Table of Contents


```
================================================================================

================================================================
```

```
Caso Virginia Investment Partners
File dati: dataset_rendimenti.csv
================================================================================
```

```
clc; clear; close all;
```

0. Caricamento dati

```
T = readtable('dataset_rendimenti.csv');   % Il file deve essere nel path

disp('Nomi colonne nel file:');
disp(T.Properties.VariableNames);

% Estraggo i rendimenti (escludo la colonna Date)
R = T{:, 2:end};   % matrice T x 4: [S&P, MSCI, LehmanAgg, IBM]

% Assegno per chiarezza
r_sp   = R(:,1);   % S&P 500
r_msci = R(:,2);   % MSCI World ex US
r_agg  = R(:,3);   % Lehman Brothers Aggregate Bond Index
r_ibm  = R(:,4);   % IBM
```

```
Warning: Column headers from the file were modified to make them valid MATLAB
identifiers before creating variable names for the table. The original column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column headers as
table variable names.
Nomi colonne nel file:
  Columns 1 through 4

    {'Date'}    {'S_P500'}    {'MSCIWorldIndex_E…'}    {'LehmanBrothersAg…'}

  Column 5
```

```
    {'IBM'}
```

1. Mean e standard deviation (mensile e annualizzata) per TUTTI gli asset

```
----------------------------------------------------------------------

assetNames = {'S&P 500', 'MSCI World ex US', 'Lehman Agg Bond', 'IBM'};

% Statistiche mensili
mu_month  = mean(R, 1);          % 1 x 4 (media mensile)
sigma_mon = std(R, 0, 1);        % 1 x 4 (dev.std mensile, sample)

% Annualizzazione
mu_ann    = (1 + mu_month).^12 - 1;   % da mensile ad annuale
sigma_ann = sigma_mon * sqrt(12);     % std annua

% Stampa tabella in percentuale
fprintf('================ STATISTICHE ASSET (%%) ================\n');
fprintf('%-20s  %12s  %12s  %12s  %12s\n', ...
        'Asset', 'Mean mon', 'Std mon', 'Mean ann', 'Std ann');
fprintf('%s\n', repmat('-',1,80));

for i = 1:numel(assetNames)
    fprintf('%-20s  %11.2f%%  %11.2f%%  %11.2f%%  %11.2f%%\n', ...
        assetNames{i}, ...
        mu_month(i)*100, sigma_mon(i)*100, ...
        mu_ann(i)*100,   sigma_ann(i)*100);
end

fprintf('%s\n\n', repmat('=',1,80));

% Estraggo IBM per comodo
mu_ibm_month  = mu_month(4);
sigma_ibm_mon = sigma_mon(4);
mu_ibm_ann    = mu_ann(4);
sigma_ibm_ann = sigma_ann(4);

================ STATISTICHE ASSET (%) ================
Asset                       Mean mon       Std mon      Mean ann       Std ann
--------------------------------------------------------------------------------
--
S&P 500                        0.78%         4.43%         9.71%        15.34%
MSCI World ex US               0.74%         4.36%         9.21%        15.11%
Lehman Agg Bond                0.51%         1.03%         6.31%         3.56%
IBM                            1.27%         9.38%        16.40%        32.49%
================================================================================
==
```

2. Portafoglio equally-weighted (S&P, MSCI, LehmanAgg)

```
----------------------------------------------------------------------

w_eq = [1/3; 1/3; 1/3];     % pesi equal-weighted per le 3 asset class
R_assets = [r_sp, r_msci, r_agg];   % solo le tre asset class

% Rendimenti mensili del portafoglio equally weighted
r_p_eq_month = R_assets * w_eq;

% Statistiche mensili
mu_p_eq_month  = mean(r_p_eq_month);
sigma_p_eq_mon = std(r_p_eq_month);

% Annualizzazione
mu_p_eq_ann    = (1 + mu_p_eq_month).^12 - 1;
sigma_p_eq_ann = sigma_p_eq_mon * sqrt(12);

fprintf('--- Punto 2: Portafoglio equally-weighted (S&P, MSCI, LehmanAgg) ---
\n');
fprintf('Media mensile:         %.2f%%\n', mu_p_eq_month*100);
fprintf('Dev.std mensile:       %.2f%%\n', sigma_p_eq_mon*100);
fprintf('Media annualizzata:    %.2f%%\n', mu_p_eq_ann*100);
fprintf('Dev.std annualizzata:  %.2f%%\n\n', sigma_p_eq_ann*100);
```

```
--- Punto 2: Portafoglio equally-weighted (S&P, MSCI, LehmanAgg) ---
Media mensile:         0.67%
Dev.std mensile:       2.76%
Media annualizzata:    8.40%
Dev.std annualizzata:  9.55%
```

