Mälardalens universitet

Seminar Report

Portfolios & Optimization

Martin Westberg
Nicolas Kuiper
Group G

Analytical Finance with MATLAB, HT 2022
Course code: MAA321
Instructor: Jean-Paul Murara

Abstract

The aim of this report is to build portfolios consisting of a selection of twelve stocks retrieved from Yahoo! Finance. Three different types of portfolios pertaining to different trading strategies are built. Their returns are then analysed over a defined period of time using MATLAB to compare the efficiency of each portfolio and the corresponding trading strategy.

Table of Contents

1. Introduction

There are an enormous number of approaches to investing in financial assets, and it is quite difficult and nearly impossible to always be successful in all of our trades. A popular way to manage risk is called diversifying, which means to hold a variety of different assets in hopes of reducing the risk of losing too much money when one asset loses too much value.

When looking deeper into the diversification of our portfolio, there also exists countless financial instruments that could help us achieve our goals, but from the start, it is important to choose our assets in a way that optimises our returns by not exposing us to too much risk. This paper focuses on figuring out the best possible way to approach an investment portfolio by using different tools for portfolio optimization.

2. Theory

2.1. Analysis of Historical Data

Data collected about past events and circumstances pertaining to a particular topic is known as historical data. In finance, it is used to assess past events and project future returns of assets, as well as to determine which variables might influence future assets' performance and returns and to what extent these variables might affect those returns. Using historical data, one can calculate historical volatility and predict future data points based on standard deviations. A financial security's history and performance can provide insight into how it has responded to external or internal factors that may have caused returns to rise or fall. Often, changes in the economic cycle or global events may be responsible for such events. The data used in this report belongs to American stocks retrieved from Yahoo! Finance and is mostly concerned with the date and closing price for each asset.

2.2. Time Value of Money

Money's time value means that a sum of money now is worth more than it will be in the future. The idea behind the time value of money is that, because money can grow only through investing, delaying an investment is a lost opportunity.

If we decided to invest a certain amount of money in a savings account, we could calculate the future value of money by considering the amount of money invested, the return it can earn, and the time frame. The formula for the future value of our investment is then defined as follows:

$$FV = PV\left(1 + \frac{r}{n}\right)^{n\times t}$$

Where FV is the future value, PV is the present value, r is the interest rate, t is the number of years and m is the number of compounding periods per year.

In MATLAB, the function '*fvfix*' (*future value with fixed periodic payments*) corresponds to the formula above and takes the *interest rate*, *number of periods*, *periodic payments*, *present investment* and *payments due* as arguments in the following way:

FutureVal = fvfix(Rate,NumPeriods,Payment,PresentVal,Due)

2.3. Portfolio

A portfolio is a collection of various assets such as stocks, bonds, cryptocurrency, real estate, and so on. These are referred to as "asset classes" in an investing portfolio. In an ideal world, an investor's portfolio would include a diverse range of asset classes to optimise potential growth and risk. This is a strategy called diversification, which involves owning many types of assets that will perform differently than one another based on market conditions. [1]

Two important characteristics of a portfolio are the weights of the assets and the expected return for the portfolio. The portfolio weights are, simply put, a fraction of the total portfolio represented by an asset, such is so the total sum of the weights always add up to 1. In mathematical notation:

$$\sum_{i=1}^{n} \omega_i = 1 \,, \ \omega_i = \frac{I_i}{V_p} \,, \ V_p = \sum_{i=1}^{n} I_i$$

With every ω_i equal to investment in asset i (I_i) divided by total value of the portfolio (V_p), where the value of the portfolio equals the sum of the investments. On the other hand, the expected return of the is calculated by multiplying the weight of each asset by its expected return. We then add the values for each investment to get the total expected return for your portfolio. In mathematical notation:

$$E[r_p] \ = \ \sum_{i=1}^{n} \omega_i \times E[r_i]$$

Where r_p is the return of the whole portfolio and r_i is the return of asset i

2.4. Efficient Frontier

The efficient frontier is a combination of investment portfolios that generates an "efficient" region within the risk-return variables. It is formally, the set of portfolios that meet the condition that no other portfolio exists with a higher return and the same risk. Any portfolio lying below the efficient frontier is considered not optimal because they provide less return for the same level of risk. Additionally, any portfolio that gathers together to the right side of the frontier line are also considered suboptimal because they have too much risk for the determined return.

Many assumptions in the efficient frontier and current portfolio theory may not accurately represent reality. One of the assumptions, for example, is that asset returns follow a normal distribution; while in actuality, securities often experience returns that are more than three standard deviations away from the mean. As a result, asset returns are considered to follow a heavy-tailed distribution. Another assumption is that investors are rational and take rational; but it is well known that this is not at all true, and there exists investors who are risk takers even though the returns may not be all that worthy. So, while these types of assumptions are quite not realistic, it is still useful to use efficient frontiers to manage a portfolio in an optimal way under "perfect" conditions.

2.5. Trading Strategies

A methodical process to buy and sell assets on the market is known as a trading strategy, and they are commonly based on certain criteria used to make rational trading decisions. Trading strategies can be either basic, like buying and holding an asset, or quite complicated such as buying, for example, a stock and its derivatives to manage any risk pertaining to that asset.

Most strategies are focused on either technical and fundamental analysis, by employing measurable data that can then be evaluated in the future to assess how accurate the strategy was. The purpose here is to create a trading strategy based on factual and technical information and to adhere to it strictly and at the same time, as market conditions or personal objectives change, a trading strategy should be regularly re-evaluated and modified in order to adapt to changes and therefore maximise our returns.

2.6. Portfolio Optimization and Sharpe Ratio

Portfolio optimization is the process of choosing, based on some objective (such as maximising returns while minimising risks), the best portfolio between a set of portfolios being considered. In our case, the portfolios under consideration are those that fall on the efficient frontier, and since all the assets discussed in this report are of the same asset class (i.e. stocks), our portfolio optimization problem focuses on optimising the weights of the portfolio assets as to maximise returns with a minimum risk.

An important tool to achieve this optimization problem is the *Sharpe ratio*. This is a mathematical formula that compares the return of an investment with the risk associated with that investment [2]. In its simplest form, the Sharpe ratio is defined as:

$$Sharpe\ ratio\ =\ \frac{r_p - r_f}{\sigma_p}$$

Where r_p is the return of the portfolio, r_f is the risk-free interest rate and σ_p is the standard deviation of the portfolio. If we consider r_f to be a fixed value, it is easy to observe that the maximum Sharpe Ratio is achieved when choosing the lowest σ_p and the highest r_p.

Fortunately, MATLAB's function '*estimateMaxSharpeRatio*' estimates the most efficient portfolio maximising the Sharpe ratio when taking the portfolio object as argument of the function [3]:

p = estimateMaxSharpeRatio(p)

3. Method

The objective is to generate three different kinds of portfolios in order to compare different trading strategies. For this, a list of twelve American stocks were selected and historical data was downloaded from Yahoo! Finance. We then used MATLAB to read and analyse the historical data and to generate two different portfolios; a long term portfolio, and a long/short portfolio. The third portfolio is a savings account with a fixed interest rate of 5%.

The first portfolio, the *long term portfolio*, starts with an initial investment of 200,000 SEK. Long portfolio means that the strategy is to purchase shares of each stock and hold them for the period of time under evaluation. This portfolio also has four different scenarios in order to assess different approaches to holding assets for a long period of time; these scenarios are:

- *Equally weighted portfolio* (**EWP**)
- *Equally shares weighted portfolio* (**ESWP**)
- *Randomly weighted portfolio* (**RWP**)
- *Optimal portfolio* (**OP**)

Where, as the names imply, an **EWP** has an equal amount of money invested in every stock; i.e. that the 200,000 SEK is divided equally across all twelve stocks. An **ESWP** has an equal amount of shares purchased with our investment, resulting in an asymmetrical investment across all twelve stocks. Next, the **RWP** was generated by randomising the amount invested in every stock, also causing an asymmetrical investment across all assets. And lastly, the **OP** was generated from the efficient frontier of all the assets, and then optimising the weights by finding the maximum Sharpe Ratio in order to maximise the return and minimise the risk.

