APS502: Final Project – Computational Project

University of Toronto

Winter 2024

Instructor: Dr. Roy Kwon

Layth Al-Nemri (1010776126)

Contents

Initial Problem Description:

Building various portfolios and their efficient frontiers using the Mean-Variance Optimization Model based on the following Exchange-Traded Fund:

SPY, GOVT, EEMV, CME, BR, CBOE, ICE, CAN

Historical **Adjusted Closing Prices** were collected using: ca.finance.yahoo.com from Jan-2015 until Jan-2024.

The collected data was from **December 2014 until January 2024** to cover the requested period.

Parameters calculations were done using the following equations:

Table 1: Equations

$r_{it} = \dfrac{Last\ Day\ of\ Month_t}{Last\ Day\ of\ Month_{t-1}} - 1$	r_{it} = Return of asset i for month t
$\bar{r}_i = \dfrac{(\sum_{t=1}^{T} r_{it})}{T}$	$\bar{r}$ = Arithmetic average of asset i, T = Total number of months
$\mu_i = \left(\prod_{t=1}^{T}(1 + r_{it})\right)^{\frac{1}{T}} - 1$	μ_i = Expected return of asset i
$\sigma_{ij} = \dfrac{\left(\sum_{t=1}^{T}(r_{it} - \bar{r}_i)(r_{jt} - \bar{r}_j)\right)}{T}$	σ_{ij} = Covariance between assets i and j

The Efficient Frontier represents the minimum risk (Portfolio Variance) to achieve the return rate range based on the weights of the involved assets.

The portfolio variance = $\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{ij}$, where w_i is the weight of asset i from the investment capital, w_j is the weight of asset j from the investment capital, and σ_{ij} = Covariance between assets i and j

It can be written in the form of matrices as w^TQw, where Q is a symmetric matrix representing the covariances. We will be trying to minimize the portfolio variance subjected to $\mu^T w \geq r_{goal}$; $\Sigma w = 1$; $-\infty \leq w \leq \infty$

Quadprog is a matlab function which would solve quadtratic programs in the following form:

$\min \frac{1}{2} w^T Q w + c^T w$, Subjected to: $Aw \leq b$, $A_{eq}\, w = b_{eq}$, $lb \leq w \leq ub$

$c = 0$, $A = -\mu^T$, $b = r_{goal}$, $A_{eq} = 1$, $b_{eq} = 1$, lb & ub are specified with each section.

Part 1-a: SPY, GOVT, EEMV Portfolio with Short Selling

Using the equations in *Table 1* the following parameters were developed (to the nearest 5th Decimal Place):

Table 2: First 3 Assets Returns and STD

Asset	SPY	GOVT	EEMV
Average Return	0.01054	0.00064	0.00235
Expected Return	0.00951	**0.00054**	0.00168
Standard Deviation	0.06386	**0.02010**	0.05178

The SPY500 had the highest expected return and proportionally the highest risk

Table 3: First 3 Assets Covariance Matrix

Covariance σ_{ij}	SPY	GOVT	EEMV
SPY	0.002039	0.000065	0.001160
GOVT	0.000065	0.000202	0.000095
EEMV	0.001160	0.000095	0.001341

To build the portfolio's efficient frontier we solve the following quadratic program:

$$\textbf{Min } 0.5 * w^T Q w \qquad \textbf{Subjected to: } -\mu^T w \le r_{goal} \ ; \ \Sigma w = 1 \ ; \ -\infty \le w \le \infty$$

For multiple values of r_{goal}.

Main code used:

```
n=ETF_count; %number of assets
mu=ETF_returns; %expected returns of assets

Q=ETF_covariance; %covariance matrix
c=zeros(n,1); %linear coefficients
A = -mu; %unequal A matrix
Aeq =[ones(1,n)]; %equal A matrix
beq =[1]; %equal b vector
ub = [inf; inf; inf;]; %upper bound
lb = [-inf; -inf; -inf;]; %lower bound with short selling
```

Computing the **expected return goal bounds**: by selecting 10 equally spaced values between ($r_{max} - r_{min}$)

- The maximum return is the return of the asset with the maximum return; Short selling is allowed, **it can go to infinity**
- The minimum return was calculated using quadprog to check the **minimum variance portfolio**:

$$\textbf{Min } 0.5 * w^T Q w \qquad \textbf{Subjected to: } \Sigma w = 1 \ ; \ -\infty \le w \le \infty$$

```
%compute minimum variance portfolio:
%with shorting
[x_min(1,:),fval_min(1,1)] = quadprog (Q, c, [], [], Aeq, beq, lb, ub);
r_min = x_min(1,:)*mu';
```