3. Mean-variance optimizer con target sigma = 10%% (annualizzato)

```
----------------------------------------------------------------------

% Statistiche delle 3 asset class (annuali)
mu_assets_month = mean(R_assets);                % 1 x 3
mu_assets_ann   = (1 + mu_assets_month).^12 - 1; % 1 x 3
Sigma_month     = cov(R_assets);                 % 3 x 3
Sigma_ann       = Sigma_month * 12;              % matrice cov annua

% Target di volatilità (ANNUALE)
sigma_target_10 = 0.10;   % 10%

% Funzione obiettivo: max mu*w  <=> min -mu*w
% w(:) forza il vettore a colonna (3x1) così il prodotto è ben definito
f_obj = @(w) -mu_assets_ann * w(:);
```

```
% Vincoli lineari: somma pesi = 1, nessun altro vincolo lineare
Aeq = [1 1 1];
beq = 1;
A   = [];
b   = [];

% Limiti sui pesi (no short selling)
lb = zeros(3,1);
ub = ones(3,1);

% Vincolo non lineare: std_portafoglio = sigma_target
nonlcon_10 = @(w) port_sigma_constraint(w, Sigma_ann, sigma_target_10);

% Punto iniziale
w0 = ones(3,1) / 3;

options = optimoptions('fmincon', ...
    'Display', 'iter', ...
    'Algorithm', 'sqp');

[w_opt_10, fval_10, exitflag_10, output_10] = fmincon( ...
    f_obj, w0, A, b, Aeq, beq, lb, ub, nonlcon_10, options);

w_opt_10 = w_opt_10(:);   % forzo colonna

% Rendimento e rischio del portafoglio ottimale
mu_p_10    = mu_assets_ann * w_opt_10;
sigma_p_10 = sqrt(w_opt_10' * Sigma_ann * w_opt_10);

fprintf('--- Punto 3: Portafoglio ottimale con target sigma = 10%% ---\n');
fprintf('Pesi ottimali (S&P, MSCI, LehmanAgg): [%.3f  %.3f  %.3f]\n',
w_opt_10);
fprintf('Rendimento atteso annuo portafoglio:  %.2f%%\n', mu_p_10*100);
fprintf('Volatilità annua portafoglio:         %.2f%%\n\n', sigma_p_10*100);
```

4. Ripetizione per target sigma = 2%%, 6%%, 10%%, 14%%, 20%% %

```
sigma_targets = [0.02 0.06 0.10 0.14 0.20];  % annuali
nT = numel(sigma_targets);

W_opt   = zeros(3, nT);   % pesi per ciascun target
mu_p    = zeros(1, nT);   % rendimenti portafogli
sigma_p = zeros(1, nT);   % volatilità portafogli

for i = 1:nT
    sig_t = sigma_targets(i);
    nonlcon_i = @(w) port_sigma_constraint(w, Sigma_ann, sig_t);
```

```
    [w_i, fval_i, exitflag_i] = fmincon( ...
        f_obj, w0, A, b, Aeq, beq, lb, ub, nonlcon_i, options);

    w_i = w_i(:);
    W_opt(:,i) = w_i;
    mu_p(i)    = mu_assets_ann * w_i;
    sigma_p(i) = sqrt(w_i' * Sigma_ann * w_i);

    fprintf('--- Punto 4: Target sigma = %.0f%%%% ---\n', sig_t*100);
    fprintf('Pesi ottimali [S&P, MSCI, LehmanAgg]: [%.3f  %.3f  %.3f]\n',
w_i);
    fprintf('Rendimento atteso annuo: %.2f%%\n', mu_p(i)*100);
    fprintf('Volatilità annua:        %.2f%%\n\n', sigma_p(i)*100);
end
```

```
 Iter  Func-count            Fval   Feasibility   Step Length       Norm of
First-order

step    optimality
    0           4   -8.410322e-02     7.553e-02     1.000e+00
0.000e+00     9.711e-02
    1           8   -6.752995e-02     1.447e-02     1.000e+00
6.442e-01     3.598e-01
    2          14   -6.537682e-02     1.334e-02     4.900e-01
8.455e-02     3.476e-01
    3          18   -6.619113e-02     1.337e-02     1.000e+00
3.362e-02     1.279e-01
    4          49   -6.619113e-02     1.337e-02     1.578e-05
1.175e-06     1.279e-01

Converged to an infeasible point.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance but constraints are not
satisfied to within the value of the constraint tolerance.

--- Punto 4: Target sigma = 2%% ---
Pesi ottimali [S&P, MSCI, LehmanAgg]: [0.041  0.058  0.901]
Rendimento atteso annuo: 6.62%
Volatilità annua:        3.34%

 Iter  Func-count            Fval   Feasibility   Step Length       Norm of
First-order

step    optimality
    0           4   -8.410322e-02     3.553e-02     1.000e+00
0.000e+00     9.711e-02
    1           8   -7.631154e-02     1.437e-03     1.000e+00
3.031e-01     2.260e-01
    2          12   -7.596815e-02     7.623e-06     1.000e+00
1.400e-02     3.144e-03
    3          16   -7.597599e-02     4.528e-07     1.000e+00
3.424e-03     2.197e-03
```