Our second portfolio, *the long/short portfolio*, was generated from the same list of stocks as the long term portfolio, but now a simple trading strategy consisting of lagging and leading indicators of simple moving averages (SMAs) was used. The lagging indicator consisted of a 20-days moving average and the leading indicator consisted of a 40-days moving average. When the lagging indicator for each stock crosses the leading indicator it gives a trading signal to short that position in hopes of making a profit out of the stock price heading – technically – to lower prices. Oppositely, when the leading indicator for each stock crosses the lagging indicator, it triggers a buying signal. The weights for this position are the optimal weights calculated from the max Sharpe ratio and the initial investment was also 200,000 SEK.

Lastly, the third portfolio consists of a simple savings account with an initial investment of 200,000 SEK and an interest rate of 5% per annum. And a periodic investment of 10% the initial investment during the 10-year period.

When all portfolios are generated, we can plot their values over time and compare to see which investing strategy would be more profitable during this investment period. Additionally, a plot of all stocks' price paths, returns, as well as cumulative returns were generated for general analysis and understanding of the stocks’ performances throughout the observed period.

4. Results

MATLAB code “I"' found in the appendix at the end of this paper was used to read the data from every Excel file and plotted in a graph of Price-vs-Timeline. Two examples of these plots are shown below in Figure 1.

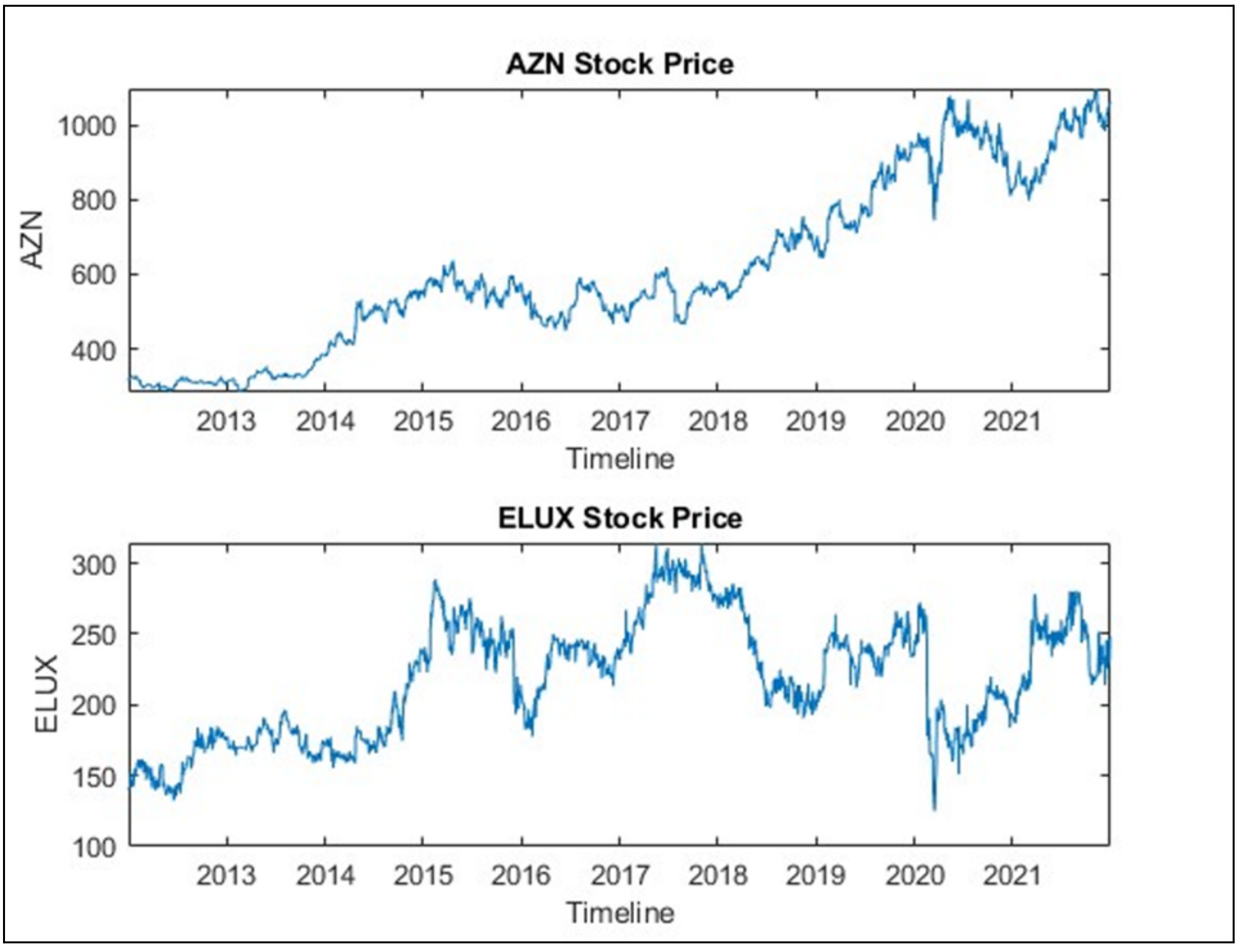

Figure 1: stock prices for AZN and ELUX plots generated with code “I” found in the appendix.

Additionally, the return of every asset was calculated using the same code. These were then plotted in a Returns-vs-Timeline plot. A sample is shown below in Figure 2.

