Application of Matlab for Finance
Week 4

Shi, Yining

September 22, 2017

Today's Class

- Portfolio Optimization
  - Minimize portfolio variance
  - Maximize the Sharpe ratio

- Global minimum variance portfolio (GMVP)
- Optimization with constraints
- Construct the efficient frontier

Harvard Management Company Case Study

- Today you are the manager of Harvard Management Company, who manages and invests the endowment funds for the Harvard University.

- On a daily basis, you need to make investment decisions to allocate capital across a range of asset classes in order to satisfy your investors.

- File `HMC data.xls` reports relevant information on the real return of 12 asset classes: mean, standard deviation and variance-covariance matrix.
- You'd like to visually explore the data to understand the relationship between risk (standard deviation) and return among these assets.

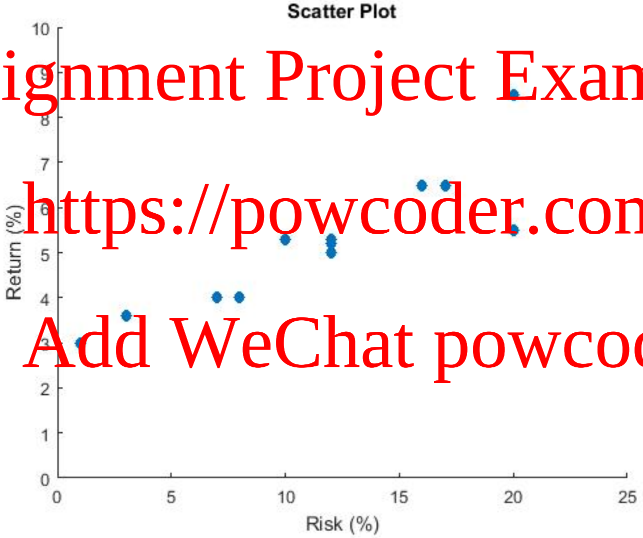
Read Data & Scatter Plot


```
[num, txt] = xlsread('HMC data.xls');
% read mean, std deviation and covariance matrix
avg_ret = num(:,1);
stdev = num(:,2);
cov_mat = num(:,3:end);
```



```
figure
scatter(stdev, avg_ret, 40,'filled'); % 40mm circle area ...
    with filled color
title('Scatter Plot');
xlabel('Risk (%)');
ylabel('Return (%)');
xlim([0 25]); % x-axis limits
ylim([0 10])
```

Scatter Plot



The Investment Decision

- The scatter plot confirms the traditional finance mantra
  - Higher risk, higher return.
- However, to efficiently manage your funds, you want the benefit of diversification in order to reduce risk.
- To do so, you will construct a portfolio contains these 12 assets, with a weight ω_i of your capital invested in asset i.
- Your objective is to find the optimal investment weight combination $w = [\omega_1, \omega_2, \ldots, \omega_{12}]'$ so that your portfolio variance σ^2 (i.e. the risk level) is the **minimum**.
- Meanwhile, sum of your weights equals to 1 as you invest **all** of your capital without borrowing.

Compute Portfolio Variance

- $w = [\omega_1, \omega_2, ..., \omega_{12}]'$ is a column vector (12x1) representing weights of capital allocated to different assets.
- Σ is the variance-covariance matrix with size (12x12)
- By definition, the portfolio variance can be calculated by

$$\sigma_p^2 = w' \Sigma w \tag{1}$$

- That is, portfolio σ_p^2 is a function of the weight vector w.
- **The optimization problem is to find the optimal weight combination w^* that minimizes the function output σ_p^{2*} with a constraint that sum of the portfolio weights w is 1.**

- We term such optimal portfolio as the **Global Minimum Variance Portfolio (GMVP).**

Global Minimum Variance Portfolio (GMVP)

- Formulate such portfolio optimization problem as follows:

$$\min \quad \sigma_p^2 = w'\Sigma w$$
$$s.t. \quad \mathbf{1}'w = 1$$

where $\mathbf{1}$ is a column vector of 1 with size (12x1) and

$$\mathbf{1}' * w = [1, 1, ..., 1] * \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{12} \end{bmatrix} = \omega_1 + \omega_2 + ... + \omega_{12} = 1$$