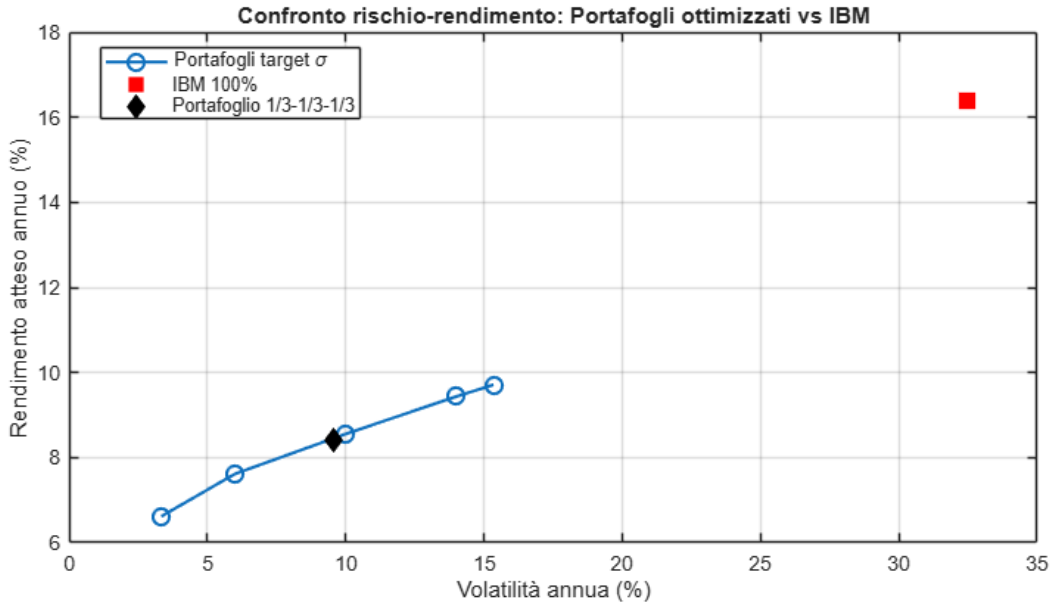
```
    4          20   -7.602218e-02     1.073e-05     1.000e+00
1.668e-02     1.961e-03
    5          24   -7.620111e-02     2.107e-04     1.000e+00
7.392e-02     9.738e-04
    6          28   -7.620253e-02     1.106e-04     1.000e+00
5.269e-02     1.088e-04
    7          32   -7.617451e-02     9.705e-08     1.000e+00
1.816e-03     2.135e-05
    8          36   -7.617449e-02     4.938e-11     1.000e+00
3.540e-05     1.601e-07

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

--- Punto 4: Target sigma = 6%% ---
Pesi ottimali [S&P, MSCI, LehmanAgg]: [0.301  0.097  0.602]
Rendimento atteso annuo: 7.62%
Volatilità annua:        6.00%

 Iter  Func-count            Fval   Feasibility   Step Length       Norm of
First-order

step    optimality
    0           4   -8.410322e-02     4.465e-03     1.000e+00
0.000e+00     9.711e-02
    1           8   -8.509312e-02     1.386e-05     1.000e+00
3.821e-02     3.111e-02
    2          12   -8.509953e-02     2.192e-07     1.000e+00
3.078e-03     2.257e-03
    3          16   -8.514574e-02     5.368e-06     1.000e+00
1.524e-02     2.096e-03
    4          20   -8.535291e-02     1.208e-04     1.000e+00
7.229e-02     1.532e-03
    5          25   -8.554304e-02     2.823e-05     1.000e+00
2.022e-01     7.270e-04
    6          29   -8.554949e-02     2.467e-07     1.000e+00
3.283e-03     1.840e-05
    7          33   -8.554943e-02     1.278e-09     1.000e+00
2.300e-04     2.763e-06
    8          37   -8.554943e-02     1.348e-11     1.000e+00
2.369e-05     2.531e-08

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

--- Punto 4: Target sigma = 10%% ---
Pesi ottimali [S&P, MSCI, LehmanAgg]: [0.551  0.128  0.321]
Rendimento atteso annuo: 8.55%
```

```
Volatilità annua:        10.00%

 Iter  Func-count            Fval   Feasibility   Step Length       Norm of
First-order

step    optimality
    0           4   -8.410322e-02     4.447e-02     1.000e+00
0.000e+00     9.711e-02
    1           8   -9.387471e-02     9.708e-04     1.000e+00
3.792e-01     3.221e-01
    2          12   -9.367877e-02     3.701e-07     1.000e+00
8.559e-03     2.888e-03