- It is important to note: **x** is the weight of investment in each asset, and **fval** is portfolios variance/2
- To compute the minimum return we multiply x with the (expected returns of assets)^T

```
%return goals range from minimum variance portfolio to maximum return between assets
%without shorting:
goal_R = linspace(r_min, max(ETF_returns), 10);

%efficient frontier values with shorting
for a=1:length(goal_R)
    b = -goal_R(a);
    [x(a,:),fval(a,1)] = quadprog (Q, c, A, b, Aeq, beq, lb, ub);
    std_devi(a,1)=(fval(a,1)*2)^0.5; %calculating standard deviation from objective
function value
end
```

Table 4: First 3 Assets Mean-Variance Portfolios (Weights) - With Short Selling – Along with the Variance and Standard Deviation

	Expected Return Goal R	**SPY**	**GOVT**	**EEMV**	**Portfolio Variance (σ^2)**	**volatility (σ)**
1	**0.00095**	0.04214	0.91612	0.04175	**0.00019**	**0.01385**
2	0.00190	0.15799	0.89048	-0.04847	0.00021	0.01437
3	0.00285	0.27557	0.86446	-0.14003	0.00025	0.01583
4	0.00381	0.39317	0.83843	-0.23161	0.00032	0.01799
5	0.00476	0.51077	0.81241	-0.32318	0.00043	0.02065
6	0.00571	0.62839	0.78638	-0.41477	0.00056	0.02363
7	0.00666	0.74598	0.76035	-0.50634	0.00072	0.02682
8	0.00761	0.86359	0.73433	-0.59792	0.00091	0.03017
9	0.00856	0.98119	0.70830	-0.68949	0.00113	0.03361
10	**0.00951**	1.09879	0.68228	-0.78107	**0.00138**	**0.03713**

The portfolio's lowest return is higher than the lowest return of any of the individual ETFs (0.095% vs 0.054%) with a lower volatility/std (0.01385 vs 0.02010). The highest return is the same but also with lower volatility.

EEMV returns are low with a standard deviation almost as high as SPY, it makes sense to be mostly short sold.

Part 1-b: SPY, GOVT, EEMV Portfolio with No Short Selling

Min $0.5 * w^T Q w$ Subjected to: $\mu^T w \leq -r_{goal}$; $\Sigma w = 1$; $0 \leq w \leq \infty$

Main code used:

```
lb_without = [0; 0; 0;]; %lower bound without short selling

%minimum variance portfolio without shorting
[x_min_without(1,:),fval_min_without(1,1)] = quadprog (Q, c, [], [], Aeq, beq,
lb_without, ub);
r_min_without = x_min_without(1,:)*mu';

% return goals without shorting
goal_R_without_3 = linspace(r_min_without, max(ETF_returns), 10);
```

- Same result as with short selling

```
%efficient frontier values without shorting
for a=1:length(goal_R_without_3)
    b = -goal_R_without_3(a);
    [x_without(a,:),fval_without(a,1)] = quadprog (Q, c, A, b, Aeq, beq, lb_without, ub);
    std_devi_without_3(a,1)=(fval_without(a,1)*2)^0.5; %calculating standard deviation
from objective function value
end
```

Table 5: First 3 Assets Mean-Variance Portfolios (Weights) - Without Short Selling – Along with the Variance and Standard Deviation

	Expected Return Goal R	**SPY**	**GOVT**	**EEMV**	**Portfolio Variance (σ^2)**	**volatility (σ)**
1	**0.00095**	0.04223	0.91610	0.04167	**0.00019**	**0.01385**
2	0.00190	0.15184	0.84816	1.92E-07	0.00021	0.01446
3	0.00285	0.25786	0.74214	3.47E-07	0.00027	0.01648
4	0.00381	0.36388	0.63612	2.33E-06	0.00038	0.01954
5	0.00476	0.46990	0.53010	3.94E-07	0.00054	0.02322
6	0.00571	0.57592	0.42408	3.22E-07	0.00074	0.02728
7	0.00666	0.68194	0.31806	2.93E-08	0.00100	0.03157
8	0.00761	0.78796	0.21204	2.67E-06	0.00130	0.03601
9	0.00856	0.89398	0.10602	1.19E-07	0.00164	0.04055
10	**0.00951**	1.00	3.09E-11	3.09E-11	**0.00204**	**0.04516**

The lowest return has the same volatility with short selling allowed and not allowed, but right after the volatility with no short selling increases for the same return goal (0.04516 vs 0.03713 for the maximum return goal) EEMV returns are low with a high standard deviation, its weights are almost zero. (*Tables 4 & 5*)

