Question 2A

First, I load the data and compute the percentage compounded return series.

Code:

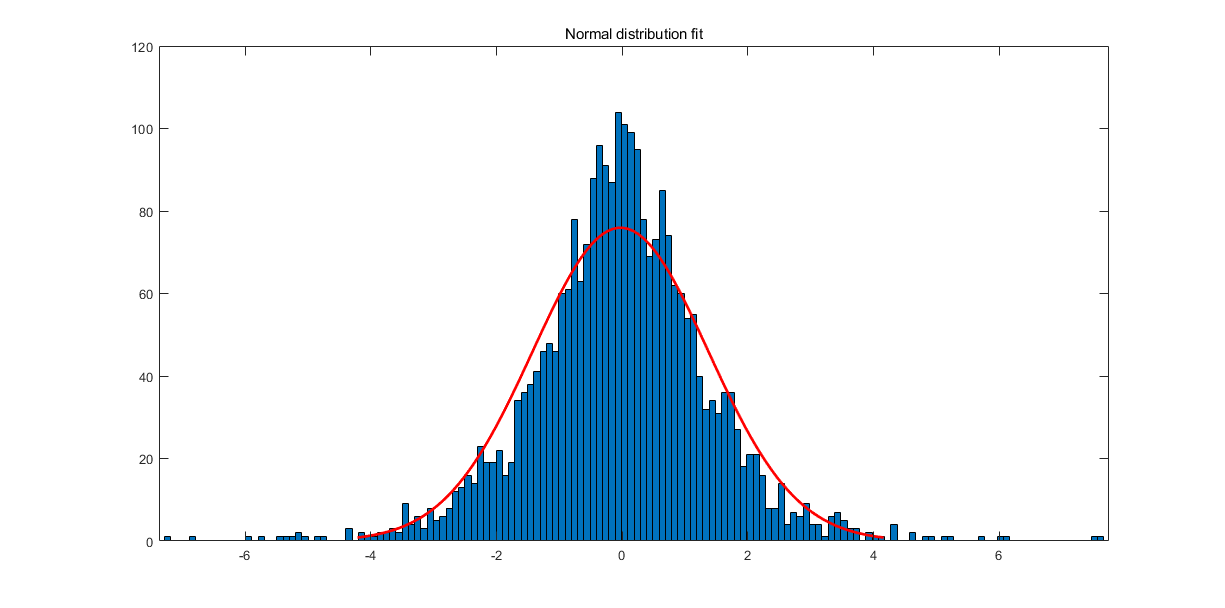
load Data\_GlobalIdx1.mat

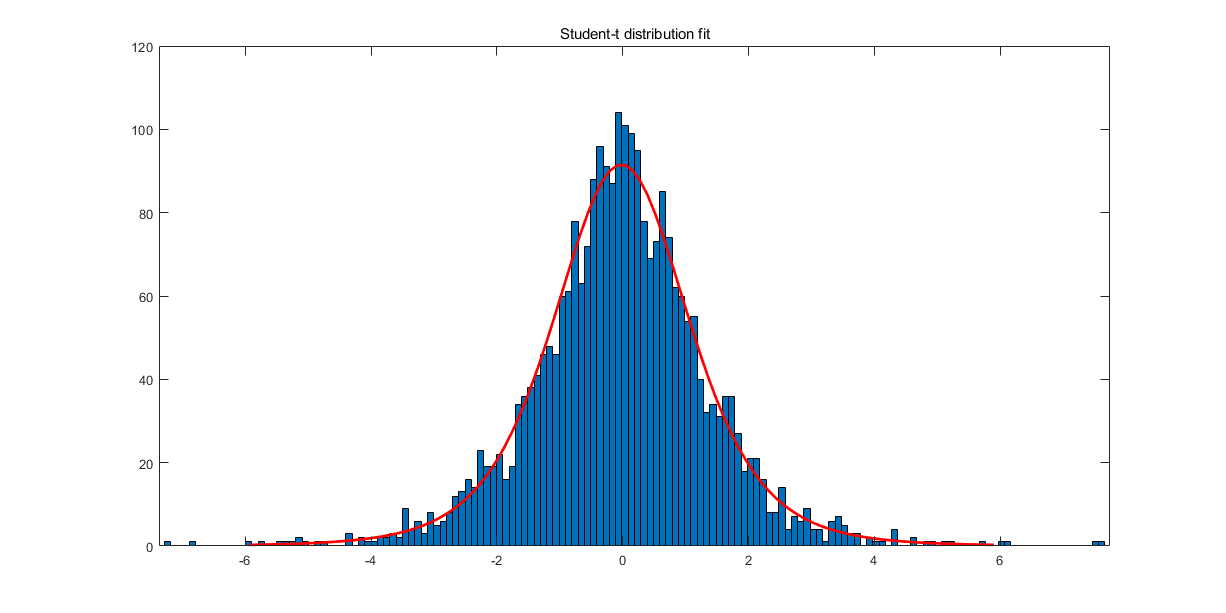
priceNIK = DataTable.NIK;

rNIK = 100\*price2ret(priceNIK);

