APS 502 Financial Engineering Project

Hruday Vishal Kanna Anand

ID: 1006874517

**Problem 1-**

**Part 1-**

The objective of part 1 is to minimize the total price of all the bonds to be purchased- that is to minimize the sum of the products of bonds with unit price (Pi) and number of units (xi) .

* min

The objective is then subjected to the constraints shown below:

1. >= 𝐿𝑡 for t= 1:6

Here in our scenario t ranges from 1:6; Ct is a matrix of all the cashflows of all 13 bonds at every time interval; is the cash overflow from time t; is the forward rate from tie t-1 to time t; Lt is the liabilities we must meet at time t; xi is the amount of each bond we buy.

1. xi, >= 0

this imposes a no short sell on the bonds as all xi, are either positive or zero.

**important inputs to the optimization program-**

1. s=[0.01 0.015 0.02 0.025 0.03 0.035]; (spot rates)
2. the table of bonds- cashflow, pricing and rating-

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **bond** | **1** | **2** | **3** | **4** | **5** | **6** | **price** | **rating** |
| 1 | 10 | 10 | 10 | 10 | 10 | 110 | 108 | B |
| 2 | 7 | 7 | 7 | 7 | 7 | 107 | 94 | B |
| 3 | 8 | 8 | 8 | 8 | 8 | 108 | 99 | B |
| 4 | 6 | 6 | 6 | 6 | 106 |  | 92.7 | B |
| 5 | 7 | 7 | 7 | 7 | 107 |  | 96.6 | B |
| 6 | 6 | 6 | 6 | 106 |  |  | 95.9 | B |
| 7 | 5 | 5 | 5 | 105 |  |  | 92.9 | A |
| 8 | 10 | 10 | 110 |  |  |  | 110 | A |
| 9 | 8 | 8 | 108 |  |  |  | 104 | A |
| 10 | 6 | 6 | 106 |  |  |  | 101 | A |
| 11 | 10 | 110 |  |  |  |  | 107 | A |
| 12 | 7 | 107 |  |  |  |  | 102 | A |
| 13 | 100 |  |  |  |  |  | 95.2 | A |

We import the cashflow matrix Ct or ‘a’ (in MATLAB file) and the array price from this table.

1. Liabilities-

liability=[ 500 200 800 400 700 900]';

1. Forward rates calculated using spot rates-



1. Modified cash flow matrix - a – with forward rates

**Part 2-**

We subject the optimization to one additional condition-

* <= 0.5 \*

This condition makes sure that at most 50% of the portfolios values comes from bonds with a B rating.

**Part 3-**

We modify the additional condition we subjected the optimization to-

* <= 0.25 \*

This condition makes sure that at most 25% of the portfolios values comes from bonds with a B rating.

**Outputs-**

Optimal bond portfolio considering various cases stated above-

|  |  |  |  |
| --- | --- | --- | --- |
| **bonds** | **Portfolio-1 (Part-1)** | **Portfolio-2**  **(Part-2)** | **Portfolio-3**  **(Part-3)** |
| 1 | 8.1818 | 0 | 0 |
| 2 | 0 | 8.4112 | 7.1267 |
| 3 | 0 | 0 | 0 |
| 4 | 0 | 5.736 | 0 |
| 5 | 5.7774 | 0 | 0 |
| 6 | 2.6202 | 0 | 0 |
| 7 | 0 | 3.2212 | 10.4052 |
| 8 | 0 | 0 | 0 |
| 9 | 6.1298 | 6.3944 | 6.4638 |
| 10 | 0 | 0 | 0 |
| 11 | 0.118 | 0.3586 | 0.4216 |
| 12 | 0 | 0 | 0 |
| 13 | 3.118 | 3.3586 | 3.4216 |
| Z1 | 0 | 0 | 0 |
| Z2 | 0 | 0 | 0 |
| Z3 | 0 | 0 | 0 |
| Z4 | 0 | 31.5239 | 742.4305 |
| Z5 | 0 | 0 | 129.6208 |
| Total Cost | 2.6400E+03 | 2.6448e+03 | 2.6796e+03 |
| Total Cost | 2640 | 2644.8 | 2679.6 |

Z1-Z5 are the amounts of cash carried forward at any given time. E.g., Z1 is the amount of cash carried over at time 1.

From the results we can see that portfolio-1 has the least cost with **2640** and portfolio -3 has the most cost with **2679.6**. Portfolio 2 has a slightly higher cost than portfolio-1 at **2644.8**. From the results obtained we can say that portfolios with more ‘A’ rated bonds will cost higher.

**Ranking-**

Portfolio-1: **Rank 1** has the lowest cost.

Portfolio-2: **Rank 2**

Portfolio-3: **Rank 3** has the highest cost.

