## Contents



## MIE377 - Laboratory 5

The purpose of this program is to solve a robut optimization problem using the Michaud resampling technique. We will generate several estimates of the asset expected returns and covariance matrix through simulation.

```
clc
clear all
format short

% Program Start
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## PART 1: Data pre-processing

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Load the sample historical data
load('lab2data.mat')

% Calculate the asset and factor returns (factor models use returns, not
% prices)
rets    = prices( 2:end, : ) ./ prices( 1:end - 1, : ) - 1;
facRets = sp500price( 2:end , 1 ) ./ sp500price( 1:end - 1, 1 ) - 1;

% Number of assets
n = size(rets,2);

% Number of observations;
N = size(rets, 1);

% Calculate the asset expected return by taking the geometric mean
mu = (geomean(rets + 1) - 1)';

% Calculate the asset covariance matrix
Q = cov(rets);

% Calculate the factor expected excess return from historical data using
% the geometric mean
avgRet = geomean(facRets + 1) - 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## PART 2: Find the MVO efficient frontier using the nominal estimates

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% min   lambda * (x' * Q x)
% s.t.  mu' x  = R
%       sum(x) = 1
%
% Note: We have a variable target return "R" in order to construct the
% efficient frontier

%-------------------------------------------------------------------------
% 3.1 Equality constraints:
%-------------------------------------------------------------------------
Aeq = [-mu'; ones(1,n)];

%-------------------------------------------------------------------------
% 3.2 Find the MVO efficient frontier using the nominal estimates
%-------------------------------------------------------------------------

% Number of steps to estimate our efficient frontier
```