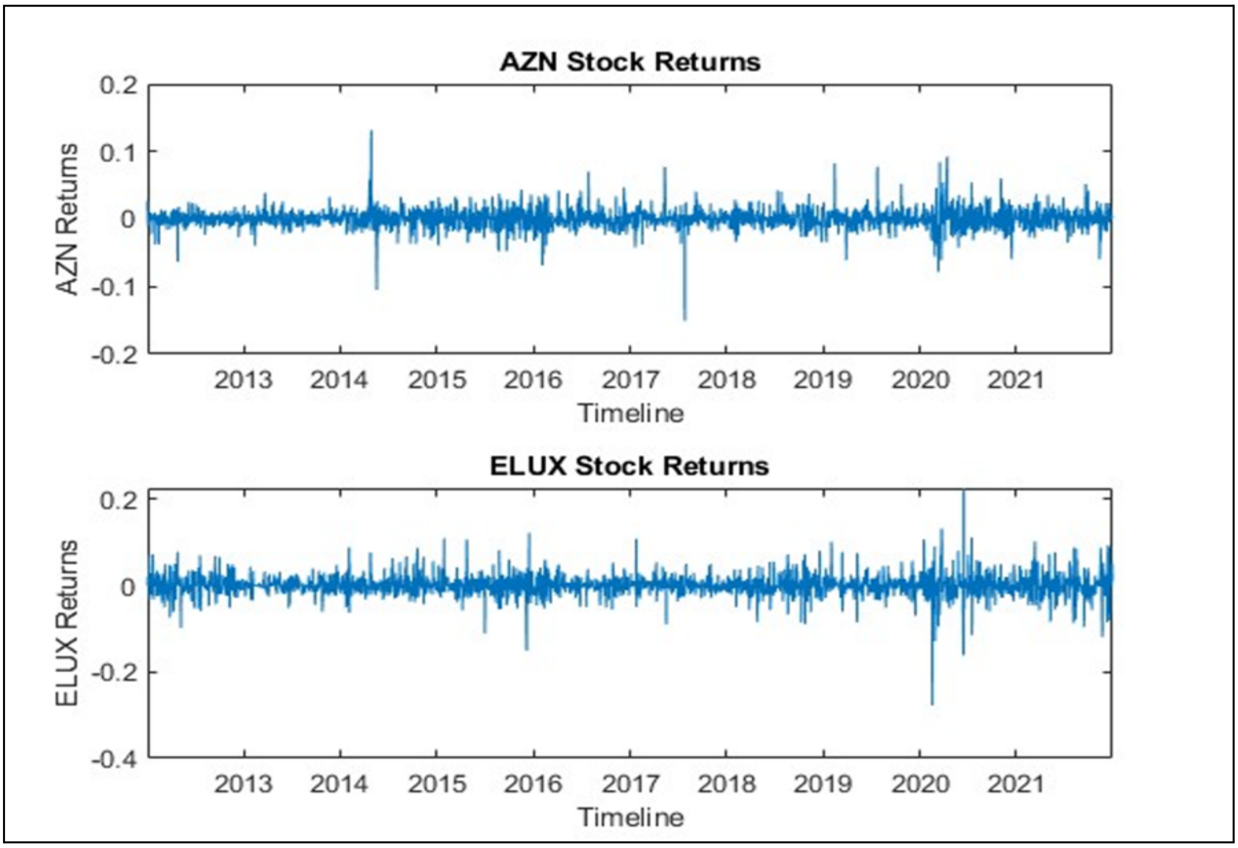

Figure 2: stock returns for AZN and ELUX generated with code "I" found in the appendix.

For the optimal portfolio weights in the long term portfolio, the efficient frontier was calculated and plotted with the MATLAB code "III" presented in the appendix. This is shown in figure 3 below:

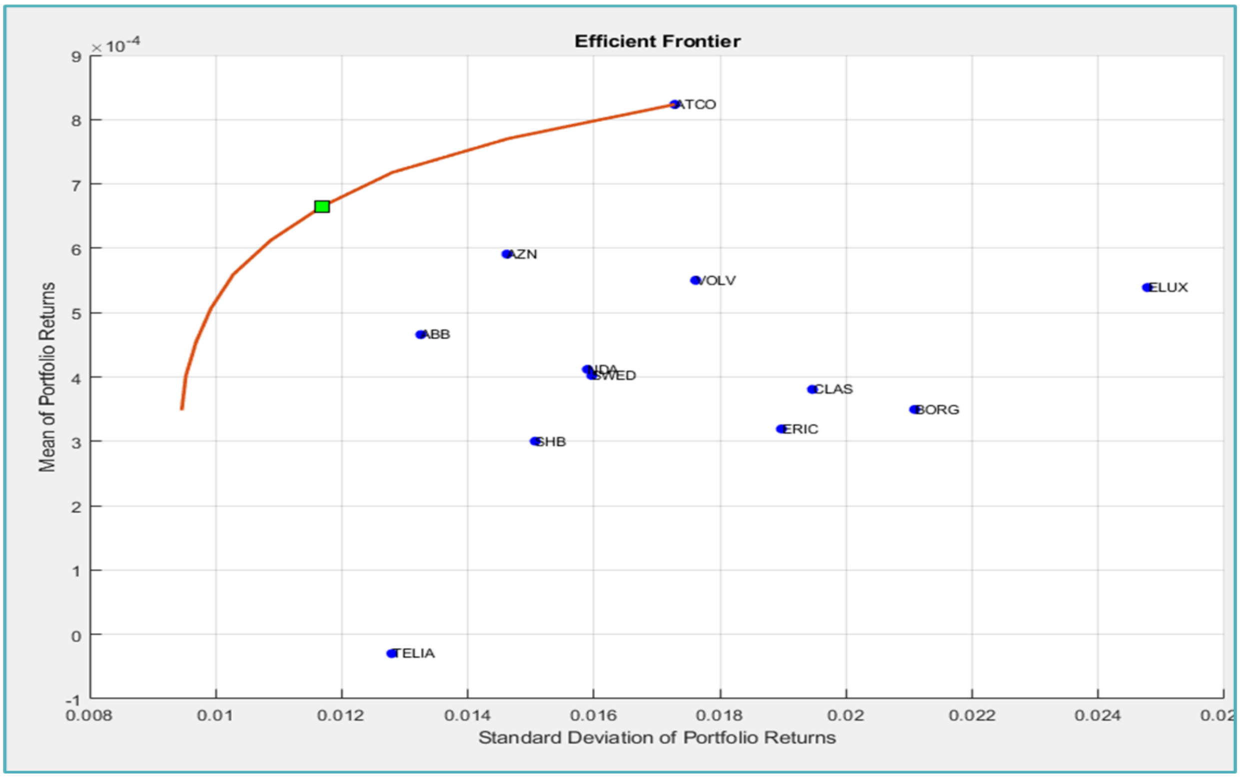

Figure 3: Efficient frontier with optimal weights highlighted in green. Plotted with code "III"

Here, the optimal portfolio weights were calculated from the max Sharpe ratio and the optimal weight were as followed:

ABB	3.5075e-09
ATCO	42.1042
AZN	39.7125
BORG	3.8881
CLAS	4.6117
ELUX	9.6835
ERIC	1.6758e-13
NDA	1.1891e-12
SHB	2.0708e-13
SWED	7.0563e-11
TELIA	5.3208e-14
VOLV	2.0134e-12

Table 1: optimal weights for long term portfolio

With the values on the right column being in percentage and adding up to 1. And the resulting Sharpe risk and return (as plotted by the green square in Figure 3) are:

[sharpeRisk,sharpeRet] = 0.011682507190639 0.000664657952005

With these results, a plot over time for the value in SEK, and a plot for cumulative returns over time for each scenario of the long term portfolio was performed with MATLAB code “II” as shown below in Figure 4 and Figure 5:

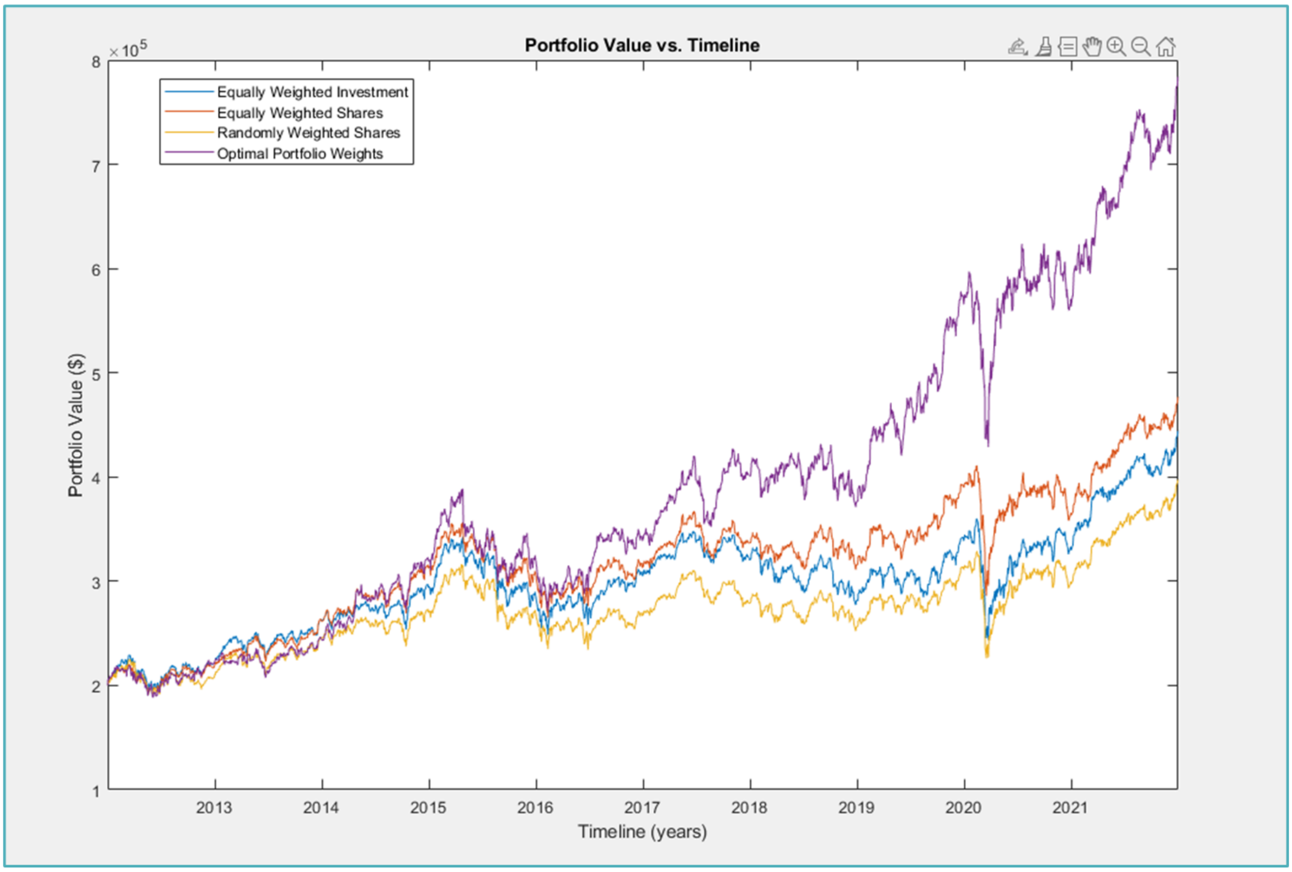

Figure 4: value of long term portfolio scenarios over time. Plotted with MATLAB code "II"

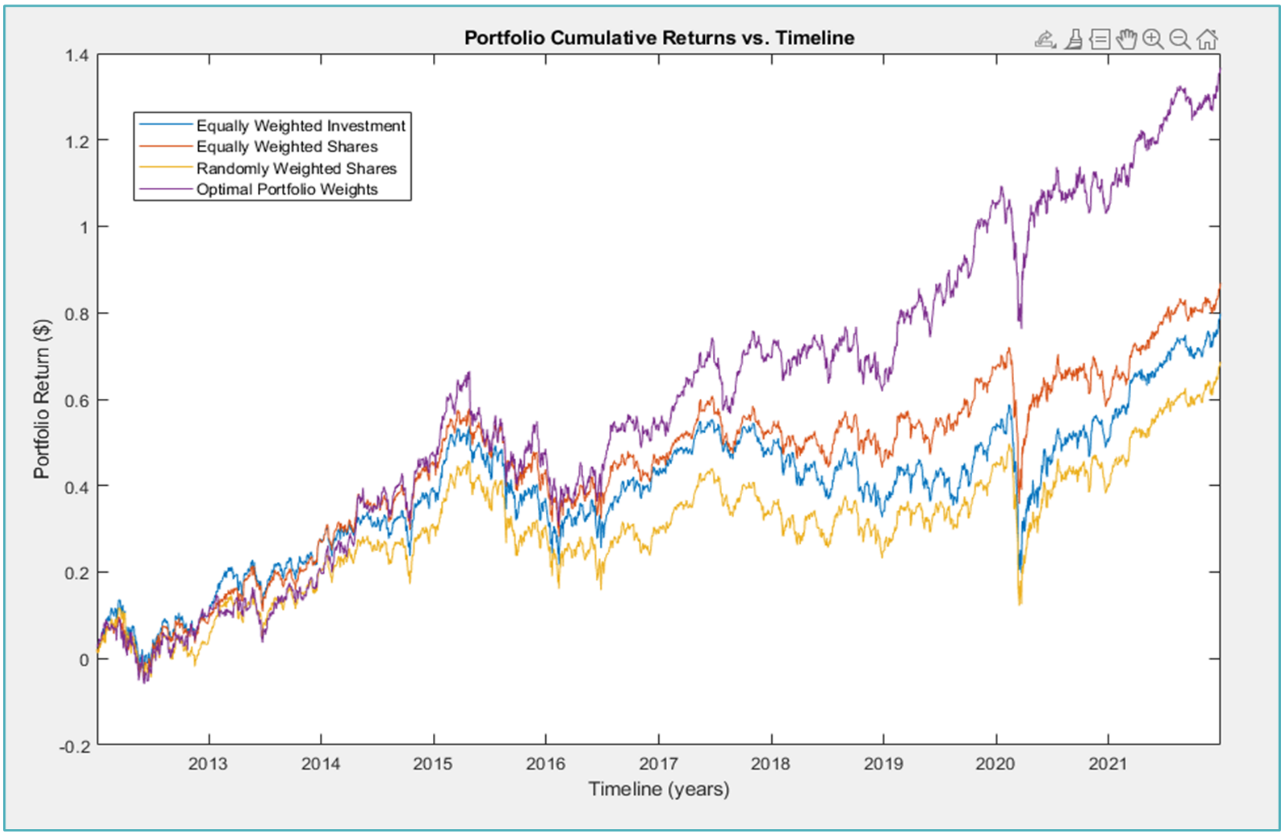

Figure 5: cumulative returns for long term portfolio scenarios over time. Plotted with MATLAB code "II"

The long /short portfolio was generated with MATLAB code “IV” which also plots the SMAs with leading and lagging indicators of every stock in our portfolio with the intent to show when the trading signals are triggered. Bellow is a sample of a stock with its SMAs and the corresponding trading signals:

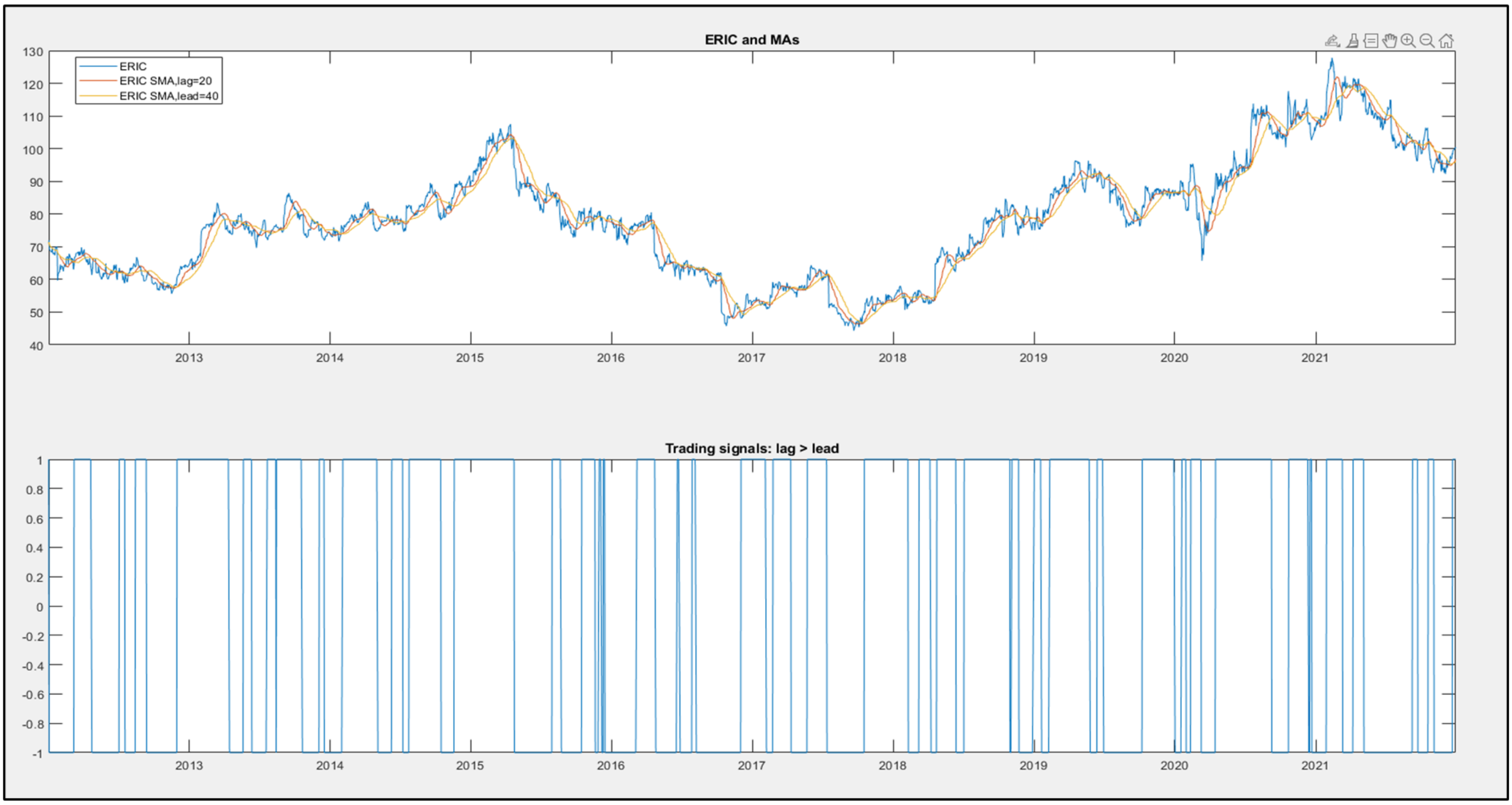

Figure 6: SMAs and trading signals for ERIC generated with MATLAB code “IV”

Additionally, code “IV” also generates a comparison plot of the cumulative log returns of the market and the cumulative log returns of the trading strategy as shown in Figure 7:

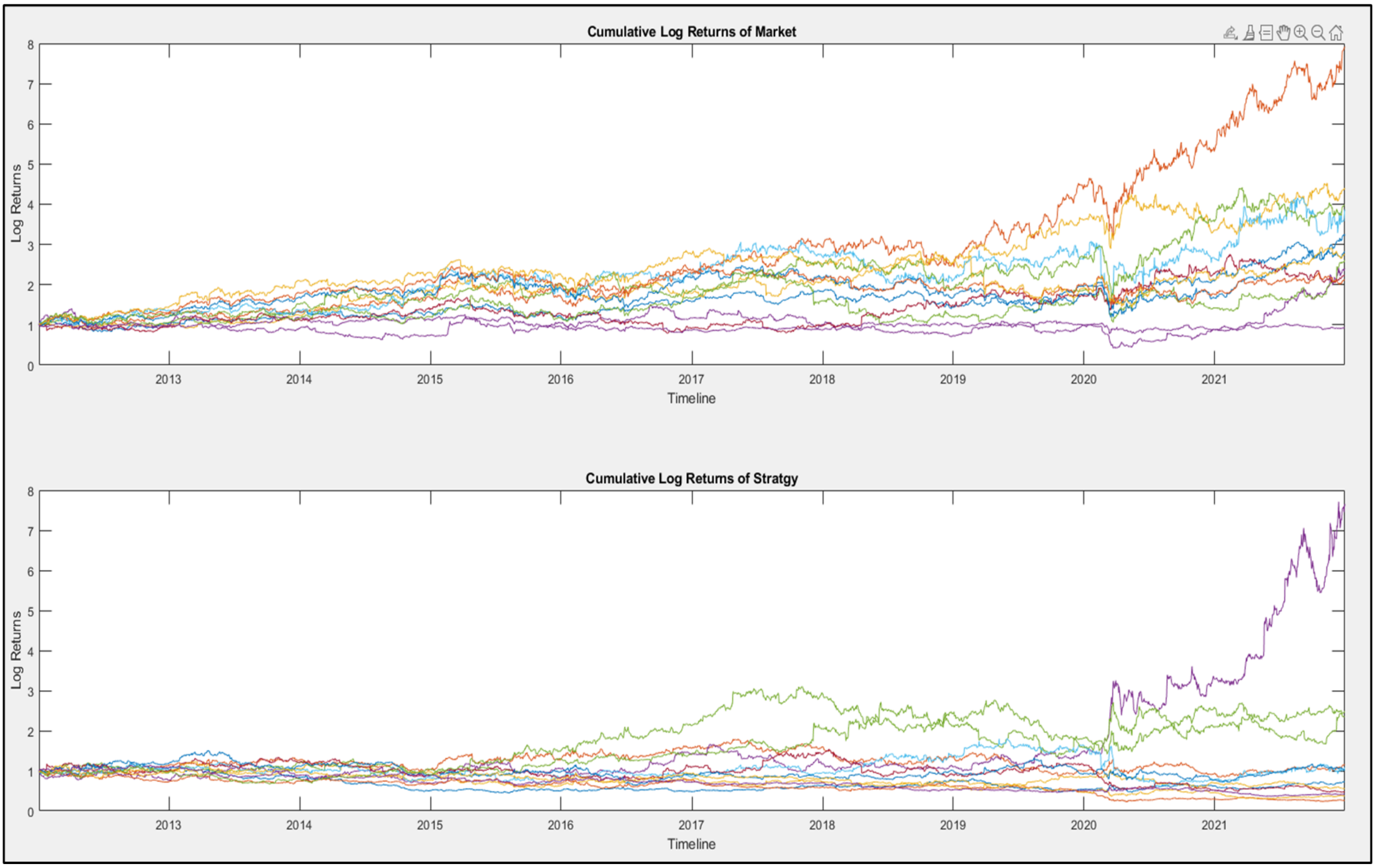

Figure 7: Cumulative Log Returns of Market vs Cumulative Log Returns of Strategy

Also, code "IV" plots the value of the portfolio over time as follows:

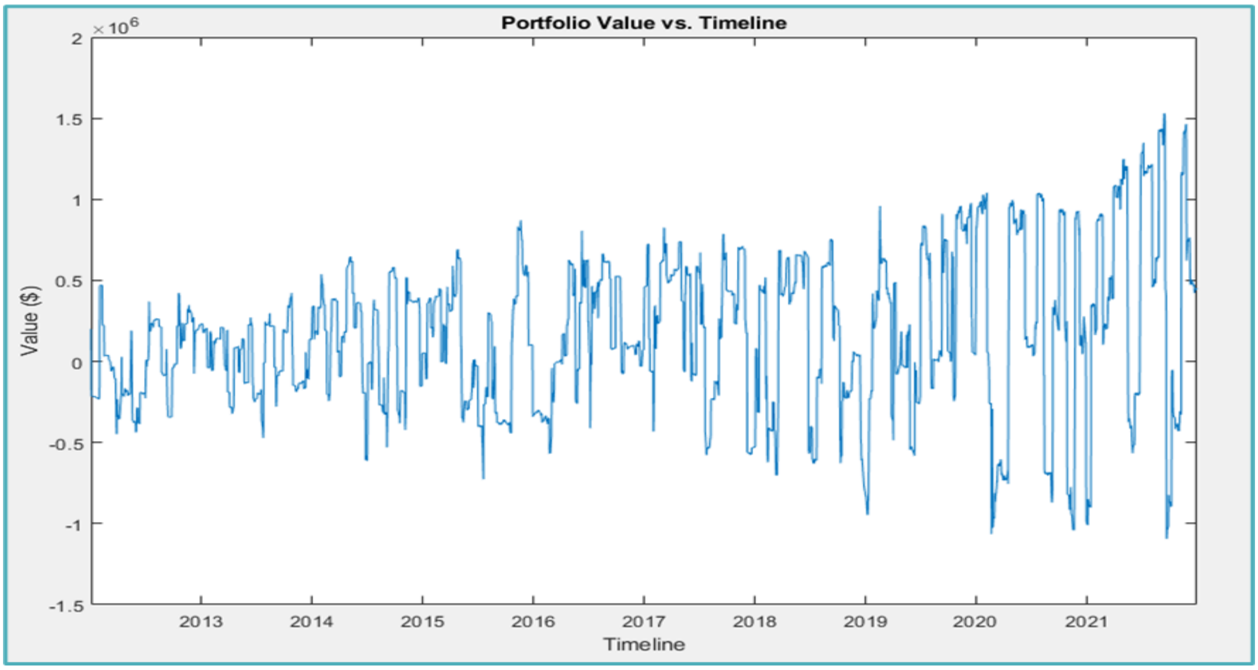

Figure 8: long/short portfolio value over time

Lastly, with help of MATLAB code "V" we can plot the savings account pertaining to our last portfolio as well as a cluttered plot of all portfolios to compare the portfolio values over time and draw conclusions of the investing strategies applied:

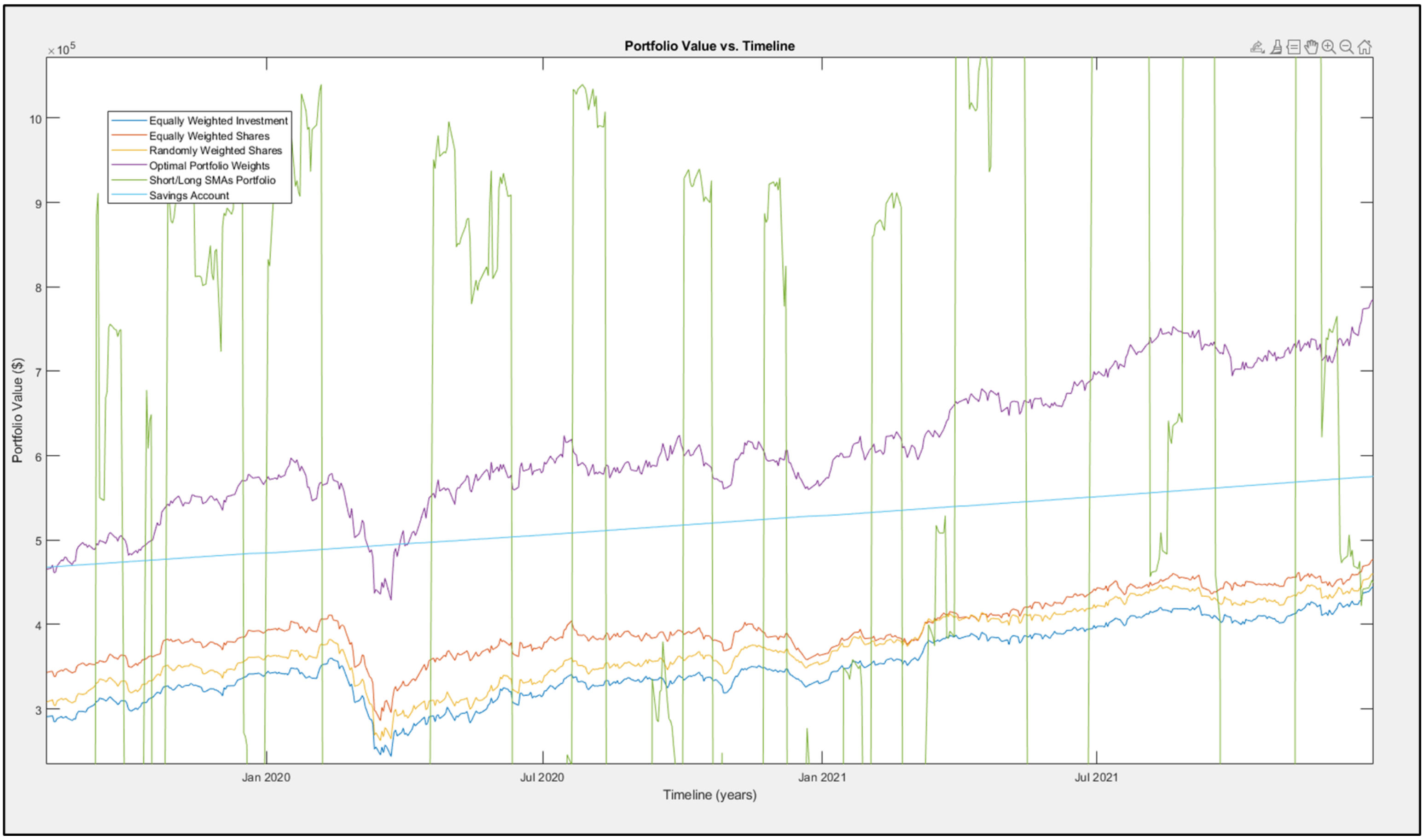

Figure 9: section of all portfolios value over time

5. Conclusion

To conclude, we can confidently say that although these trading strategies are quite simple, they do provide a useful insight to the theory learnt throughout the course and how to use the tools to analyse different financial assets.

Within the different approaches of portfolio 1 (long term portfolio) it is evident that optimising the portfolio by the usage of the efficient frontier and the max Sharpe ratio, the returns of such strategy overcome by far the returns of other strategies, which is as expected by theory. Additionally, although the long/short portfolio shows signs of being *extremely* volatile, it is worth noting that the strategy could be extremely successful when used in shorter periods of time. Lastly, when it comes to the savings account, it is noticeable that the overall return of such investment is quite safe and worthy, but it is important to emphasise that in reality, most interest rates vary over time, and are well below the chosen 5% here. Overall, we can say that the most successful strategy was the long portfolio with optimal weights.

References

[1] C. Tardi. Financial Portfolio: What It Is, and How to Create and Manage One, Investopedia, https://www.investopedia.com/terms/p/portfolio. asp, accessed 13 October 2022.

[2] J. Fernando. Sharpe Ratio Formula and Definition With Examples, Investopedia, https://www.investopedia.com/terms/s/sharperatio.asp, accessed 15 October 2022.

[3] MATLAB. 9.12.0.2009381 (R2022a) Update 4, The MathWorks Inc, 2022.

