontier, c(1,2,3,4))

frontierweights <- getWeights(effFrontier)
colnames(frontierweights) <- c("eq1", "eq2", "fixed_income","real_estate",
                              "commod")
risk_return <- frontierPoints(effFrontier)

annualisedpoints <- data.frame(targetRisk=risk_return[, "targetRisk"] * sqrt(12),
                              targetRetrun=risk_return[,"targetReturn"] *12)

plot(annualisedpoints)

#frotnier weights
barplot(t(frontierweights), main="Frontier Weights", col=cm.colors(ncol(frontierweights)+2),
legend=colnames(frontierweights))


#minimum variance portfolio
mvp <- minvariancePortfolio(monthlyret_ts, spec = portfolioSpec(), constraints = "LongOnly")
mvp_weights <- getWeights(mvp)


tangencyport <- tangencyPortfolio(monthlyret_ts, spec = portfolioSpec(), constraints = "LongOnly")
tangencyport_weights <- getWeights(tangencyport)

barplot(tangencyport_weights, main="Tangency Portfolio", xlab="Assets", ylab = "Weights (%)",
col=cm.colors(ncol(frontierweights)))

df <- data.frame(tangencyport_weights)
assets <- colnames(frontierweights)
ggplot(data=df, aes(x=assets, y=tangencyport_weights, fill=assets))+
  geom_bar(stat="identity", position=position_dodge(), color="black")+
```

```
  geom_text(aes(label=sprintf("%.02f %%", tangencyport_weights*100)),
            position = position_dodge(width=0.9), vjust=-0.25, check_overlap = T) +
  ggtitle("Tangency Portfolio Optimal Weights") + theme(plot.title = element_text(hjust = 0.5)) +
  labs(x="Assets", y="Weights (%)")

# Highest Sharpe portfolio
effFrontier_sameSR <- portfolioFrontier(
  monthlyret_ts[,c(4,5)], `setRiskFreeRate<-`(portfolioSpec(), 0.001),
  constraints = "LongOnly"
)

plot(effFrontier_sameSR, c(1,2,3,4,8))
```