T = length(rNIK);

a) Then, I plot the histogram, and fit with normal distribution and t-distribution, the plots are listed below:





Clearly, the student-t distribution fits better both at the tails and at the center.

Code:

figure

histfit(rNIK, 150, 'normal')

title('Normal distribution fit');

figure

histfit(rNIK, 150, 'tlocationscale')

title('Student-t distribution fit');

b) 2.5% VaR and 1% VaR

The daily 2.5% VaR is about -2.8691\*10^3, 1% VaR is around -3.5941\*10^3.

c) 2.5% ES and 1% ES

The daily 2.5% ES is about -3.7841\*10^3, 1% VaR is around -4.6643 \*10^3.

Code:

% 2.5% and 1% daily VaR

capital=100000;

sort\_rNIK = sort(rNIK);

p=0.025;

quantileVaR = round(p \* T);

VaRHS025 = capital \* sort\_rNIK(quantileVaR)/100; % 2.5% VaR

ESHS025 = capital \* mean(sort\_rNIK(1:quantileVaR))/100; % 2.5% ES

p=0.01;

quantileVaR = round(p \* T);

VaRHS001 = capital \* sort\_rNIK(quantileVaR)/100; % 1% VaR

ESHS001 = capital \* mean(sort\_rNIK(1:quantileVaR))/100; % 1% ES

d) Comments

Clearly the 1% VaR is larger than 2.5% VaR because only 1% of returns will below this amount historically speaking. The same applies to Expected Shortfall. Also since Expected Shortfall is the expected value of the returns below the threshold, this value will be smaller than the VaR as we expected. Here 1% VaR is about -3.5941\*10^3, means there is 1% possibility that we are going to lose at least RM3.5941&10^3 historically speaking, 1% Expected Shortfall is about -4.6643\*10^3, means the average of all losses in the worst 1% cases.

Question 3B

a) First, I construct the portfolio blotter, and get the total wealth and initial portfolio based on total wealth, price, and initial holdings. The total wealth is 9235000.

Code:

%01 setup the information of portfolio-blotter

Asset={'Forex', 'bond', 'large-cap equity', 'small-cap equity'};

Price=[73; 120; 32; 45];

Holding=[40000; 35000; 45000; 15000];

UnitCost=[10/10000;20/10000;10/10000;30/10000]; %transaction cost

Blotter=dataset({Price,'Price'},{Holding,'InitHolding'},'obsnames',Asset);

Wealth = sum(Blotter.Price.\*Blotter.InitHolding)

Blotter.InitPort=(Blotter.Price.\*Blotter.InitHolding)/Wealth;

Blotter.UnitCost=UnitCost;

disp(Blotter);

b) I use 'portsim' to simulate the returns and call 'ret2tick' to get prices.

The plot:



Code:

AssetMean=[0.08; 0.23; 0.15; 0.13];

AssetCovar=[ 0.0064 0.00408 0.00192 0.00000;

0.00408 0.0289 0.0204 0.0119;

0.00192 0.0204 0.0576 0.0336;

0.0000 0.0119 0.0336 0.1225];

X=portsim(AssetMean'/12,AssetCovar/12,72); %6 years

[Y,T]=ret2tick(X,[],1/12); %return to price

plot(T,log(Y));

title('Simulated Asset Price from Return series');

xlabel('Year');

ylabel('Log Price');

legend(Asset)

c) I set up the portfolio with the constraints. The portfolio is displayed below:

Portfolio - Attributes:

BuyCost: []

SellCost: []

RiskFreeRate: []

AssetMean: [4×1 double]

AssetCovar: [4×4 double]

TrackingError: []

TrackingPort: []

Turnover: []

BuyTurnover: []

SellTurnover: []

Name: 'Asset Allocation Portfolio'

NumAssets: 4

AssetList: {'Forex' 'bond' 'large-cap equity' 'small-cap equity'}

InitPort: [4×1 double]

AInequality: []

bInequality: []

AEquality: []

bEquality: []

