Coursework 3 Answer

Junbiao Li - 209050796

December 6, 2023

Contents

1 Question a)

Since we need to find that the minimum variance portfolio of the 10 stocks. We first calculate the weekly returns of the 10 stocks and expected return and variance of return, which can be calculate as

$$\begin{aligned}
\mathbb{E}\left[R_i\right] &= \frac{1}{60}\sum_{t=1}^{60} R_i(t), \quad i=1,\cdots,10 \\
\sigma_i^2 &= \frac{1}{60}\sum_{t=1}^{60} \left(R_i(t)-\mathbb{E}\left[R_i\right]\right)^2, \quad i=1,\cdots,10 \\
\sigma_{ij} &= \frac{1}{60}\sum_{t=1}^{60} \left(R_i(t)-\mathbb{E}\left[R_i\right]\right)\left(R_j(t)-\mathbb{E}\left[R_j\right]\right), \quad i\neq j
\end{aligned}$$

where $R_i(t) = \frac{P_i(t)-P_i(t-1)}{P_i(t-1)}$.

Hence, using MATLAB, we have the following results:

$$\text{expected return} = \begin{pmatrix} 0.0032 \\ 0.0035 \\ -0.0005 \\ 0.0010 \\ -0.0016 \\ 0.0180 \\ -0.0017 \\ 0.0049 \\ -0.0045 \\ 0.0014 \end{pmatrix},$$

covariances =

$$\begin{pmatrix}
0.0018 & 0.0002 & 0.0009 & 0.0008 & 0.0006 & 0.0009 & 0.0009 & 0.0004 & 0.0009 & 0.0005 \\
0.0002 & 0.0008 & 0.0006 & 0.0003 & 0.0003 & 0.0002 & 0.0004 & -0.0001 & 0.0004 & 0.0003 \\
0.0009 & 0.0006 & 0.0023 & 0.0012 & 0.0014 & 0.0012 & 0.0013 & 0.0001 & 0.0011 & 0.0007 \\
0.0008 & 0.0003 & 0.0012 & 0.0020 & 0.0009 & 0.0010 & 0.0011 & 0.0005 & 0.0010 & 0.0008 \\
0.0006 & 0.0003 & 0.0014 & 0.0009 & 0.0016 & 0.0007 & 0.0011 & -0.0001 & 0.0009 & 0.0006 \\
0.0009 & 0.0002 & 0.0012 & 0.0010 & 0.0007 & 0.0032 & 0.0008 & 0.0003 & 0.0001 & 0.0007 \\
0.0009 & 0.0004 & 0.0013 & 0.0011 & 0.0011 & 0.0008 & 0.0016 & 0.0002 & 0.0013 & 0.0007 \\
0.0004 & -0.0001 & 0.0001 & 0.0005 & -0.0001 & 0.0003 & 0.0002 & 0.0010 & 0.0003 & 0.0003 \\
0.0009 & 0.0004 & 0.0011 & 0.0010 & 0.0009 & 0.0001 & 0.0013 & 0.0003 & 0.0022 & 0.0006 \\
0.0005 & 0.0003 & 0.0007 & 0.0008 & 0.0006 & 0.0007 & 0.0007 & 0.0003 & 0.0006 & 0.0009
\end{pmatrix}$$

Now we can solve the following optimization problem using MATLAB with the solver for quadratic objective functions with linear constraints. The code is shown in Appendix 1.

$$\min_{x_1,\cdots,x_n} \quad \sigma_p^2$$
$$\text{Subject to} \quad \sum_{i=1}^{n} x_i = 1$$

Setting `H=2*covariances;f=zeros(n_stocks,1);Aeq=ones(1,n_stocks);beq=1;` The result is

- Weights of the portfolio:

AHT.L:	0.031339
CCH.L:	0.446839
FRAS.L:	−0.158178
MNG.L:	−0.069499
RMV.L:	0.297457
RR.L:	0.040424
SDR.L:	−0.034508
SHEL.L:	0.390948
STJ.L:	−0.008146
TSCO.L:	0.063325

- Variance of the portfolio: 0.000314
- Expected return of the portfolio: 0.004029

2 Question b)

In this question, we need to find the Minimum variance portfolio with a specific expected return, which is

$$\min_{x_1,\cdots,x_n} \sigma_p^2$$
$$\text{Subject to } \mathbb{E}_p = \mathbb{E}_0$$
$$\sum_{i=1}^{n} x_i = 1$$

Like the previous question, we can solve the problem using MATLAB, however, we need to add the constraint $\mathbb{E}_p = \mathbb{E}_0$. The code is shown in Appendix 2.