Appendix

MATLAB CODE “I”

```
%% Data and Stock Prices/Returns plot
% we read all the files
ABB = readtable("ABB.ST.csv","VariableNamingRule","preserve");
ATCO = readtable("ATCO-B.ST.csv","VariableNamingRule","preserve");
AZN = readtable("AZN.ST.csv","VariableNamingRule","preserve");
BORG = readtable("BORG.ST.csv","VariableNamingRule","preserve");
CLAS = readtable("CLAS-B.ST.csv","VariableNamingRule","preserve");
ELUX = readtable("ELUX-A.ST.csv","VariableNamingRule","preserve");
ERIC = readtable("ERIC-A.ST.csv","VariableNamingRule","preserve");
NDA = readtable("NDA-SE.ST.csv","VariableNamingRule","preserve");
SHB = readtable("SHB-A.ST.csv","VariableNamingRule","preserve");
SWED = readtable("SWED-A.ST.csv","VariableNamingRule","preserve");
TELIA = readtable("TELIA.ST.csv","VariableNamingRule","preserve");
VOLV = readtable("VOLV-B.ST.csv","VariableNamingRule","preserve");
% ******* Note that data is from '2012-01-01' to '2022-01-01' (10 years)
AssetList = {'ABB' 'ATCO' 'AZN' 'BORG' 'CLAS' 'ELUX' 'ERIC' 'NDA' 'SHB' 'SWED'
'TELIA' 'VOLV'};
Date = datetime(ABB.Date); % generate time series
Close_Matrix = [ABB.Close ATCO.Close AZN.Close BORG.Close CLAS.Close
ELUX.Close ERIC.Close NDA.Close SHB.Close SWED.Close TELIA.Close VOLV.Close];
dailyReturns = tick2ret(Close_Matrix);
figure(1) % figure contains all stock prices over time
tiledlayout(4,3)
nexttile
plot(Date,ABB.Close)
title('ABB Stock Price')
xlabel('Timeline')
ylabel('ABB')
nexttile
plot(Date,ATCO.Close)
title('ATCO Stock Price')
xlabel('Timeline')
ylabel('ATCO')
nexttile
plot(Date,AZN.Close)
title('AZN Stock Price')
xlabel('Timeline')
ylabel('AZN')
```

```
nexttile
plot(Date,BORG.Close)
title('BORG Stock Price')
xlabel('Timeline')
ylabel('BORG')
nexttile
plot(Date,CLAS.Close)
title('CLAS Stock Price')
xlabel('Timeline')
ylabel('CLAS')
nexttile
plot(Date,ELUX.Close)
title('ELUX Stock Price')
xlabel('Timeline')
ylabel('ELUX')
nexttile
plot(Date,ERIC.Close)
title('ERIC Stock Price')
xlabel('Timeline')
ylabel('ERIC')
nexttile
plot(Date,NDA.Close)
title('NDA Stock Price')
xlabel('Timeline')
ylabel('NDA')
nexttile
plot(Date,SHB.Close)
title('SHB Stock Price')
xlabel('Timeline')
ylabel('SHB')
nexttile
plot(Date,SWED.Close)
title('SWED Stock Price')
xlabel('Timeline')
ylabel('SWED')
nexttile
plot(Date,TELIA.Close)
title('TELIA Stock Price')
xlabel('Timeline')
ylabel('TELIA')
nexttile
plot(Date,VOLV.Close)
title('VOLV Stock Price')
```

```
xlabel('Timeline')
ylabel('VOLV')
figure(2) % figure contains all stock returns over time
tiledlayout(4,3)
nexttile
plot(Date(2:end),tick2ret(ABB.Close))
title('ABB Stock Returns')
xlabel('Timeline')
ylabel('ABB Returns')
nexttile
plot(Date(2:end),tick2ret(ATCO.Close))
title('ATCO Stock Returns')
xlabel('Timeline')
ylabel('ATCO Returns')
nexttile
plot(Date(2:end),tick2ret(AZN.Close))
title('AZN Stock Returns')
xlabel('Timeline')
ylabel('AZN Returns')
nexttile
plot(Date(2:end),tick2ret(BORG.Close))
title('BORG Stock Returns')
xlabel('Timeline')
ylabel('BORG Returns')
nexttile
plot(Date(2:end),tick2ret(CLAS.Close))
title('CLAS Stock Returns')
xlabel('Timeline')
ylabel('CLAS Returns')
nexttile
plot(Date(2:end),tick2ret(ELUX.Close))
title('ELUX Stock Returns')
xlabel('Timeline')
ylabel('ELUX Returns')
nexttile
plot(Date(2:end),tick2ret(ERIC.Close))
title('ERIC Stock Returns')
xlabel('Timeline')
ylabel('ERIC Returns')
nexttile
plot(Date(2:end),tick2ret(NDA.Close))
title('NDA Stock Returns')
xlabel('Timeline')
```

```
ylabel('NDA Returns')
nexttile
plot(Date(2:end),tick2ret(SHB.Close))
title('SHB Stock Returns')
xlabel('Timeline')
ylabel('SHB Returns')
nexttile
plot(Date(2:end),tick2ret(SWED.Close))
title('SWED Stock Returns')
xlabel('Timeline')
ylabel('SWED Returns')
nexttile
plot(Date(2:end),tick2ret(TELIA.Close))
title('TELIA Stock Returns')
xlabel('Timeline')
ylabel('TELIA Returns')
nexttile
plot(Date(2:end),tick2ret(VOLV.Close))
title('VOLV Stock Returns')
xlabel('Timeline')
ylabel('VOLV Returns')
```

MATLAB CODE “II”

```
%% Portfolio 1: LONG PORTFOLIO
invest = 200000;
m = length(AssetList);
[n,~] = size(ABB);
% EWP: equally weighted portfolio
OWP = 1/m*invest*ones(1,m); % investment amount distributed to each asset
NmbrShares_1 = OWP./Close_Matrix(1,:); % number of shares of each asset
Initial_Portf_Value_1 = NmbrShares_1.*Close_Matrix(1,:); % initial portfolio
value of 200000
Portf_Value_1 = zeros(n,1)
for i=1:n
   Portf_Value_1(i) = sum(NmbrShares_1.*Close_Matrix(i,:));
end
Portf_Ret_1 = price2ret(Portf_Value_1);
Portf_cumRet_1 = cumsum(Portf_Ret_1);
% EWP: equally weighted shares portfolio
NmbrShares_2 = invest/sum(Close_Matrix(1,:)); % number of shares of each asset
Initial_Portf_Value_2 = NmbrShares_2.*Close_Matrix(1,:); % initial portfolio
value of 200000
Portf_Value_2 = zeros(n,1)
for i=1:n
   Portf_Value_2(i) = sum(NmbrShares_2.*Close_Matrix(i,:));
end
Portf_Ret_2 = price2ret(Portf_Value_2);
Portf_cumRet_2 = cumsum(Portf_Ret_2);
% RWP: randomly weighted portfolio
nmbrSim = 20000;
for i=1:nmbrSim
   rndm_weight = rand(1,m); % generate random 1xm matrix
   rndm_weight = rndm_weight./sum(rndm_weight); % sum of weights equal 1
   rndm_Invest = rndm_weight*invest; % random investment amount distributed to
each asset
   NmbrShares_3 = rndm_Invest./Close_Matrix(1,:); % number of shares of each
asset
   rndm_Portf_Value(i) = NmbrShares_3*Close_Matrix(end,:)';
end
maximum = max(rndm_Portf_Value)
Portf_Value_3 = zeros(n,1)
for i=1:n
   Portf_Value_3(i) = sum(NmbrShares_3.*Close_Matrix(i,:));
```

```
end
Portf_Ret_3 = price2ret(Portf_Value_3);
Portf_cumRet_3 = cumsum(Portf_Ret_3);
% Optimal Portfolio Weights
p = Portfolio('AssetList',AssetList);
p = estimateAssetMoments(p,dailyReturns); % estimate mean and variance of
portfolio
p = setDefaultConstraints(p); % this means we cannot short the asset and we
need to invest 100% of the money
sharpeWeight = estimateMaxSharpeRatio(p); % estimate sharpe ratio for best
portfolio
[sharpeRisk,sharpeRet] = estimatePortMoments(p,sharpeWeight)
pwgt = estimateFrontierByRisk(p, [0.05, 0.1, 0.20]);
portf_weight = sharpeWeight'.*invest;
NmbrShares_4 = portf_weight./Close_Matrix(1,:); % number of shares of each
asset
Portf_Value_4 = zeros(n,1)
for i=1:n
   Portf_Value_4(i) = sum(NmbrShares_4.*Close_Matrix(i,:));
end
Portf_Ret_4 = price2ret(Portf_Value_4);
Portf_cumRet_4 = cumsum(Portf_Ret_4);
figure(3)
grid on
plot(Date,Portf_Value_1)
hold on
plot(Date,Portf_Value_2)
hold on
plot(Date,Portf_Value_3)
hold on
plot(Date,Portf_Value_4)
hold off
legend("Equally Weighted Investment","Equally Weighted Shares","Randomly
Weighted Shares","Optimal Portfolio Weights",'Location','best')
title("Portfolio Value vs. Timeline")
xlabel("Timeline (years)")
ylabel("Portfolio Value ($)")
figure(4)
grid on
plot(Date(2:end),Portf_cumRet_1)
hold on
plot(Date(2:end),Portf_cumRet_2)
hold on
```

```
plot(Date(2:end),Portf_cumRet_3)
hold on
plot(Date(2:end),Portf_cumRet_4)
hold off
legend("Equally Weighted Investment","Equally Weighted Shares","Randomly Weighted Shares","Optimal Portfolio Weights",'Location','best')
title("Portfolio Cumulative Returns vs. Timeline")
xlabel("Timeline (years)")
ylabel("Portfolio Return ($)")
```