**MATLAB Code and Outputs-**

%loading in the spot rates into a variable s  
s=[0.01 0.015 0.02 0.025 0.03 0.035];  
  
%creating a forward rate matrix 'f' using the spot rates from above  
f= zeros(6,6);  
for i=1:6  
 for j=i:6  
 f(i,j)= (((1+s(j))^(j)/(1+s(i))^(i))^(1/(j-i)))-1;  
 end  
end  
  
% importing the bond price array price array and adding the carry over  
% elements to it  
for i=14:18  
 price(i)=1;  
end  
  
%creating our liabilities variable for the cash needed at various time  
%intervals  
liability=[ 500 200 800 400 700 900]';  
  
%creating our limitations and conditions for our model  
lb=zeros(18,1);  
ub=ones(18,1)\*inf;  
aeq=[];  
beq=[];  
%variable a is the imported cash flow of all the bonds  
a=tablebonds';  
% carry over cash and forward rates are added to the model  
for i=14:18  
 a(:,i)=0;  
 a(i-13,i)=-1;  
 a(i-12,i)=1+f(i-13,i-12);  
end  
  
% PART-1:the linprog optimization model is used  
[x,fval] = linprog(price,-a,-liability,aeq,beq,lb,ub)  
  
%calculate total price  
total= x(1:13)' \* price(1:13)  
  
  
% PART-2:the linprog optimization model is used along with an additional  
% condition- at most 50% of the portfolio value can be in B rated bonds  
liability(7)=0;  
for i=1:18  
 if i<=6  
 a(7,i)= -0.5\*price(i,1);  
 end  
 if (i>=7 & i<=13)  
 a(7,i)= 0.5\*price(i,1);  
 end  
 if i>=14  
 a(7,i)=0;  
 end  
end  
  
[x2,fval2] = linprog(price,-a,-liability,aeq,beq,lb,ub)  
  
%calculate total price  
total= x2(1:13)' \* price(1:13)  
  
  
% PART-3:the linprog optimization model is used along with an additional  
% condition- at most 25% of the portfolio value can be in B rated bonds  
for i=1:18  
 if i<=6  
 a(7,i)= -0.75\*price(i,1);  
 end  
 if (i>=7 & i<=13)  
 a(7,i)= 0.25\*price(i,1);  
 end  
 if i>=14  
 a(7,i)=0;  
 end  
end  
[x3,fval3] = linprog(price,-a,-liability,aeq,beq,lb,ub)  
  
%calculate total price  
total= x3(1:13)' \* price(1:13)

Optimal solution found.  
  
x =  
  
 8.1818  
 0  
 0  
 0  
 5.7774  
 2.6202  
 0  
 0  
 6.1298  
 0  
 0.1180  
 0  
 3.1180  
 0  
 0  
 0  
 0  
 0  
  
fval =  
  
 2.6400e+03  
  
total =  
  
 2.6400e+03  
  
Optimal solution found.  
  
x2 =  
  
 0  
 8.4112  
 0  
 5.7360  
 0  
 0  
 3.2212  
 0  
 6.3944  
 0  
 0.3586  
 0  
 3.3586  
 0  
 0  
 0  
 31.5239  
 0  
  
fval2 =  
  
 2.6763e+03  
  
total =  
  
 2.6448e+03  
  
Optimal solution found.  
  
x3 =  
  
 0  
 7.1267  
 0  
 0  
 0  
 0  
 10.4052  
 0  
 6.4638  
 0  
 0.4216  
 0  
 3.4216  
 0  
 0  
 0  
 742.4305  
 129.6208  
  
fval3 =  
  
 3.5517e+03  
  
total =  
  
 2.6796e+03

[*Published with MATLAB® R2017a*](http://www.mathworks.com/products/matlab)

**Problem 2-**

**Part-1-**

For this problem we consider, SPY stock as **asset a**, GOVT stock as **asset b**, and EEMV stock as **asset c.**

The monthly return on each asset is calculated as-

* -1=

where is the adjusted closing price of the asset at time t and is the adjusted closing price of the asset one time period before. Here is in the form of a decimal, if we need percentage value, we multiply by 100.

Next we calculate the arithmetic average return , the geometric average return , and covariance / standard deviation . with the formulas sated below-

The objective here is to minimize the overall portfolio variance while subjecting it to various conditions and boundaries.

* Min (optimize)

Subjected to-

1. (when short selling is not allowed)
2. = R

Here R is the expected return of the portfolio, we vary the range of R from the lowest expected return to the highest expected return of the individual assets and form an efficient frontier along with a table of all the weights, expected returns, and standard deviations of the portfolios.

**Input to the optimization program-**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SPY (a)** | **GOVT (b)** | **EEMV (c)** |
| Date | Adj Close | Adj Close | Adj Close |
| 1/1/2014 | 154.9601 | 21.94003 | 44.75371 |
| 2/1/2014 | 162.0132 | 21.98463 | 46.35922 |
| 3/1/2014 | 162.6394 | 21.89809 | 47.83094 |
| 4/1/2014 | 164.4928 | 22.05971 | 48.7006 |
| 5/1/2014 | 168.3101 | 22.23335 | 49.83784 |
| 6/1/2014 | 170.9657 | 22.19321 | 50.23085 |
| 7/1/2014 | 169.4764 | 22.16896 | 51.32072 |
| 8/1/2014 | 176.1646 | 22.43352 | 52.97924 |
| 9/1/2014 | 172.9259 | 22.28562 | 50.60148 |
| 10/1/2014 | 177.8258 | 22.51465 | 51.24457 |
| 11/1/2014 | 182.711 | 22.70645 | 50.15299 |
| 12/1/2014 | 181.2472 | 22.6499 | 47.91908 |
| 1/1/2015 | 176.8477 | 23.36556 | 49.2209 |
| **Continues till 1/1/2021** |  |  |  |

Each column containing the adjusted closing price of the assets is imported into MATLAB using the import function, as a column vector.