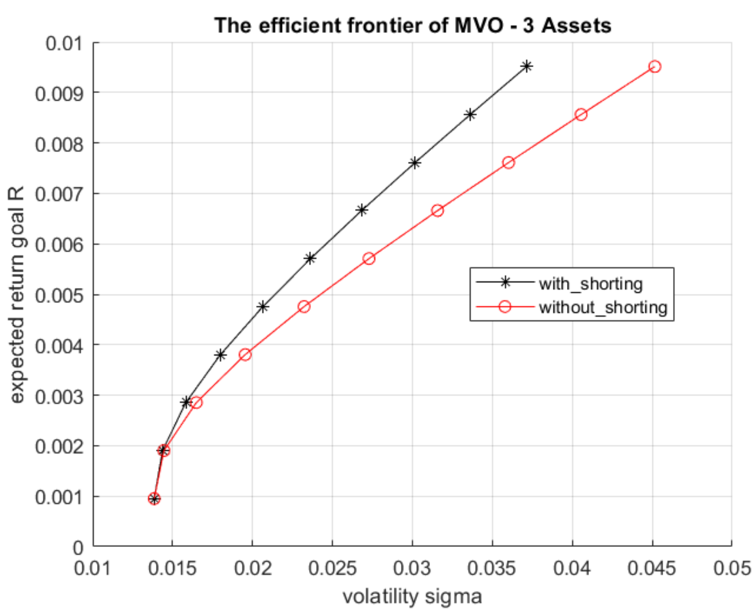

Figure 1: Efficient Frontiers for the First 3 Assets - With and Without Short Selling

From *Figure 1*: we can see how both portfolios start from the same point, but in the end short selling allowed a lower volatility for the same return.
We can also see a bigger area under the curve with short selling vs without short selling.

Part 2-a: SPY, GOVT, EEMV, CME, BR, CBOE, ICE, CAN Portfolio with No Short Sell

Min $0.5*w^TQw$ Subjected to: $\mu^T w \leq -r_{goal}$; $\Sigma w = 1$; $0 \leq w \leq \infty$

Using the equations in *Table 1* the following parameters were developed (to the nearest 5th Decimal Place):

Table 6: All 8 Assets Returns and STD

Asset	SPY	GOVT	EEMV	CME	BR	CBOE	ICE	ACN
Average Return	0.01054	0.00064	0.00235	0.01253	0.01753	0.01340	0.01278	0.01668
Expected Return	0.00951	**0.00054**	0.00168	0.01108	**0.01545**	0.01142	0.01102	0.01458
Standard Deviation	0.06386	**0.02010**	0.05178	0.07662	**0.09188**	0.08808	0.08445	0.09173

Broadridge Financial Solutions had the highest expected return and proportionally the highest risk.

Table 7: All 8 Assets Covariance Matrix

Covariance σ_{ij}	SPY	GOVT	EEMV	CME	BR	CBOE	ICE	ACN
SPY	0.00204	0.00006	0.00116	0.00089	0.00194	0.00100	0.00179	0.00244
GOVT	0.00006	0.00020	0.00010	-0.00002	0.00021	0.00004	0.00014	0.00015
EEMV	0.00116	0.00010	0.00134	0.00032	0.00098	0.00034	0.00077	0.00115
CME	0.00089	-0.00002	0.00032	0.00294	0.00116	0.00169	0.00190	0.00110
BR	0.00194	0.00021	0.00098	0.00116	0.00422	0.00114	0.00197	0.00285
CBOE	0.00100	0.00004	0.00034	0.00169	0.00114	0.00388	0.00163	0.00136
ICE	0.00179	0.00014	0.00077	0.00190	0.00197	0.00163	0.00357	0.00235
ACN	0.00244	0.00015	0.00115	0.00110	0.00285	0.00136	0.00235	0.00421

Main code used:

```
n=ETF_count; %number of assets
mu=ETF_returns; % expected returns of assets

Q=ETF_covariance; % covariance matrix
c=zeros(n,1); %linear coefficients
A = -mu; %unequal A matrix
Aeq =[ones(1,n)]; %equal A matrix
beq =[1]; %equal b vector
ub = [inf; inf; inf; inf; inf; inf; inf; inf;]; %upper bound
lb_without = [0; 0; 0; 0; 0; 0; 0; 0;]; %lower bound without short selling

%compute minimum variance portfolio:
%without shorting
[x_min_without(1,:),fval_min_without(1,1)] = quadprog (Q, c, [], [], Aeq, beq,
lb_without, ub);
r_min_without = x_min_without(1,:)*mu';

%return goals without shorting
goal_R_without = linspace(r_min_without, max(ETF_returns), 10);
```

- Same method with different results since the data is different

```
%efficient frontier values without shorting
for a=1:length(goal_R_without)
    b = -goal_R_without(a);
    [x_without(a,:),fval_without(a,1)] = quadprog (Q, c, A, b, Aeq, beq, lb_without, ub);
    std_devi_without(a,1)=(fval_without(a,1)*2)^0.5; %calculating standard deviation from
objective function value
end
```