Setting `H=2*covariances;f=zeros(n_stocks,1);target_return=0.015;` `Aeq=[ones(1,n_stocks);expected_returns];beq=[1;target_return];` The result is

- Weights of the portfolio:

AHT.L:	0.060745
CCH.L:	0.686850
FRAS.L:	−0.364956
MNG.L:	−0.129785
RMV.L:	0.214895
RR.L:	0.570912
SDR.L:	−0.231232
SHEL.L:	0.455774
STJ.L:	−0.026677
TSCO.L:	−0.236526

- Variance of the portfolio: 0.001171
- Expected return of the portfolio: 0.015000

3 Question c)

In this question, we need to Maximizing the utility, which is $u = \mathbb{E}_p - 0.5\sigma_p^2$. Hence we need to setting `kappa=0.5;H=2*kappa*covariances;f=-expected_returns;` `Aeq=ones(1, n_stocks);beq=1;` The code is shown in Appendix 3.

The result is

- Weights of the portfolio:

AHT.L:	0.407439
CCH.L:	3.516466
FRAS.L:	−2.802765
MNG.L:	−0.840536
RMV.L:	−0.758477
RR.L:	6.825118
SDR.L:	−2.550508
SHEL.L:	1.220050
STJ.L:	−0.245152
TSCO.L:	−3.771635

- Maximum utility portfolio: 0.074028
- Expected return of the portfolio: 0.144340

4 Question d)

In this question, we are required to find the maximum return portfolio with a certain level of risk such that $\sigma_p^2 = 0.004$ which is

$$\begin{gathered}\max_{x_1,\cdots,x_n} \mathbb{E}_p \\ \text{Subject to } \sigma_p^2 = 0.004 \\ \sum_{i=1}^{n} x_i = 1\end{gathered}$$

Since it is not a quadratic programming problem, we need to use the Lagrangian method to solve it.

We have the Lagrangian function

$$\begin{aligned}L(x_1,\cdots,x_n,\mu,\lambda) &= \mathbb{E}_p - \mu\left(\sum_{i=1}^{n} x_i - 1\right) - \lambda\left(\sigma_p^2 - 0.004\right) \\ &= \mathbf{x}^T\mathbf{r} - \mu\left(\mathbf{x}^T\mathbf{1} - 1\right) - \lambda\left(\mathbf{x}^T\mathbf{\Sigma}\mathbf{x} - 0.004\right) \\ \text{where } \mathbf{r} &= (\mathbb{E}_1,\cdots,\mathbb{E}_n)^T \\ \mathbf{\Sigma} &= (\sigma_{ij})_{n\times n}\end{aligned}$$

Now we can letting the derivatives of the Lagrangian with respect to $x_1, \cdots, x_n, \mu, \lambda$ be zero, which is

$$
\begin{aligned}
\frac{\partial L}{\partial x_i} &= \mathbf{r} - \mu^T \mathbf{1} - 2\lambda \mathbf{\Sigma x} = 0 \\
\frac{\partial L}{\partial \mu} &= \mathbf{x}^T \mathbf{1} - 1 = 0 \\
\frac{\partial L}{\partial \lambda} &= \mathbf{x}^T \mathbf{\Sigma x} - 0.004 = 0
\end{aligned}
$$

Hence, we can use MATLAB to solve the equation The code is shown in Appendix 4. The result is

- Weights of the portfolio:

AHT.L:	0.092301
CCH.L:	0.944393
FRAS.L:	−0.586839
MNG.L:	−0.194476
RMV.L:	0.126301
RR.L:	1.140152
SDR.L:	−0.442326
SHEL.L:	0.525337
STJ.L:	−0.046562
TSCO.L:	−0.558282

- Variance of the portfolio: 0.004000
- Expected return of the portfolio: 0.026772

Appendix

A MATLAB Code

Listing 1: Question a)

```
clear
filename = 'Historical Prices.xlsx';
opts = detectImportOptions(filename);
data = readtable(filename, opts);

% calculate weekly returns
prices = table2array(data(:, 2:end));
returns = (prices(2:end, :) - prices(1:end-1, :)) ./ ...
    prices(1:end-1, :);

% calculate expected returns and variances
expected_returns = mean(returns);
variances = var(returns);

% calculate covariances
covariances = cov(returns);

% set up the optimization problem
n_stocks = size(returns, 2);
H = 2 * covariances;
f = zeros(n_stocks, 1);

Aeq = ones(1, n_stocks);
beq = 1;

options = optimoptions('quadprog', 'Display', 'off');
[x, fval] = quadprog(H, f, [], [], Aeq, beq, [], [], [], ...
    options);

optimal_weights = x;
variance_portfolio = fval;
expected_return_portfolio = expected_returns * ...
    optimal_weights;

fprintf('Weights of the portfolio:\n');
for i = 1:n_stocks
    fprintf('%s: %f\n', ...
        data.Properties.VariableNames{i+1}, ...
```

```
        optimal_weights(i));
end
fprintf('Variance of the portfolio: %f\n', ...
    variance_portfolio);
fprintf('Expected return of the portfolio: %f\n', ...
    expected_return_portfolio);
```