**Outputs of the program-**

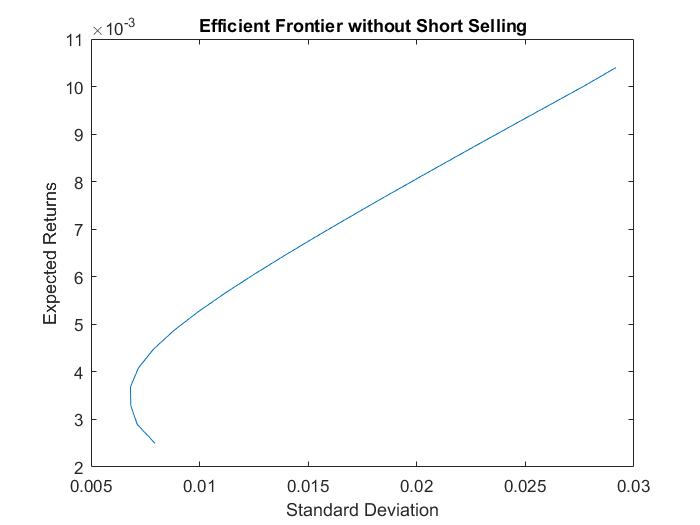
1. Table containing the expected returns and standard deviation of all the assets-

|  |  |  |  |
| --- | --- | --- | --- |
|  | **asset a** | **asset b** | **asset c** |
| **Arithmetic Mean of returns** | 0.0112 | 0.0026 | 0.0046 |
| **Expected Returns** | 0.0104 | 0.0025 | 0.0038 |
| **Standard Deviation** | 0.0413 | 0.0112 | 0.038 |

1. Covariance matrix-

|  |  |  |  |
| --- | --- | --- | --- |
| **Covariance Matrix** | **Asset a** | **Asset b** | **Asset c** |
| **Asset a** | 0.0017 | -0.0001 | 0.0011 |
| **Asset b** | -0.0001 | 0.0001 | 0 |
| **Asset c** | 0.0011 | 0 | 0.0014 |

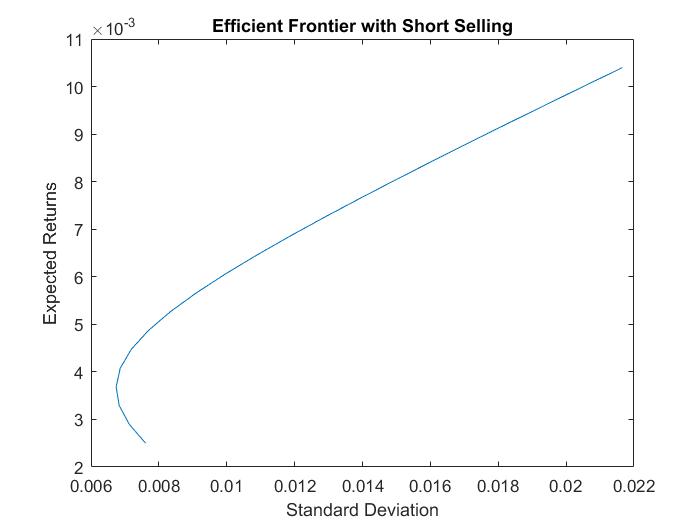
1. Efficient Frontier without short selling-



1. Table of optimal weights created without short selling-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Expected Returns** | **Variance** | **W1 (a)** | **W2 (b)** | **W3 (c)** |
| 1 | 0.0025 | 6.30703E-05 | 1.29653E-17 | 1 | 2.21424E-15 |
| 2 | 0.002895 | 5.08352E-05 | 0.041981142 | 0.909995243 | 0.048023615 |
| 3 | 0.00329 | 4.66811E-05 | 0.099522015 | 0.896411462 | 0.004066523 |
| 4 | 0.003685 | 4.64619E-05 | 0.150350329 | 0.849649165 | 5.05458E-07 |
| 5 | 0.00408 | 5.15539E-05 | 0.200493466 | 0.799500206 | 6.32793E-06 |
| 6 | 0.004475 | 6.19627E-05 | 0.250638076 | 0.74935853 | 3.3944E-06 |
| 7 | 0.00487 | 7.76886E-05 | 0.300782686 | 0.699216856 | 4.5775E-07 |
| 8 | 0.005265 | 9.87323E-05 | 0.350926857 | 0.649073005 | 1.38421E-07 |
| 9 | 0.00566 | 0.000125094 | 0.401070988 | 0.598928963 | 4.92271E-08 |
| 10 | 0.006055 | 0.000156772 | 0.451215109 | 0.54878487 | 2.05125E-08 |
| 11 | 0.00645 | 0.000193778 | 0.501355352 | 0.498621608 | 2.30408E-05 |
| 12 | 0.006845 | 0.000236086 | 0.551502084 | 0.44849042 | 7.49683E-06 |
| 13 | 0.00724 | 0.000283716 | 0.601646579 | 0.398348178 | 5.24238E-06 |
| 14 | 0.007635 | 0.000336664 | 0.651790863 | 0.348204891 | 4.24579E-06 |
| 15 | 0.00803 | 0.000394929 | 0.70193522 | 0.298061963 | 2.817E-06 |
| 16 | 0.008425 | 0.000458513 | 0.752079351 | 0.247917917 | 2.73196E-06 |
| 17 | 0.00882 | 0.000527414 | 0.802223465 | 0.197773792 | 2.74262E-06 |
| 18 | 0.009215 | 0.000601632 | 0.852367619 | 0.147629862 | 2.51842E-06 |
| 19 | 0.00961 | 0.000681166 | 0.902512132 | 0.097487703 | 1.65025E-07 |
| 20 | 0.010005 | 0.000766019 | 0.952656266 | 0.047343678 | 5.52275E-08 |
| 21 | 0.0104 | 0.000851014 | 1.000000229 | 2.22048E-16 | 2.22048E-16 |