Table 8: All 8 Assets Mean-Variance Portfolios (Weights) - Without Short Selling – Along with the Variance and Standard Deviation

	Expected Return Goal R	SPY	GOVT	EEMV	CME	BR	CBOE	ICE	ACN	Portfolio Variance (σ^2)	volatility (σ)
1	**0.00131**	0.00256	0.87572	0.05530	0.05833	8.1E-07	0.00808	1.2E-06	2.1E-06	0.00018	**0.01347**
2	0.00288	0.09732	0.76684	1.7E-07	0.09716	0.00569	0.03296	6.9E-08	0.00003	0.00022	0.01477
3	**0.00445**	0.08537	0.64945	4.6E-08	0.12862	0.05728	0.05587	4.8E-07	0.02340	0.00033	**0.01809**
4	0.00602	0.06042	0.53668	5.7E-09	0.16030	0.10465	0.07768	9.0E-08	0.06027	0.00050	0.02238
5	0.00759	0.03544	0.42392	1.9E-08	0.19198	0.15199	0.09949	6.9E-08	0.09719	0.00074	0.02717
6	0.00917	0.01134	0.31086	2.5E-07	0.22359	0.19926	0.12127	4.2E-07	0.13368	0.00104	0.03224
7	**0.01074**	1.1E-05	0.19345	2.6E-10	0.25421	0.24546	0.14271	5.1E-10	0.16416	0.00141	**0.03749**
8	0.01231	5.0E-05	0.07216	1.3E-08	0.28397	0.29071	0.16384	4.9E-08	0.18927	0.00184	0.04285
9	0.01388	1.1E-08	2.6E-09	1.9E-09	0.18922	0.44194	0.13395	8.6E-09	0.23490	0.00242	0.04920
10	**0.01545**	1.9E-12	1.3E-13	1.1E-13	9.4E-13	1.0E+00	1.2E-12	8.8E-13	1.0E-10	0.00422	**0.06497**

With more ETFs included, this portfolio's lowest return is higher than with only 3 assets (0.131% vs 0.095%) And with a lower volatility/std (0.01347 vs 0.01385). (*Tables 5 & 8*)
The highest return goal increased and so did its volatility; but for a fair comparison we check a close return:

Return_goal_7 from *Table 8*	vs	Return_goal_10 from *Tables 4&5*:
1.074% ; 0.03749 std	vs	0.951% ; 0.04516 & 0.03713 std

It shows that having more assets in the portfolio (with higher individual returns) gives better results.

Part 2-b: SPY, GOVT, EEMV, CME, BR, CBOE, ICE, CAN Portfolio with 5% Min Weight

$$\textbf{Min } 0.5^*w^TQw \qquad \textbf{Subjected to: } \mu^Tw \le -r_{goal} \;\; ; \;\; \Sigma w = 1 \;\; ; \;\; 0.5 \le w \le \infty$$

Main code used:

```
lb_5 = [0.05; 0.05; 0.05; 0.05; 0.05; 0.05; 0.05; 0.05;]; %lower bound 5%

%minimum variance portfolio with at least 5% weight
[x_min_5(1,:),fval_min_5(1,1)] = quadprog (Q, c, [], [], Aeq, beq, lb_5, ub);
r_min_5 = x_min_5(1,:)*mu';
```

- Same method with different results since the bounds are different

```
%compute maximum return portfolio for "at least 5%"
f = -ETF_returns;
[x_max_5(1,:),fval_max_5(1,1)] = linprog(f, [], [], Aeq, beq, lb_5, ub);
```

- With a minimum weight of 5%, the **maximum return** is now bounded and can be calculated using linprog to solve the following linear program:

$$\textbf{Min } -\mu^Tw \qquad \textbf{Subjected to } \Sigma w = 1 \;\; ; \;\; 0.05 \le w \le \infty$$

Which gives the same result as giving a weight of 5% to the lowest 7 asset and 35% to the highest one.

```
%return goal with at least 5%
goal_R_5 = linspace(r_min_5, -fval_max_5(1,1), 10);

%efficient frontier values at least 5%
for a=1:length(goal_R_5)
    b = -goal_R_5(a);
    [x_5(a,:),fval_5(a,1)] = quadprog (Q, c, A, b, Aeq, beq, lb_5, ub);
    std_devi_5(a,1)=(fval_5(a,1)*2)^0.5; %calculating standard deviation from OF value
end
```

Table 9: All 8 Assets Mean-Variance Portfolios (Weights) – With 5% Min Weight – Along with the Variance and Standard Deviation