- We want to find the optimal w^* and the corresponding minimized $\sigma_p^{2*} = w^{*'}\Sigma w^*$

Optimization: `fmincon`

- `x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)`

$$\min_{x} f(x) \begin{cases} A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases}$$

- `fun`: the name of the objective function whose value to minimize
- `x0`: the starting point value of input x
- `A,b, Aeq, beq, lb, ub`: elements to define constraints
- `nonlcon`: the indicator for nonlinear constraint function
- If no specific value used, use `[]` to skip
- `options`: define for iterations, limits, increments for the optimization process as well as screen display.

Optimisation: `fmincon`

- `x1 = fmincon(@myfun(x),x0,..)` finds the optimal value `x1` that minimizes the result of function `myfun`, with starting point at `x0`.

- `x1 = fmincon(@(x)myfun(x,y),x0,..)` is the same as above
  - the function `myfun(x,y)` requires 2 inputs
  - `@(x)` defines that `x` is the optimizer and treats `y` as a constant
- `[x1, fval1] = fmincon(@(x)myfun(x,y),x0,..)` is the same as above, but with two outputs

  - `x1` is the optimal value of `x` that minimizes the function output
  - `fval1` is the minimized value of `myfun`
  - `fval1 = myfun(x1,y)`

Optimisation: `fmincon`

- `x1 = fmincon(@myfun(x),x0,A,b)` specifies the 1st type of constraint **`A*x`**$\leq$**`b`**
- `x1 = fmincon(@myfun(x),x0,A,b,Aeq,beq)` has 2 types of constraints **`Aeq*x = beq`**, in addition to `A*x`$\leq$`b`.
- `x1 = fmincon(@myfun(x),x0,[],[],Aeq,beq)` has 1 type of constraint `Aeq*x = beq` only.
- `x1 = fmincon(@myfun(x),x0,A,b,Aeq,beq,lb,ub)` has all 3 types of constraints with additional lower and upper bounds on `x`, so that the solution is always in the range **`lb`**$\leq$**`x`**$\leq$ **`ub`**.

Step 1: Define the Objective Function

- Going back to our global minimum variance portfolio question

$$\min \quad \sigma_p^2 = w'\Sigma w$$
$$s.t. \quad \mathbf{1}'w = 1$$

- Create a function file named `compute_pvar` outputs the portfolio variance (`pvar`) with inputs weight vector (`w`) and covariance matrix (`cov_mat`)


```
function [pvar] = compute_pvar( w, cov_mat )

% w is a column vector
pvar = w' * cov_mat * w;

end
```

Step 2: Define the Constraint

$$\min \ \sigma_p^2 = w'\Sigma w$$
$$s.t. \quad \mathbf{1}'w = 1$$

- Compare the actual constraint against the 3 types of constraint formats, and find the matching type

$$\mathbf{1}'w = 1$$

$$\begin{cases} A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases}$$

- `Aeq` $= \mathbf{1}' =$ `[1,1,..,1]`
- `beq = 1`

Step 3: Optimize

```
w1 = fmincon(@(w)compute_pvar(w, cov_mat), w0, [], ...
[], Aeq, beq,..)
```



```
% constraint: sum of weights is 1
N = length(avg_ret);
Aeq = ones(1, N);
beq = 1;
w0 = ones(N,1)*(1/N); % starting weight
% suppress message display
options = optimset('Display', 'off');

% option 1: return optimal weight vector w1
w1 = fmincon(@(w)compute_pvar(w, cov_mat), w0, [], [], ...
    Aeq, beq, [], [], [], options);
% option 2: return both the optiomal weight w1
%            and the minimized portfolio variance pvar1
[w1,pvar1] = fmincon(@(w)compute_pvar(w, cov_mat), w0, ...
    [], [], Aeq, beq, [], [], [], options);
disp(['The variance of GMVP is ', num2str(pvar1)]);
```

With Further Constraints

- Add two more constraints to the portfolio variance optimization:
  - NO short-selling is allowed. That is $w_i \geq 0 \quad \forall i \in [1, N]$.
  - Diversification Requirements: NO individual asset class can have a weight greater than 20%. That is $w_i \leq 0.2 \quad \forall i \in [1, N]$.