1. Efficient Frontier with short selling-



1. Table of optimal weights created with short selling-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Expected Returns** | **Variance** | **W1 (a)** | **W2 (b)** | **W3 (c)** |
| 1 | 0.0025 | 5.78974E-05 | -0.015663409 | 0.923066576 | 0.092596832 |
| 2 | 0.002895 | 5.08352E-05 | 0.041984372 | 0.910011208 | 0.048004421 |
| 3 | 0.00329 | 4.66809E-05 | 0.099632152 | 0.896955839 | 0.003412009 |
| 4 | 0.003685 | 4.54344E-05 | 0.157279932 | 0.88390047 | -0.041180402 |
| 5 | 0.00408 | 4.70958E-05 | 0.214927712 | 0.870845101 | -0.085772814 |
| 6 | 0.004475 | 5.16651E-05 | 0.272575493 | 0.857789733 | -0.130365225 |
| 7 | 0.00487 | 5.91423E-05 | 0.330223273 | 0.844734364 | -0.174957637 |
| 8 | 0.005265 | 6.95274E-05 | 0.387871053 | 0.831678995 | -0.219550048 |
| 9 | 0.00566 | 8.28203E-05 | 0.445518834 | 0.818623626 | -0.26414246 |
| 10 | 0.006055 | 9.90211E-05 | 0.503166614 | 0.805568258 | -0.308734871 |
| 11 | 0.00645 | 0.00011813 | 0.560814394 | 0.792512889 | -0.353327283 |
| 12 | 0.006845 | 0.000140146 | 0.618462174 | 0.77945752 | -0.397919695 |
| 13 | 0.00724 | 0.000165071 | 0.676109955 | 0.766402151 | -0.442512106 |
| 14 | 0.007635 | 0.000192903 | 0.733757735 | 0.753346783 | -0.487104518 |
| 15 | 0.00803 | 0.000223643 | 0.791405515 | 0.740291414 | -0.531696929 |
| 16 | 0.008425 | 0.000257292 | 0.849053296 | 0.727236045 | -0.576289341 |
| 17 | 0.00882 | 0.000293848 | 0.906701076 | 0.714180676 | -0.620881752 |
| 18 | 0.009215 | 0.000333311 | 0.964348856 | 0.701125308 | -0.665474164 |
| 19 | 0.00961 | 0.000375683 | 1.021996637 | 0.688069939 | -0.710066575 |
| 20 | 0.010005 | 0.000420963 | 1.079644417 | 0.67501457 | -0.754658987 |
| 21 | 0.0104 | 0.00046915 | 1.137292197 | 0.661959201 | -0.799251398 |

**Comparison of the various portfolios using returns from February 2021:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Asset a** | **Asset b** | **Asset c** |
| Returns in Feb 2021 | 0.0278 | -0.0200 | 0.0099 |

The portfolios we will be considering are the minimum variance portfolio with short selling, minimum variance portfolio without short selling, equally weighted portfolio, and a portfolio with 60% asset a, 30% asset b and 10% asset c.

Table consisting weights of minimum variance portfolio-

|  |  |  |
| --- | --- | --- |
| **Weights** | **Minimum Variance Portfolio** | |
| **without short** | **with short** |
| W1 | 0.1273 | 0.1532 |
| W2 | 0.8725 | 0.8848 |
| W3 | 0.0001 | -0.038 |

Table consisting of the expected returns of the various portfolios-

|  |  |
| --- | --- |
| **Portfolios** | **Expected Returns** |
| the minimum variance portfolio with short selling | -0.0138 |
| minimum variance portfolio without short selling | -0.0139 |
| equally weighted | 0.0059 |
| 60% asset a, 30% asset b and 10% asset c | 0.0117 |

From the table we can see that the minimum variance portfolios have the least expected returns with **-0.0138** (with short selling) and **-0.0139**(without short selling). This is because the minimum variance portfolio is **weighted heavily on asset b** as its variance and covariance with other assets is very low**.**  The low variance and covariance attributes to very low expected returns, which in turn caused our portfolio to perform very poorly.This assethas a negative expected return in Feb 2021.

The next portfolio is the equally weighted portfolio with an expected return of **0.0059.** The portfolio with the most expected return is the one we weighted ourselves with an expected return of **0.0117.** This is because 60% is invested into SPY which has the highest expected return of all three assets as it mimics the performance of S&P 500.

**Ranking-**

minimum variance portfolio without short selling- **rank 4** (lowest expected returns)

the minimum variance portfolio with short selling- **rank 3**

equally weighted portfolio- **rank 2**

Manually weighted (60% asset a, 30% asset b and 10% asset c)- **rank 1** (highest expected returns)

**MATLAB Code and Outputs-**

% Calculate arithimatic and geometric mean return for asset a, SPY is  
% imported into the program  
for i=2:85  
 return\_a(i-1)= (SPY(i,1)/SPY(i-1,1))-1;  
end  
mean\_return\_a=sum(return\_a)/84  
geo\_mean\_a=((prod(return\_a+1))^(1/84))-1  
  
  
% Calculate arithimatic and geometric mean return for asset b, GOVT is  
% imported into the program  
for i=2:85  
 return\_b(i-1)= (GOVT(i,1)/GOVT(i-1,1))-1;  
end  
mean\_return\_b=sum(return\_b)/84  
geo\_mean\_b=((prod(return\_b+1))^(1/84))-1  
  