	Expected Return Goal R	SPY	GOVT	EEMV	CME	BR	CBOE	ICE	ACN	Portfolio Variance (σ^2)	volatility (σ)
1	**0.00409**	0.05000	0.65000	0.05000	0.05000	0.05000	0.05000	0.05000	0.05000	0.00033	**0.01828**
2	0.00508	0.05000	0.55980	0.05000	0.12132	0.05919	0.05969	0.05000	0.05000	0.00042	0.02053
3	0.00608	0.05000	0.48267	0.05000	0.14181	0.09931	0.07617	0.05000	0.05003	0.00054	0.02332
4	0.00707	0.05000	0.40569	0.05000	0.16167	0.13503	0.09146	0.05000	0.05614	0.00069	0.02632
5	0.00806	0.05000	0.32896	0.05000	0.18051	0.16374	0.10486	0.05000	0.07193	0.00087	0.02946
6	0.00906	0.05012	0.25219	0.05000	0.19932	0.19236	0.11823	0.05001	0.08777	0.00107	0.03270
7	0.01005	0.05004	0.17550	0.05000	0.21816	0.22100	0.13160	0.05000	0.10371	0.00130	0.03601
8	0.01105	0.05006	0.09876	0.05000	0.23698	0.24962	0.14497	0.05000	0.11960	0.00155	0.03936
9	0.01204	0.05000	0.05000	0.05000	0.18496	0.33856	0.12930	0.05000	0.14718	0.00186	0.04310
10	**0.01303**	0.05000	0.05000	0.05000	0.05000	0.65000	0.05000	0.05000	0.05000	0.00260	**0.05096**

Return_goal_1 from *Table 9* vs Return_goal_3 from *Table 8*

0.409% ; 0.01828 std vs 0.445% ; 0.01809 std

Introducing a minimum weight gave all ETFs an effect and a contribution, the weights are more balanced which would constrain the optimization. The return range got smaller, and the volatility is relatively higher. But it still performed better than the portfolio with 3 ETFs and no short selling.

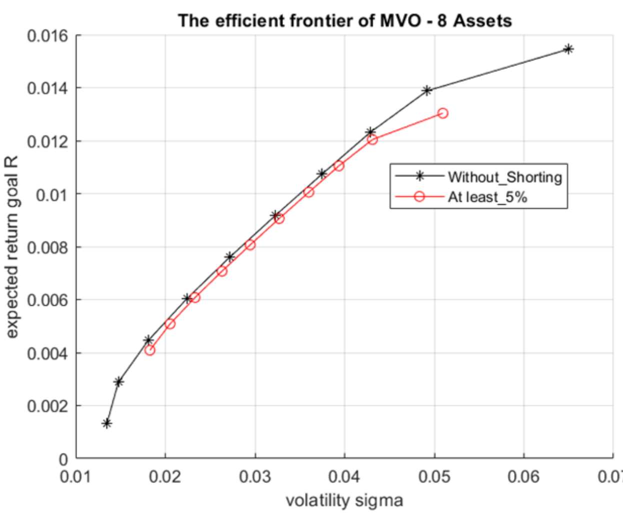

Figure 2: Efficient Frontiers for All 8 Assets - Without Short Selling and With At Least 5% Weights

From *Figure 2*: we can see how both the beginning and ending of each portfolio is different, having a minimum weight limited the range and decreased the area under the curve.
It did increase riskiness but not by a considerable amount.

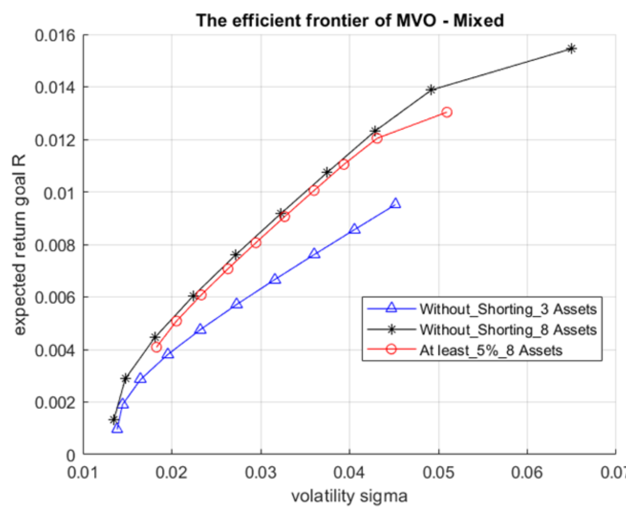

Figure 3: Efficient Frontiers for the First 3 Assets Without Short Selling - All 8 Assets Without Short Selling - and All 8 Assets With At Least 5% Weights

From *Figure 3*: Arranged starting from the best portfolio:

1. 8 ETFs with no short selling.
2. 8 ETFs with 5% weight limit
3. 3 ETFs with no short selling

The area under the curve increases, the frontiers size increases and shifts to the lift (lower risk) the better the portfolio.