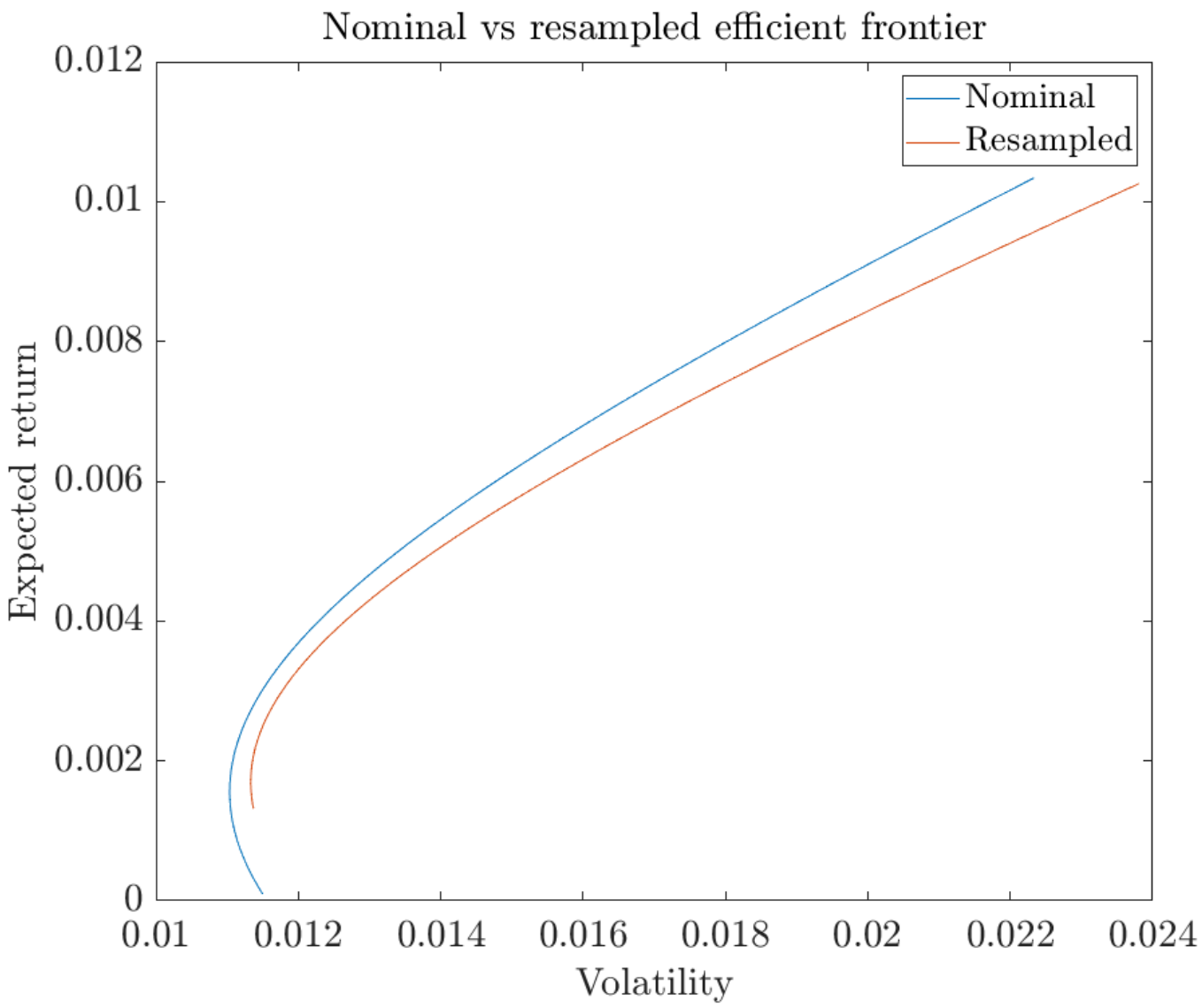
```
NoSteps = 100;

% Allocate space for our nominal MVO exp. return and volatility
MVOexpRet =  zeros(NoSteps,1);
MVOvol    = zeros(NoSteps,1);

% Increase the tolerance of 'quadprog'
options = optimoptions('quadprog','TolFun',1e-9, 'display','off');

% Solve the nominal MVO, increasing the target return at each step
for t = 1 : NoSteps

    % Set the target return proportional to our estimate 'targetRet'
    beq = [(-avgRet * t / 20); 1];

    % Find the nominal portfolio with 'quadprog'
    x = quadprog(  2 * Q, [], [], [], Aeq, beq, [], [], [], options );

    MVOexpRet(t) = mu' * x;
    MVOvol(t)    = sqrt( x' * Q * x );

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

### PART 4: Michaud resampling

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Number of simulations
NoSims = 30;

% Number of draws per simulation
T = 100;

% Number of steps to estimate our resampled efficient frontiers
NoStepsRes = 300;

% Allocate space for our sample portfolios per iteration
x = zeros(n, NoStepsRes, NoSims);

for i = 1 : NoSims

    % Draw the sample returns using our original expected returns and
    % covariance matrix
    sampleRets = mvnrnd(mu, Q, T);

    % Estimate our sample parameters
    mu_l = (geomean(sampleRets + 1) - 1)';
    Q_l  = cov(sampleRets);

    % Equality constraint matrix Aeq
    Aeq = [mu_l'; ones(1,n)];

    for t = 1 : NoStepsRes

        % Equality constraint constants beq
        beq = [(avgRet * t / 20); 1];

        % Solve the sample portfolio with 'quadprog'
        x(:,t,i) = quadprog(  2 * Q_l, [], [], [], Aeq, beq, [], [], [], options );

    end

end

x_avg = mean(x,3);
resampExpRet = (mu' * x_avg)';
resampVol    = sqrt( diag(x_avg' * Q * x_avg) );

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

### PART 5: Plot the nominal and resampled efficient frontiers

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fig2 = figure(2);
plot(MVOvol, MVOexpRet);
hold on
plot(resampVol, resampExpRet);
set(gca,'TickLabelInterpreter', 'latex','fontsize',22);
ylabel('Expected return','interpreter', 'latex','FontSize',22);
xlabel('Volatility','interpreter', 'latex','FontSize',22);
legend({'Nominal', 'Resampled'}, 'interpreter', 'latex');
title('Nominal vs resampled efficient frontier','interpreter', 'latex','FontSize',22);

set(fig2,'Units','Inches', 'Position', [0 0 10, 8]);
    pos3 = get(fig2,'Position');
    set(fig2,'PaperPositionMode','Auto','PaperUnits','Inches',...
        'PaperSize',[pos3(3), pos3(4)])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program End
```




*Published with MATLAB® R2023b*