  
% Calculate arithimatic and geometric mean return for asset c, EEMV is  
% imported into the program  
for i=2:85  
 return\_c(i-1)= (EEMV(i,1)/EEMV(i-1,1))-1;  
end  
mean\_return\_c=sum(return\_c)/84  
geo\_mean\_c=((prod(return\_c+1))^(1/84))-1  
  
% Calculating Varience and Standard Deviation of all 3 assets  
aVar=sum((return\_a-mean\_return\_a).^2)/84;  
bVar=sum((return\_b-mean\_return\_b).^2)/84;  
cVar=sum((return\_c-mean\_return\_c).^2)/84;  
  
aSTD=sqrt(aVar)  
bSTD=sqrt(bVar)  
cSTD=sqrt(cVar)  
  
% creating the Covarience matrix for all 3 assets  
abCOV=sum((return\_a-mean\_return\_a).\*(return\_b-mean\_return\_b))/84;  
acCOV=sum((return\_a-mean\_return\_a).\*(return\_c-mean\_return\_c))/84;  
bcCOV=sum((return\_b-mean\_return\_b).\*(return\_c-mean\_return\_c))/84;  
  
COV=[aVar,abCOV,acCOV;abCOV,bVar,bcCOV;acCOV,bcCOV,cVar]  
  
%Setting bounds and restricting conditions for the optimization including  
%short selling  
A=-[0,0,0];  
B=-[0];  
RN=0;  
aeq=[1 1 1; geo\_mean\_a,geo\_mean\_b,geo\_mean\_c];  
Beq=[1; RN];  
Ub=[inf,inf,inf];  
Lb=[-inf, -inf, -inf];  
F=[0 0 0]';  
  
% Creating the optimized table with the Expected return, Standard  
% Deviation, and weights of all assets  
i=1;  
for RN =0.0025:0.000395:0.0104  
 Beq=[1; RN];  
 K(i,1)=RN;  
 [x,fval]=quadprog(COV,F,A,B,aeq,Beq, Lb, Ub);  
 K(i,2)=fval;  
 K(i,3)=x(1);  
 K(i,4)=x(2);  
 K(i,5)=x(3);  
 i=i+1;  
end  
K(:,2)=sqrt(K(:,2));  
  
%Plotting the Efficient Frontier  
figure;  
plot(K(:,2),K(:,1))  
title('Efficient Frontier with Short Selling');  
  
%Modifing bounds and restricting conditions for the optimization without  
%short selling  
Lb=[0, 0, 0];  
  
%creating the optimized table again  
i=1;  
for RN =0.0025:0.000395:0.0104  
 Beq=[1; RN];  
 K(i,1)=RN;  
 [x,fval]=quadprog(COV,F,A,B,aeq,Beq, Lb, Ub);  
 K(i,2)=fval;  
 K(i,3)=x(1);  
 K(i,4)=x(2);  
 K(i,5)=x(3);  
 i=i+1;  
end  
K(:,2)=sqrt(K(:,2));  
  
%Plotting the Efficient Frontier  
figure;  
plot(K(:,2),K(:,1))  
title('Efficient Frontier without Short Selling');  
  
  
%calculating reutuns for FEB 2021  
feb\_closing=[379.12 26.52 62.29];  
feb\_returns(1)=(feb\_closing(1)/SPY(85))-1;  
feb\_returns(2)=(feb\_closing(2)/GOVT(85))-1;  
feb\_returns(3)=(feb\_closing(3)/EEMV(85))-1;  
feb\_returns;  
  
  
%minimum varience portfolio with short selling  
Lb=[-inf, -inf, -inf];  
aeq=[1 1 1];  
Beq=[1];  
[x,fval]=quadprog(COV,F,A,B,aeq,Beq, Lb, Ub);  
Rmins=sum(feb\_returns.\*x.')  
  
  
%minimum varience portfolio without short selling  
Lb=[0, 0, 0];  
[x,fval]=quadprog(COV,F,A,B,aeq,Beq, Lb, Ub);  
Rmin=sum(feb\_returns.\*x.')  
  
  
%Equally weighted portfolio  
W=[0.333 0.333 0.333];  
Req=sum(feb\_returns.\*W)  
  
  
%60%,30%,10% weighted portfolio  
W=[0.6 0.3 0.1];  
Rwt=sum(feb\_returns.\*W)

mean\_return\_a =  
  
 0.0112  
  
geo\_mean\_a =  
  
 0.0104  
  
mean\_return\_b =  
  
 0.0026  
  
geo\_mean\_b =  
  
 0.0025  
  
mean\_return\_c =  
  
 0.0046  
  
geo\_mean\_c =  
  
 0.0038  
  
aSTD =  
  
 0.0413  
  
bSTD =  
  
 0.0112  
  
cSTD =  
  
 0.0380  
  
COV =  
  
 0.0017 -0.0001 0.0011  
 -0.0001 0.0001 -0.0000  
 0.0011 -0.0000 0.0014  
  
Minimum found that satisfies the constraints.  
  
Optimization completed because the objective function is non-decreasing in   
feasible directions, to within the default value of the optimality tolerance,  
and constraints are satisfied to within the default value of the constraint tolerance.  
  
Minimum found that satisfies the constraints.  
  
Optimization completed because the objective function is non-decreasing in   
feasible directions, to within the default value of the optimality tolerance,  
and constraints are satisfied to within the default value of the constraint tolerance.  
  
Minimum found that satisfies the constraints.  
  