Appendix

Bonus Graph: Allowing short selling with 8 assets

From *Figure 4*: It is clear how short selling would decrease the risk and shift the frontier to the left.

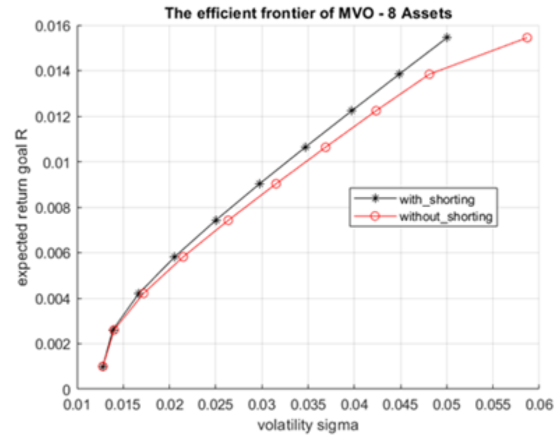

Figure 4: Efficient Frontiers for All 8 Assets – With and Without Short selling

Part 1 Code:

```
ETF = {'SPY.csv', 'GOVT.csv', 'EEMV.csv'}; %define the series of CSV file names
ETF_count = numel(ETF);

ETF_value = cell(1, ETF_count); %define a cell array to store the data from each CSV file
for i = 1:ETF_count                          %loop through each CSV file
    ETF_value{i} = readtable(ETF{i});             %read the CSV file
    ETF_value{i}.Date = datetime(ETF_value{i}.Date); %convert the first column to datetime format
    ETF_value{i}.Month = month(ETF_value{i}.Date);      %extract month and year information
end
%define a cell array to store return of each end of month
end_of_month = cell(1, ETF_count);
%getting total number of months
monthYear = year(ETF_value{1}.Date) * 100 + month(ETF_value{1}.Date);
total_months_count = size(histcounts(monthYear, unique(monthYear)),2);
zeros_1 = zeros(total_months_count,1);

for i = 1:ETF_count         %getting the end of month values
    past_month = 12;
    p = 1;
    end_of_month{i} = table(zeros_1,zeros_1,'VariableNames',{'End_of_Month', 'Return'});
    for j = 1:height(ETF_value{i})
        current_month = ETF_value{i}.Month(j);
        if current_month ~= past_month
            end_of_month{i}.End_of_Month(p) = past_month;
            end_of_month{i}.Return(p) = ETF_value{i}.AdjClose(j-1);
            p = p+1;
        end
        past_month = ETF_value{i}.Month(j);
    end
end
months_returns = cell(1, ETF_count);       %define a cell array to store months returns
Average_returns = zeros(1, ETF_count);     %define a matrix to store arithmatic average
ETF_returns = zeros(1, ETF_count);         %define a matrix to store ETF returns
months_count = total_months_count - 1;     %getting included months number (T)
zeros_2 = zeros(months_count,1);
for i = 1:ETF_count
    months_returns{i} = table(zeros_2, zeros_2, zeros_2, 'VariableNames', {'Month', 'Return',
'Return_minus_Average'});
    for j = 2:height(end_of_month{i})
        months_returns{i}.Month(j-1) = end_of_month{i}.End_of_Month(j);
        %calculating monthly_returns(i) = (last_day_of_month(i)/ last_day_of_month(i-1)-1
        months_returns{i}.Return(j-1) = ((end_of_month{i}.Return(j)/end_of_month{i}.Return(j-1))-1);
    end
```

```
    %calcualting arithmatic average = avg(r_it)
    Average_returns(i) = mean(months_returns{i}.Return);
    %calcualting geometric expected return of asset i = ((product(1+r_it))^(1/T))-1
    ETF_returns(i) = ((prod((months_returns{i}.Return)+1))^(1/months_count))-1;

    for k = 1:months_count %calculating r_it - avg(r_it)
        months_returns{i}.Return_minus_Average(k)=months_returns{i}.Return(k)-Average_returns(i);
    end
end

ETF_covariance = zeros(ETF_count, ETF_count);     %define a matrix to store ETF covariance
for i = 1:ETF_count
    for j = 1:ETF_count
        Difference_product = 1;
        Difference_product_sum = 0;
        if ETF_covariance(i,j) == 0 %to avoid calculating the covariance twice
            for k = 1:months_count
                %[r_it - avg(r_it)] * [r_jt - avg(r_jt)]
                Difference_product = months_returns{i}.Return_minus_Average(k) *
months_returns{j}.Return_minus_Average(k);
                %sum{[r_it - avg(r_it)] * [r_jt - avg(r_jt)]}
                Difference_product_sum = Difference_product_sum + Difference_product;
            end
            %covariance between assets i and j=sum{[r_it-avg(r_it)]*[r_jt-avg(r_jt)]}/T
            ETF_covariance(i,j) = Difference_product_sum/months_count;
            ETF_covariance(j,i) = ETF_covariance(i,j);
        end
    end
end

n=ETF_count; % number of assets
mu=ETF_returns; % expected returns of assets
Q=ETF_covariance; % covariance matrix
c=zeros(n,1); %linear coefficients
A = -mu; %unequal A matrix
Aeq =[ones(1,n)]; %equal A matrix
beq =[1]; %equal b vector
ub = [inf; inf; inf;]; %upper bound
lb = [-inf; -inf; -inf;]; %lower bound with short selling
lb_without = [0; 0; 0;]; %lower bound without short selling
%compute minimum variance portfolio:
[x_min(1,:),fval_min(1,1)] = quadprog (Q, c, [], [], Aeq, beq, lb, ub);
r_min = x_min(1,:)*mu';                         %with shorting
[x_min_without(1,:),fval_min_without(1,1)] = quadprog (Q, c, [], [], Aeq, beq, lb_without, ub);
r_min_without = x_min_without(1,:)*mu';    %without shorting

%expected return goals from minimum variance portfolio to maximum return between assets
goal_R = linspace(r_min, max(ETF_returns), 10);                 %with shorting
goal_R_without_3 = linspace(r_min_without, max(ETF_returns), 10);      % without shorting

for a=1:length(goal_R)              %efficient frontier values with shorting
    b = -goal_R(a);
    [x(a,:),fval(a,1)] = quadprog (Q, c, A, b, Aeq, beq, lb, ub);
    std_devi(a,1)=(fval(a,1)*2)^0.5; %calculating standard deviation from objective function value
end

for a=1:length(goal_R_without_3)    %efficient frontier values without shorting
    b = -goal_R_without_3(a);
    [x_without(a,:),fval_without(a,1)] = quadprog (Q, c, A, b, Aeq, beq, lb_without, ub);
    std_devi_without_3(a,1)=(fval_without(a,1)*2)^0.5; %calculating standard deviation from objective
function value
end
hold on
plot(std_devi, goal_R, '-k*');
plot(std_devi_without_3, goal_R_without_3, '-ro');
xlabel('volatility sigma')
ylabel('expected return goal R')
title('The efficient frontier of MVO - 3 Assets')
grid on;
legend('with\_shorting', 'without\_shorting', 'Location', 'best');
hold off
```