MATLAB CODE “III”

```
%% Plot Efficient Frontier
f = figure;
tabgp = uitabgroup(f); % Define tab group
tab1 = uitab(tabgp,'Title','Efficient Frontier Plot');
ax = axes('Parent',tab1);
[mean,cov] = getAssetMoments(p);
scatter(ax,sqrt(diag(cov)),mean,'blue','filled')
xlabel('Risk')
ylabel('Expected Return')
text(sqrt(diag(cov)),mean,AssetList,'FontSize',8);
hold on
[risk2, ret2] = plotFrontier(p,10);
plot(sharpeRisk,sharpeRet,'s','markers',10,'MarkerEdgeColor','black',...
               'MarkerFaceColor','green');
hold off
tab2 = uitab(tabgp,'Title','Optimal Portfolio Weight'); % Create tab
% Column names and column format
columnname = {'Ticker','Weight (%)'};
columnformat = {'char','numeric'};
% Define the data as a cell array
AssetList = AssetList';
data =
table2cell(table(AssetList(sharpeWeight>0),sharpeWeight(sharpeWeight>0)*100));
% Create the uitable
uit = uitable(tab2, 'Data', data,...
           'ColumnName', columnname,...
           'ColumnFormat', columnformat,...
           'RowName',[]);
% Set width and height
uit.Position(3) = 450; % Widght
uit.Position(4) = 350; % Height
```

MATLAB CODE "IV"

```
%% Portfolio 2: Moving Averages / Long-Short Portfolio
clc;
lag_0 = 20; lead_0 = 40; method = 'Simple';
lag=movavg(Close_Matrix,method,lag_0);
lead=movavg(Close_Matrix,method,lead_0);
Position = (lag >lead)*2-1;
Position(1,:)=1;
Returns = tick2ret(Close_Matrix);
Strategy = Position([2:end],:).*Returns;
Market_CumReturns = exp(cumsum(Returns));
Strategy_CumReturns = exp(cumsum(Strategy));
intRate = 0.05/252;
p3 = Portfolio("AssetList",AssetList,"RiskFreeRate",intRate);
p3 = estimateAssetMoments(p3,Returns)
p3 = setDefaultConstraints(p3);
p3 = setBounds(p3,-0.5,1) % allow shorting
p3 = setBudget(p3,1,1) % invest all the money
sharpeOptWgts = estimateMaxSharpeRatio(p3);
[mean,cov] = getAssetMoments(p);
[riskSharpe,returnSharpe]= estimatePortMoments(p3,sharpeOptWgts);
plotFrontier(p3,50)
hold on
scatter(riskSharpe,returnSharpe,'red','filled')
text(sqrt(diag(cov)),mean,AssetList,'FontSize',8);
x=linspace(0,.55);
y=(returnSharpe-intRate)/riskSharpe*x+intRate;
plot(riskSharpe,returnSharpe,'*')
plot(diag(sqrt(cov)), mean,'s','MarkerSize',10,'MarkerFaceColor','blue')
hold off
figure
subplot(2,1,1)
plot(Date(2:end),Market_CumReturns)
title("Cumulative Log Returns of Market")
xlabel("Timeline")
ylabel("Log Returns")
subplot(2,1,2)
plot(Date(2:end),Strategy_CumReturns)
title("Cumulative Log Returns of Stratgy")
xlabel("Timeline")
ylabel("Log Returns")
```

```
hold off
portf_weight = sharpeOptWgts'.*invest;
NmbrShares_5 = portf_weight./Close_Matrix(1,:); % number of shares of each
asset
Portf_Value_5 = zeros(n,1);
for i=1:n
   for j=1:m
   Portf_Value_5(i) = sum(Position(i,:).*NmbrShares_5.*Close_Matrix(i,:));
   end
end
Portf_Ret_5 = price2ret(Portf_Value_5);
Portf_cumRet_5 = cumsum(Portf_Ret_5);
figure
plot(Date,Portf_Value_5)
title("Portfolio Value vs. Timeline")
xlabel("Timeline")
ylabel("Value ($)")
hold off
figure
grid on
plot(Date,Portf_Value_1)
hold on
plot(Date,Portf_Value_2)
hold on
plot(Date,Portf_Value_3)
hold on
plot(Date,Portf_Value_4)
hold on
plot(Date,Portf_Value_5)
hold off
legend("Equally Weighted Investment","Equally Weighted Shares","Randomly
Weighted Shares","Optimal Portfolio Weights","Short/Long SMAs
Portfolio",'Location','best')
title("Portfolio Value vs. Timeline")
xlabel("Timeline (years)")
ylabel("Portfolio Value ($)")
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,1),lag(:,1),lead(:,1)])
title('ABB and MAs')
legend('ABB','ABB SMA,lag=20','ABB SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,1),'LineWidth',0.5)
```

```
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,2),lag(:,2),lead(:,2)])
title('ATCO and MAs')
legend('ATCO','ATCO SMA,lag=20','ATCO SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,2),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,3),lag(:,3),lead(:,3)])
title('AZN and MAs')
legend('AZN','AZN SMA,lag=20','AZN SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,3),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,4),lag(:,4),lead(:,4)])
title('BORG and MAs')
legend('BORG','BORG SMA,lag=20','BORG SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,4),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,5),lag(:,5),lead(:,5)])
title('CLAS and MAs')
legend('CLAS','CLAS SMA,lag=20','CLAS SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,5),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,6),lag(:,6),lead(:,6)])
title('ELUX and MAs')
legend('ELUX','ELUX SMA,lag=20','ELUX SMA,lead=40','Location','best')
```

```
subplot(2,1,2)
plot(Date,Position(:,6),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,7),lag(:,7),lead(:,7)])
title('ERIC and MAs')
legend('ERIC','ERIC SMA,lag=20','ERIC SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,7),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,8),lag(:,8),lead(:,8)])
title('NDA and MAs')
legend('NDA','NDA SMA,lag=20','NDA SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,8),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,9),lag(:,9),lead(:,9)])
title('SHB and MAs')
legend('SHB','SHB SMA,lag=20','SHB SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,9),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,10),lag(:,10),lead(:,10)])
title('SWED and MAs')
legend('SWED','SWED SMA,lag=20','SWED SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,10),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,11),lag(:,11),lead(:,11)])
```

```
title('TELIA and MAs')
legend('TELIA','TELIA SMA,lag=20','TELIA SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,11),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
figure
subplot(2,1,1)
plot(Date,[Close_Matrix(:,12),lag(:,12),lead(:,12)])
title('VOLV and MAs')
legend('VOLV','VOLV SMA,lag=20','VOLV SMA,lead=40','Location','best')
subplot(2,1,2)
plot(Date,Position(:,12),'LineWidth',0.5)
title('Trading signals: lag > lead')
hold off
```

MATLAB CODE “V”

```
%% SAVINGS ACCOUNT
clc
format bank
rate = 5/100;
savings = 200000;
savingsFV(1)=savings;
payments = savings*0.1;
for i=2:n
   savingsFV(i) = fvfix(rate,i/252,payments,savings,0); % loop daily compound
interest
end
savingsFV = savingsFV'; % transpose matrix
plot(Date,savingsFV)
title("Savings Account")
xlabel("Timeline")
ylabel("Value")
figure
grid on
plot(Date,Portf_Value_1)
hold on
plot(Date,Portf_Value_2)
hold on
plot(Date,Portf_Value_3)
hold on
plot(Date,Portf_Value_4)
hold on
plot(Date,Portf_Value_5)
hold on
plot(Date,savingsFV)
hold off
legend("Equally Weighted Investment","Equally Weighted Shares","Randomly
Weighted Shares","Optimal Portfolio Weights","Short/Long SMAs
Portfolio","Savings Account",'Location','best')
title("Portfolio Value vs. Timeline")
xlabel("Timeline (years)")
ylabel("Portfolio Value ($)")
```