- The portfolio optimization problem is formulated as:

$$
\begin{aligned}
\min \quad & \sigma_P^2 = w' \Sigma w \\
s.t. \quad & \mathbf{1}' w = 1 \\
& 0 \leq w_i \leq 0.2 \quad \forall i \in [1, N]
\end{aligned}
$$

With Further Constraints

- Same objective function `compute_pvar`
- Define the constraints

$$\mathbf{1}'w = 1$$

$$0 \leq w_i \leq 0.2 \quad \forall i \in [1, N]$$

$$\begin{cases} A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases}$$

- `Aeq`$= \mathbf{1}' =$ `[1,1,..,1]`, and `beq = 1`
- `lb = [0,0,..,0]` and `ub =[0.2,0.2,..,0.2]`

With Further Constraints


```
%% With Further Constraints
% weights with ceiling (20%) and no short sell
Aeq = ones(1, N);
beq = 1;
lb = zeros(1, N);
ub = ones(1,N) * 0.2;

w0 = ones(N,1)*(1/N); % initial weight
w2 = fmincon(@(w)compute_pvar(w, cov_mat), w0, [], [], ...
    Aeq, beq, lb, ub, [], options);
[w2,pvar2] = fmincon(@(w)compute_pvar(w, cov_mat), w0, ...
    [], [], Aeq, beq, lb, ub, [], options);
disp(['The minimised portfolio variance is ', ...
    num2str(pvar2)]);
```

With a Target Return

- Minimize portfolio variance for a given target return 10%

$$\begin{aligned} \min \quad & w'\Sigma w \\ s.t. \quad & \mathbf{1}'w = 1 \\ & \mathbf{r}'w = 10\% \end{aligned}$$

where $\mathbf{r}$ is the column vector of asset returns (`avg_ret`).

$$\begin{cases} A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases}$$

- `Aeq` $= \begin{bmatrix} \mathbf{1}' \\ \mathbf{r}' \end{bmatrix}$, and `beq` $= \begin{bmatrix} 1 \\ 10 \end{bmatrix}$

With a Target Return


```
%% With a target return of 10%
Aeq = [ones(1,N); avg_ret'];
beq = [1; 10];

w0 = ones(N,1)*(1/N);
w3 = fmincon(@(w)compute_pvar(w, cov_mat), w0, [], [], ...
    Aeq, beq, [], [], [], options);
[w3,pvar3] = fmincon(@(w)compute_pvar(w, cov_mat), w0, ...
    [], [], Aeq, beq, [], [], [], options);
disp(['The minimised portfolio variance is ', ...
    num2str(pvar3)]);
```

Maximize Sharpe Ratio

- The portfolio Sharpe Ratio is calculated as

$$\text{Sharpe Ratio} = \frac{r_p - r_f}{\sigma_p}$$

$$r_p = w'\mathbf{r} \qquad \sigma_p = \sqrt{w'\Sigma w}$$

r_p is the portfolio return, $\mathbf{r}$ is the vector of individual asset's average return

- Step 1: Define the function

```
function [p_sharpe] = compute_sharpe(w, avg_ret, ...
    cov_mat)
% w and avg_ret are column vectors
pvar = w' * cov_mat * w;
pret = w' * avg_ret;
% assume risk free rate = 3% annually
% 3 as return in percentage
p_sharpe= (pret - 3) / sqrt(pvar);
end
```

Maximize Sharpe Ratio

- Maximise the Sharpe ratio allows a relative optimization
  - For the same level of risk σ_p, max Sharpe = max return r_p
  - For the same level of return r_p, max Sharpe = min risk σ_p
- Sharpe ratio is a function of portfolio weight $f(w)$
- The formulation of our maximization problem is

$$\begin{aligned} \max \quad & SR = f(w) \\ s.t. \quad & \mathbf{1}'w = 1 \end{aligned}$$

which is equivalent to the following minimization problem

$$\begin{aligned} \min \quad & -SR = -f(w) \\ s.t. \quad & \mathbf{1}'w = 1 \end{aligned}$$