Part 2 Code:

```
%clear
clearvars -except std_devi_without_3 goal_R_without_3
clc

%define the series of CSV file names
ETF = {'SPY.csv', 'GOVT.csv', 'EEMV.csv', 'CME.csv', 'BR.csv', 'CBOE.csv', 'ICE.csv', 'ACN.csv'};
ETF_count = numel(ETF);
%define a cell array to store the data from each CSV file
ETF_value = cell(1, ETF_count);
for i = 1:ETF_count  %loop through each CSV file
    ETF_value{i} = readtable(ETF{i});            %read the CSV file
    ETF_value{i}.Date = datetime(ETF_value{i}.Date);           %convert the first column to datetime
format
    ETF_value{i}.Month = month(ETF_value{i}.Date);       %extract month and year information
end

end_of_month = cell(1, ETF_count); %define a cell array to store return end of month
monthYear=year(ETF_value{1}.Date)*100+ month(ETF_value{1}.Date); %getting total no.months
total_months_count = size(histcounts(monthYear, unique(monthYear)),2);
zeros_1 = zeros(total_months_count,1);
for i = 1:ETF_count  %getting the end of month values
    past_month = 12;
    p = 1;
    end_of_month{i} = table(zeros_1, zeros_1, 'VariableNames', {'End_of_Month', 'Return'});
    for j = 1:height(ETF_value{i})
        current_month = ETF_value{i}.Month(j);
        if current_month ~= past_month
            end_of_month{i}.End_of_Month(p) = past_month;
            end_of_month{i}.Return(p) = ETF_value{i}.AdjClose(j-1);
            p = p+1;
        end
        past_month = ETF_value{i}.Month(j);
    end
end

months_returns = cell(1, ETF_count);             %define a cell array to store months returns
Average_returns = zeros(1, ETF_count);     %define a matrix to store arithmatic average
ETF_returns = zeros(1, ETF_count);         %define a matrix to store ETF returns
months_count = total_months_count - 1;     %getting included months number (T)
zeros_2 = zeros(months_count,1);
for i = 1:ETF_count
    months_returns{i} = table(zeros_2, zeros_2, zeros_2, 'VariableNames', {'Month', 'Return',
'Return_minus_Average'});
    for j = 2:height(end_of_month{i})
        months_returns{i}.Month(j-1) = end_of_month{i}.End_of_Month(j);
        %calculating monthly_returns(i) = (last_day_of_month(i)/ last_day_of_month(i-1))-1
        months_returns{i}.Return(j-1) = ((end_of_month{i}.Return(j)/end_of_month{i}.Return(j-1))-1);
    end
    %calcualting arithmatic average = avg(r_it)
    Average_returns(i) = mean(months_returns{i}.Return);
    %calcualting geometric expected return of asset i = ((product(1+r_it))^(1/T))-1
    ETF_returns(i) = ((prod((months_returns{i}.Return)+1))^(1/months_count))-1;

    for k = 1:months_count  %calculating r_it - avg(r_it)
        months_returns{i}.Return_minus_Average(k) = months_returns{i}.Return(k)-Average_returns(i);
    end
end

%define a matrix to store ETF covariance
ETF_covariance = zeros(ETF_count, ETF_count);
for i = 1:ETF_count
    for j = 1:ETF_count
        Difference_product = 1;
        Difference_product_sum = 0;
        if ETF_covariance(i,j) == 0 %to avoid calculating the covariance twice
            for k = 1:months_count