Optimization completed because the objective function is non-decreasing in   
feasible directions, to within the default value of the optimality tolerance,  
and constraints are satisfied to within the default value of the constraint tolerance.  
  
Rmins =  
  
 -0.0138  
  
Minimum found that satisfies the constraints.  
  
Optimization completed because the objective function is non-decreasing in   
feasible directions, to within the default value of the optimality tolerance,  
and constraints are satisfied to within the default value of the constraint tolerance.  
  
Rmin =  
  
 -0.0139  
  
Req =  
  
 0.0059  
  
Rwt =  
  
 0.0117

[*Published with MATLAB® R2017a*](http://www.mathworks.com/products/matlab)

**Part-2-**

We repeat our process we did in part 1 of getting the efficient frontier and the table of weights with additional assets added on.

**Input to the optimization program-**

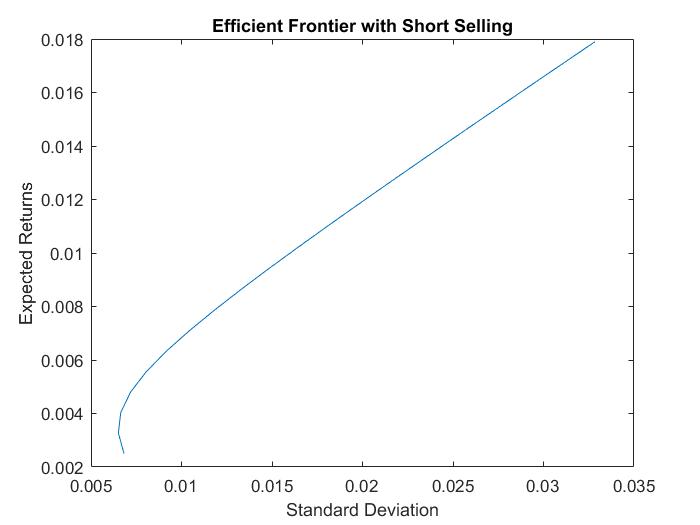
The input to our program is a table similar to part 1, with the addition of 5 columns representing the 5 additional assets we added to our optimization model.

Each column containing the adjusted closing price of the assets is imported into MATLAB using the import function, as a column vector.

**Outputs of the program-**

1. Efficient Frontier without short selling-
2. Table of optimal weights created without short selling-

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Expected Returns** | **Variance** | **W1** | **W2** | **W3** | **W4** | **W5** | **W6** | **W7** | **W8** |
| 1 | 0.0025 | 6.30703E-05 | 1.0322E-17 | 1 | 1.3881E-16 | 1.66765E-16 | 1.89634E-16 | 3.73748E-16 | 6.90601E-17 | 1.9947E-16 |
| 2 | 0.00327 | 4.66262E-05 | 0.075683571 | 0.891977147 | 0.017492025 | 1.53799E-05 | 5.37719E-08 | 0.000283463 | 0.014548218 | 1.41904E-07 |
| 3 | 0.00404 | 4.62477E-05 | 0.118550217 | 0.826219794 | 0.000110231 | 0.032240352 | 3.40953E-05 | 2.98891E-05 | 0.022012556 | 0.000802867 |
| 4 | 0.00481 | 5.63457E-05 | 0.104204647 | 0.772000693 | 1.53173E-06 | 0.056335836 | 0.019539 | 1.46425E-05 | 0.024304933 | 0.023598716 |
| 5 | 0.00558 | 7.3695E-05 | 0.081341068 | 0.724115629 | 3.17593E-07 | 0.07493286 | 0.050724693 | 1.70142E-06 | 0.026226637 | 0.042657093 |
| 6 | 0.00635 | 9.80925E-05 | 0.058422409 | 0.676231526 | 1.47245E-07 | 0.093462389 | 0.081923857 | 8.44269E-07 | 0.028257218 | 0.06170161 |
| 7 | 0.00712 | 0.000129537 | 0.035486978 | 0.628349438 | 1.93282E-06 | 0.111989159 | 0.113114055 | 1.162E-06 | 0.030285801 | 0.080771473 |
| 8 | 0.00789 | 0.000168029 | 0.013644628 | 0.580079261 | 1.5989E-06 | 0.13054583 | 0.144222586 | 4.12486E-07 | 0.032147671 | 0.099358012 |
| 9 | 0.00866 | 0.000213603 | 3.40082E-05 | 0.528843373 | 7.95883E-09 | 0.149306996 | 0.174531841 | 4.48805E-09 | 0.032648166 | 0.114635604 |
| 10 | 0.00943 | 0.000266531 | 4.65488E-05 | 0.47269882 | 7.48217E-08 | 0.1684098 | 0.203537347 | 1.94328E-08 | 0.030905383 | 0.124402006 |
| 11 | 0.0102 | 0.000326864 | 1.70256E-06 | 0.416575052 | 6.37573E-09 | 0.187511066 | 0.232548348 | 1.92734E-09 | 0.02917224 | 0.134191584 |
| 12 | 0.01097 | 0.000394605 | 2.53424E-07 | 0.360435516 | 1.94828E-09 | 0.206613164 | 0.261555188 | 9.19005E-10 | 0.027432445 | 0.143963431 |
| 13 | 0.01174 | 0.000469753 | 1.61971E-07 | 0.30429487 | 1.42088E-09 | 0.225713031 | 0.290561861 | 1.0153E-09 | 0.025696801 | 0.153733273 |
| 14 | 0.01251 | 0.000552308 | 2.34211E-07 | 0.24815072 | 1.32601E-08 | 0.244800309 | 0.319568303 | 1.02982E-08 | 0.023985445 | 0.163494965 |
| 15 | 0.01328 | 0.000642269 | 3.17804E-08 | 0.192013817 | 4.39069E-09 | 0.263913732 | 0.348575216 | 3.44118E-09 | 0.022223637 | 0.173273558 |
| 16 | 0.01405 | 0.000739637 | 1.17356E-07 | 0.135865544 | 2.73751E-08 | 0.282985929 | 0.377581404 | 2.16988E-08 | 0.020541396 | 0.183025561 |
| 17 | 0.01482 | 0.000844412 | 2.23218E-07 | 0.079714727 | 6.00633E-08 | 0.302048831 | 0.406587447 | 4.81187E-08 | 0.018877078 | 0.192771586 |
| 18 | 0.01559 | 0.000956594 | 4.05835E-07 | 0.023561246 | 1.01465E-07 | 0.321101827 | 0.43559347 | 8.15174E-08 | 0.017231683 | 0.202511184 |
| 19 | 0.01636 | 0.001092826 | 8.78784E-07 | 6.67687E-08 | 5.56998E-08 | 0.281223961 | 0.568781762 | 6.90508E-08 | 3.74054E-05 | 0.149955801 |
| 20 | 0.01713 | 0.001337627 | 7.60063E-08 | 3.30873E-08 | 3.01833E-08 | 0.175224324 | 0.790735142 | 4.57191E-08 | 2.45335E-07 | 0.034040103 |
| 21 | 0.0179 | 0.001713346 | 2.44324E-10 | 1.45761E-10 | 1.19236E-10 | 0.006029118 | 0.99397088 | 3.73263E-10 | 7.36034E-10 | 4.30498E-10 |