LowerBound: [4×1 double]

UpperBound: []

LowerBudget: 1

UpperBudget: 1

GroupMatrix: [2×4 double]

LowerGroup: []

UpperGroup: [2×1 double]

GroupA: []

GroupB: []

LowerRatio: []

UpperRatio: []

MinNumAssets: []

MaxNumAssets: []

BoundType: [4×1 categorical]

To validate the portfolio:

lowerboundupperbound

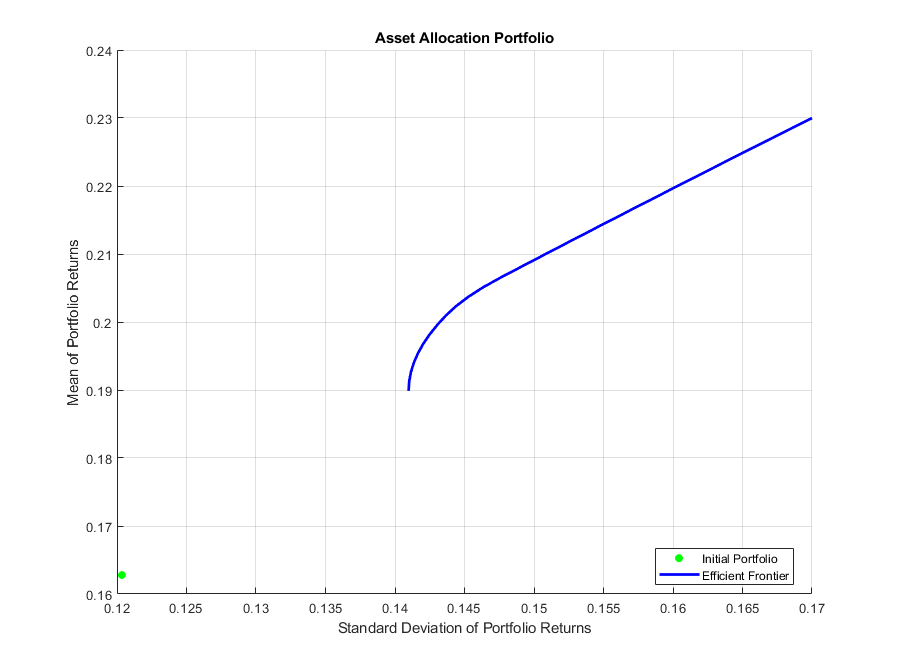
0.0000 0.1500

0.4000 1.0000

0.0000 0.4500

0.0000 0.4500

The bounds fit our constraints. The plotted efficient frontier:



Code:

%03 construct a portfolio

p=Portfolio('Name','Asset Allocation Portfolio','AssetList',Asset,'InitPort',Blotter.InitPort);

p=setDefaultConstraints(p);

p=setGroups(p,[1,0,0,0],[],0.15); %Equity allocation is no more than 85% of the portfolio

p=addGroups(p,[0,0,1,1],[],0.45); %Emerging equity is no more than 35% of the portfolio

p=setAssetMoments(p,AssetMean/12,AssetCovar/12); %Portfolio weights are nonnegative and sum to 1.

p=estimateAssetMoments(p,Y,'DataFormat','Prices');

p.AssetMean=12\*p.AssetMean;

p.AssetCovar=12\*p.AssetCovar;

disp(p);

%04 Validate your portfolio

[lowerbound, upperbound]=estimateBounds(p);

disp('lowerboundupperbound')

disp([lowerbound,upperbound]);

%05 Plot efficient frontier

figure;

plotFrontier(p,30);

hold on

Question 4A

a) I loaded the data and constructed the MA(1) - GJR(1, 1) model, the fitted equation is:

code:

load Data\_GlobalIdx1.mat

priceNIK = DataTable.NIK;

rNIK = 100\*price2ret(priceNIK);

T = length(rNIK);

model = arima('MALags',1,'Variance',gjr(1,1));

fit = estimate(model,rNIK);

b) The 1-period of one-day-ahead forecasts for the conditional return is -0.0296%, the volatility is 2.6018%.

Code:

[E0,V0] = infer(fit,rNIK);

[Y,YMSE,V] = forecast(fit,5,'Y0',rNIK,'E0',E0,'V0',V0);

c) The 2.5% VaR is about RM-3191.

Code:

p = 0.025;

capital = 100000;

gjrNquantileVaR = Y(1) + norminv(p,0,1) \* sqrt(V(1));

gjrNVaR = capital \* gjrNquantileVaR / 100;