                %[r_it - avg(r_it)] * [r_jt - avg(r_jt)]
                Difference_product = months_returns{i}.Return_minus_Average(k) *
months_returns{j}.Return_minus_Average(k);
```

```
                %sum{[r_it - avg(r_it)] * [r_jt - avg(r_jt)]}
                Difference_product_sum = Difference_product_sum + Difference_product;
            end
            %covariance between assets i and j = sum{[r_it - avg(r_it)] * [r_jt - avg(r_jt)]}/T
            ETF_covariance(i,j) = Difference_product_sum/months_count;
            ETF_covariance(j,i) = ETF_covariance(i,j);
        end
    end
end

n=ETF_count; % number of assets
mu=ETF_returns; % expected returns of assets
Q=ETF_covariance; % covariance matrix
c=zeros(n,1); %linear coefficients
A = -mu; %unequal A matrix
Aeq =[ones(1,n)]; %equal A matrix
beq =[1]; %equal b vector
ub = [inf; inf; inf; inf; inf; inf; inf; inf;]; %upper bound
lb_without = [0; 0; 0; 0; 0; 0; 0; 0;]; %lower bound without short selling
lb_5 = [0.05; 0.05; 0.05; 0.05; 0.05; 0.05; 0.05; 0.05;]; %lower bound 5%

%compute minimum variance portfolio:
[x_min_without(1,:),fval_min_without(1,1)] = quadprog (Q, c, [], [], Aeq, beq, lb_without, ub);
r_min_without = x_min_without(1,:)*mu';    %without shorting
[x_min_5(1,:),fval_min_5(1,1)] = quadprog (Q, c, [], [], Aeq, beq, lb_5, ub);
r_min_5 = x_min_5(1,:)*mu'; %at least 5%
%compute maximum return portfolio for "at least 5%"
f = -ETF_returns;
[x_max_5(1,:),fval_max_5(1,1)] = linprog(f, [], [], Aeq, beq, lb_5, ub);

%expected return goals range from minimum variance portfolio to maximum return between assets
goal_R_without = linspace(r_min_without, max(ETF_returns), 10); %without shorting
goal_R_5 = linspace(r_min_5, -fval_max_5(1,1), 10);      %at least 5%

for a=1:length(goal_R_without)      %efficient frontier values without shorting
    b = -goal_R_without(a);
    [x_without(a,:),fval_without(a,1)] = quadprog (Q, c, A, b, Aeq, beq, lb_without, ub);
    std_devi_without(a,1)=(fval_without(a,1)*2)^0.5; %calculating standard deviation from objective
function value
end

for a=1:length(goal_R_5)     %efficient frontier values at least 5%
    b = -goal_R_5(a);
    [x_5(a,:),fval_5(a,1)] = quadprog (Q, c, A, b, Aeq, beq, lb_5, ub);
    std_devi_5(a,1)=(fval_5(a,1)*2)^0.5; %calculating standard deviation from objective function value
end

hold on
plot(std_devi_without, goal_R_without, '-k*');
plot(std_devi_5, goal_R_5, '-ro');
xlabel('volatility sigma')
ylabel('expected return goal R')
title('The efficient frontier of MVO - 8 Assets')
grid on;
legend('Without\_Shorting', 'At least\_5%', 'Location', 'best');
hold off
hold on
plot(std_devi_without_3, goal_R_without_3, '-b^');
plot(std_devi_without, goal_R_without, '-k*');
plot(std_devi_5, goal_R_5, '-ro');
xlabel('volatility sigma')
ylabel('expected return goal R')
title('The efficient frontier of MVO - Mixed')
grid on;
legend('Without\_Shorting\_3 Assets','Without\_Shorting\_8 Assets', 'At least\_5%\_8 Assets', 'Location',
'best');
hold off
```