1. Efficient Frontier with short selling-
2. Table of optimal weights created with short selling-

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Expected Returns** | **Variance** | **W1** | **W2** | **W3** | **W4** | **W5** | **W6** | **W7** | **W8** |
| 1 | 0.0025 | 4.65438E-05 | 0.162943509 | 0.910310376 | 0.009432981 | 0.001427869 | -0.073082245 | 0.000957137 | 0.019348496 | -0.031338122 |
| 2 | 0.00327 | 4.24097E-05 | 0.178819152 | 0.885068366 | -0.025132121 | 0.02142931 | -0.047051833 | -0.01022951 | 0.01770002 | -0.020603385 |
| 3 | 0.00404 | 4.41447E-05 | 0.194694796 | 0.859826357 | -0.059697224 | 0.041430752 | -0.021021421 | -0.021416157 | 0.016051544 | -0.009868648 |
| 4 | 0.00481 | 5.17488E-05 | 0.21057044 | 0.834584347 | -0.094262326 | 0.061432194 | 0.005008991 | -0.032602804 | 0.014403069 | 0.000866089 |
| 5 | 0.00558 | 6.52219E-05 | 0.226446084 | 0.809342337 | -0.128827428 | 0.081433636 | 0.031039404 | -0.043789451 | 0.012754593 | 0.011600826 |
| 6 | 0.00635 | 8.45642E-05 | 0.242321728 | 0.784100327 | -0.16339253 | 0.101435078 | 0.057069816 | -0.054976098 | 0.011106117 | 0.022335563 |
| 7 | 0.00712 | 0.000109776 | 0.258197371 | 0.758858318 | -0.197957632 | 0.12143652 | 0.083100228 | -0.066162745 | 0.009457641 | 0.0330703 |
| 8 | 0.00789 | 0.000140856 | 0.274073015 | 0.733616308 | -0.232522734 | 0.141437961 | 0.10913064 | -0.077349393 | 0.007809165 | 0.043805038 |
| 9 | 0.00866 | 0.000177806 | 0.289948659 | 0.708374298 | -0.267087836 | 0.161439403 | 0.135161052 | -0.08853604 | 0.006160689 | 0.054539775 |
| 10 | 0.00943 | 0.000220624 | 0.305824303 | 0.683132289 | -0.301652939 | 0.181440845 | 0.161191464 | -0.099722687 | 0.004512213 | 0.065274512 |
| 11 | 0.0102 | 0.000269312 | 0.321699946 | 0.657890279 | -0.336218041 | 0.201442287 | 0.187221876 | -0.110909334 | 0.002863737 | 0.076009249 |
| 12 | 0.01097 | 0.000323869 | 0.33757559 | 0.632648269 | -0.370783143 | 0.221443729 | 0.213252288 | -0.122095981 | 0.001215261 | 0.086743986 |
| 13 | 0.01174 | 0.000384295 | 0.353451234 | 0.607406259 | -0.405348245 | 0.241445171 | 0.239282701 | -0.133282628 | -0.000433215 | 0.097478723 |
| 14 | 0.01251 | 0.00045059 | 0.369326878 | 0.58216425 | -0.439913347 | 0.261446612 | 0.265313113 | -0.144469275 | -0.00208169 | 0.10821346 |
| 15 | 0.01328 | 0.000522754 | 0.385202522 | 0.55692224 | -0.474478449 | 0.281448054 | 0.291343525 | -0.155655923 | -0.003730166 | 0.118948197 |
| 16 | 0.01405 | 0.000600787 | 0.401078165 | 0.53168023 | -0.509043551 | 0.301449496 | 0.317373937 | -0.16684257 | -0.005378642 | 0.129682935 |
| 17 | 0.01482 | 0.000684689 | 0.416953809 | 0.506438221 | -0.543608653 | 0.321450938 | 0.343404349 | -0.178029217 | -0.007027118 | 0.140417672 |
| 18 | 0.01559 | 0.000774461 | 0.432829453 | 0.481196211 | -0.578173756 | 0.34145238 | 0.369434761 | -0.189215864 | -0.008675594 | 0.151152409 |
| 19 | 0.01636 | 0.000870101 | 0.448705097 | 0.455954201 | -0.612738858 | 0.361453822 | 0.395465173 | -0.200402511 | -0.01032407 | 0.161887146 |
| 20 | 0.01713 | 0.000971611 | 0.46458074 | 0.430712191 | -0.64730396 | 0.381455264 | 0.421495585 | -0.211589158 | -0.011972546 | 0.172621883 |
| 21 | 0.0179 | 0.00107899 | 0.480456384 | 0.405470182 | -0.681869062 | 0.401456705 | 0.447525997 | -0.222775805 | -0.013621022 | 0.18335662 |