Maximize Sharpe Ratio

- Put a negative sign – on the `compute_sharpe` objective function: maximization problem into a minimization one:


```
%% Maximize Sharpe ratio
Aeq = ones(1,N);
beq = 1;

w0 = ones(N,1) * (1/N);
% NOTE: use - to convert max optimization to a min ...
    optimization problem
w4 = fmincon(@(w)-compute_sharpe(w, avg_ret, ...
    cov_mat), w0, [], [], Aeq, beq, [], [], [], ...
    options);
[w4,fval4] = fmincon(@(w)-compute_sharpe(w, avg_ret, ...
    cov_mat), w0, [], [], Aeq, beq, [], [], [], ...
    options);
% fval4 = -sr1 as the fun = -compute_sharp
sr = - fval4;
disp(['The maximised portfolio Sharpe ratio is ', ...
    num2str(sr)]);
```

Construct Efficient Frontier

- The efficient frontier is the best return vs. risk investment combinations that one can obtain via portfolio diversification on risky assets.
- All assets that lie on the frontier are the most efficient.
- Any asset that is inside the frontier is not efficient or optimal
- To construct the efficient frontier, we minimize the portfolio variance for a range of target returns.
- We **exclude Cash** since it is the risk-free asset.
- Loop through all target return levels, find the optimal portfolio weights that minimize the portfolio risk with respect to the target return.
- With the optimal weights, calculate the portfolio variance and plot against the target return.

Construct Efficient Frontier

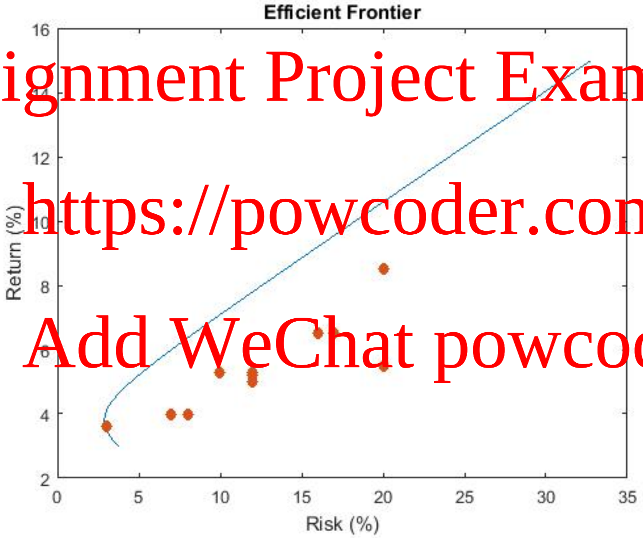

```
target_ret = 3:0.2:15;
M = length(target_ret);
pstdevs = zeros(M,1); % empty maxtrix for portfolio stdev
w0 = ones(N-1,1)*(1/N-1);

% Minimize variance for a range of target returns
for i=1:M %N-1 and end-1 to exclude cash
  Aeq = [ones(1,N-1); avg_ret(1:end-1)'];
  beq = [1; target_ret(i)];
  w_opt = fmincon(@(w) compute_pvar(w, cov_mat(1:end-1, ...
      1:end-1)), w0, [], [], Aeq, beq, [], [], [], options);
  pstdevs(i)  = sqrt(compute_pvar(w_opt, ...
      cov_mat(1:end-1, 1:end-1)));
end
```

Construct Efficient Frontier

- Plot the efficient frontier


```
% plot
figure
plot(pstdevs, target_ret)
hold on % hold on and add individual assets without cash
scatter(stdev(1:end-1), avg_ret(1:end-1), 40, 'filled');
title('Efficient Frontier')
xlabel('Risk (%)');
ylabel('Return (%)')
```

Construct Efficient Frontier



TakeAway

- After today's class, you will be able to finish the Coursework Question 1.
- Ensure that you write relevant comments on your CW code.
- When define the function for calculation, the dimensions of mean and standard deviation in the CW are not the same as in this class exercises'. Write the codes accordingly.
- Do not forget to assume your risk free rate.