Listing 2: Question b)

```
clear
filename = 'Historical Prices.xlsx';
opts = detectImportOptions(filename);
data = readtable(filename, opts);

% calculate weekly returns
prices = table2array(data(:, 2:end));
returns = (prices(2:end, :) - prices(1:end-1, :)) ./ ...
    prices(1:end-1, :);

% calculate expected returns and variances
expected_returns = mean(returns);
variances = var(returns);

% calculate covariances
covariances = cov(returns);

% set up the optimization problem
n_stocks = size(returns, 2);
H = 2 * covariances;
f = zeros(n_stocks, 1); % Since we are minimizing ...
    variance, there is no linear term

target_return = 0.015;
Aeq = [ones(1, n_stocks); expected_returns];
beq = [1; target_return];

options = optimoptions('quadprog', 'Display', 'off');
[x, fval] = quadprog(H, f, [], [], Aeq, beq, [], [], [], ...
    options);

optimal_weights = x;
variance_portfolio = fval;
expected_return_portfolio = expected_returns * ...
    optimal_weights;
```

```
fprintf('Weights of the portfolio:\n');
for i = 1:n_stocks
    fprintf('%s: %f\n', ...
        data.Properties.VariableNames{i+1}, ...
        optimal_weights(i));
end
fprintf('Variance of the portfolio: %f\n', ...
    variance_portfolio);
fprintf('Expected return of the portfolio: %f\n', ...
    expected_return_portfolio);
```

Listing 3: Question c)

```
clear
filename = 'Historical Prices.xlsx';
opts = detectImportOptions(filename);
data = readtable(filename, opts);

% calculate weekly returns
prices = table2array(data(:, 2:end));
returns = (prices(2:end, :) - prices(1:end-1, :)) ./ ...
    prices(1:end-1, :);

% calculate expected returns and variances
expected_returns = mean(returns);
variances = var(returns);

% calculate covariances
covariances = cov(returns);

kappa = 0.5;
n_stocks = length(expected_returns);
H = 2*kappa * covariances;
f = -expected_returns;

% linear equalities: sum(weights) = 1
Aeq = ones(1, n_stocks);
beq = 1;

options = optimoptions('quadprog', 'Display', 'off');
[x, fval] = quadprog(H, f, [], [], Aeq, beq, [], [], [], ...
    options);
```

```
optimal_weights = x;
utility = -fval;
expected_return_portfolio = expected_returns * ...
    optimal_weights;

fprintf('Weights of the portfolio:\n');
for i = 1:n_stocks
     fprintf('%s: %f\n', ...
         data.Properties.VariableNames{i+1}, ...
         optimal_weights(i));
end
fprintf('Maximum utility portfolio: %f\n', utility);
fprintf('Expected return of the portfolio: %f\n', ...
    expected_return_portfolio);
```

Listing 4: Question d)

```
clear
filename = 'Historical Prices.xlsx';
opts = detectImportOptions(filename);
data = readtable(filename, opts);

% calculate weekly returns
prices = table2array(data(:, 2:end));
returns = (prices(2:end, :) - prices(1:end-1, :)) ./ ...
    prices(1:end-1, :);

% calculate expected returns and variances
expected_returns = mean(returns);
variances = var(returns);

% calculate covariances
covariances = cov(returns);
n_stocks = length(expected_returns);

syms x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 mu lamb
weights = [x1, x2, x3, x4, x5, x6, x7, x8, x9, x10];

% Define the objective function
expected = expected_returns*weights';

% Define the constraints
g1 = sum(weights) - 1 == 0;
```

```
    var = weights * covariances * weights';
    g2 = var - 0.004 == 0;

    W = expected - mu * lhs(g1) - lamb * lhs(g2);

    % Calculate the derivatives of the Lagrangian
    dL_dx = arrayfun(@(i) diff(W, weights(i)) == 0, ...
        1:length(weights));
    dL_dmu = diff(W, mu) == 0;
    dL_dlamb = diff(W, lamb) == 0;

    system = [dL_dx, dL_dmu, dL_dlamb];
    solutions = vpasolve(system, [weights, mu, lamb]);

    optimal_weights = [solutions.x1, solutions.x2, ...
        solutions.x3, solutions.x4, solutions.x5, solutions.x6, ...
        solutions.x7, solutions.x8, solutions.x9, solutions.x10];
    expected_return_portfolio = sum(optimal_weights .* ...
        expected_returns);

    variance_portfolio = optimal_weights * covariances * ...
        optimal_weights';


    fprintf('Weights of the portfolio:\n');
    for i = 1:n_stocks
        fprintf('%s: %f\n', ...
            data.Properties.VariableNames{i+1}, ...
            optimal_weights(i));
    end
    fprintf('Variance of the portfolio: %f\n', ...
        variance_portfolio);
    fprintf('Expected return of the portfolio: %f\n', ...
        expected_return_portfolio);
```