**MATLAB code and Outputs-**

%Calculating returns of all the stocks that were imported into MATLAB  
for i=2:85  
 returns(1,i-1)= (SPY(i,1)/SPY(i-1,1))-1;  
end  
for i=2:85  
 returns(2,i-1)= (GOVT(i,1)/GOVT(i-1,1))-1;  
end  
for i=2:85  
 returns(3,i-1)= (EEMV(i,1)/EEMV(i-1,1))-1;  
end  
for i=2:85  
 returns(4,i-1)= (CME(i,1)/CME(i-1,1))-1;  
end  
for i=2:85  
 returns(5,i-1)= (BR(i,1)/BR(i-1,1))-1;  
end  
for i=2:85  
 returns(6,i-1)= (CBOE(i,1)/CBOE(i-1,1))-1;  
end  
for i=2:85  
 returns(7,i-1)= (ICE(i,1)/ICE(i-1,1))-1;  
end  
for i=2:85  
 returns(8,i-1)= (ACN(i,1)/ACN(i-1,1))-1;  
end  
  
%Calculate arthimatic mean of returns  
for i=1:8  
 mean\_returns(i) =sum(returns(i,:))/84;  
end  
  
%Calculate expected returns  
for i=1:8  
 geo\_mean\_returns(i) =((prod(returns(i,:)+1))^(1/84))-1;  
end  
  
%Calculate covarience matrix  
for i=1:8  
 for j=1:8  
 COV(i,j)=sum((returns(i,:)-mean\_returns(i)).\*(returns(j,:)-mean\_returns(j)))/84;  
 end  
end  
  
  
%Setting bounds and restricting conditions for the optimization including  
%short selling  
A=-[0,0,0,0,0,0,0,0];  
B=-[0];  
RN=0;  
aeq=[1 1 1 1 1 1 1 1; geo\_mean\_returns];  
Beq=[1; RN];  
Ub=[inf,inf,inf,inf,inf,inf,inf,inf];  
Lb=[-inf, -inf, -inf,-inf, -inf, -inf,-inf, -inf];  
F=[0 0 0 0 0 0 0 0]';  
  
% Creating the optimized table with the Expected return, Standard  
% Deviation, and weights of all assets  
i=1;  
for RN =0.0025:0.00077:0.0179  
 Beq=[1; RN];  
 K(i,1)=RN;  
 [x,fval]=quadprog(COV,F,A,B,aeq,Beq, Lb, Ub);  
 K(i,2)=fval;  
 K(i,3)=x(1);  
 K(i,4)=x(2);  
 K(i,5)=x(3);  
 K(i,6)=x(4);  
 K(i,7)=x(5);  
 K(i,8)=x(6);  
 K(i,9)=x(7);  
 K(i,10)=x(8);  
 i=i+1;  
end  
K(:,2)=sqrt(K(:,2));  
  
%Plotting the Efficient Frontier  
figure;  
plot(K(:,2),K(:,1))  
title('Efficient Frontier with Short Selling');  
  
%Modifing bounds and restricting conditions for the optimization without  
%short selling  
Lb=[0, 0, 0, 0, 0, 0, 0, 0];  
  
%creating the optimized table again  
i=1;  
for RN =0.0025:0.00077:0.0179  
 Beq=[1; RN];  
 K(i,1)=RN;  
 [x,fval]=quadprog(COV,F,A,B,aeq,Beq, Lb, Ub);  
 K(i,2)=fval;  
 K(i,3)=x(1);  
 K(i,4)=x(2);  
 K(i,5)=x(3);  
 K(i,6)=x(4);  
 K(i,7)=x(5);  
 K(i,8)=x(6);  
 K(i,9)=x(7);  
 K(i,10)=x(8);  
 i=i+1;  
end  
K(:,2)=sqrt(K(:,2));  
%Plotting the Efficient Frontier  
figure;  
plot(K(:,2),K(:,1))  
title('Efficient Frontier without Short Selling');

Minimum found that satisfies the constraints.  
  
Optimization completed because the objective function is non-decreasing in   
feasible directions, to within the default value of the optimality tolerance,  
and constraints are satisfied to within the default value of the constraint tolerance.  
  
  
  
  
Minimum found that satisfies the constraints.  
  
Optimization completed because the objective function is non-decreasing in   
feasible directions, to within the default value of the optimality tolerance,  
and constraints are satisfied to within the default value of the constraint tolerance.

[*Published with MATLAB® R2017a*](http://www.mathworks.com/products/matlab)