    3          16   -9.369346e-02     3.735e-07     1.000e+00
4.756e-03     2.517e-03
    4          20   -9.376566e-02     9.148e-06     1.000e+00
2.354e-02     2.388e-03
    5          24   -9.408820e-02     2.058e-04     1.000e+00
1.117e-01     1.793e-03
    6          28   -9.469203e-02     1.540e-03     1.000e+00
3.053e-01     4.966e-04
    7          32   -9.435808e-02     3.655e-06     1.000e+00
2.167e-02     4.834e-04
    8          36   -9.435742e-02     1.800e-07     1.000e+00
3.232e-03     2.407e-05
    9          40   -9.435742e-02     9.178e-08     1.000e+00
2.313e-03     1.022e-07

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

--- Punto 4: Target sigma = 14%% ---
Pesi ottimali [S&P, MSCI, LehmanAgg]: [0.785  0.157  0.058]
Rendimento atteso annuo: 9.44%
Volatilità annua:        14.00%

 Iter  Func-count            Fval   Feasibility   Step Length       Norm of
First-order

step    optimality
    0           4   -8.410322e-02     1.045e-01     1.000e+00
0.000e+00     9.711e-02
    1           8   -9.537150e-02     5.436e-02     1.000e+00
4.638e-01     1.418e+01
    2          12   -9.592781e-02     5.262e-02     1.000e+00
1.561e-01     2.081e+00
    3          16   -9.666154e-02     4.922e-02     1.000e+00
2.059e-01     3.267e-01
    4          20   -9.710731e-02     4.657e-02     1.000e+00
1.251e-01     6.501e-02
    5          21   -9.710731e-02     4.657e-02     7.000e-01
3.381e-16     6.501e-02
```

```
Converged to an infeasible point.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance but constraints are not
satisfied to within the value of the constraint tolerance.

--- Punto 4: Target sigma = 20%% ---
Pesi ottimali [S&P, MSCI, LehmanAgg]: [1.000  0.000  0.000]
Rendimento atteso annuo: 9.71%
Volatilità annua:        15.34%
```

5. Grafico rischio-rendimento (portafogli ottimizzati vs IBM)

```
----------------------------------------------------------------------

figure;
hold on; grid on; box on;

% Portafogli ottimizzati (per i vari target di sigma)
plot(sigma_p*100, mu_p*100, 'o-', 'LineWidth', 1.5, 'MarkerSize', 8);

% Punto IBM
plot(sigma_ibm_ann*100, mu_ibm_ann*100, ...
     'rs', 'MarkerSize', 10, 'MarkerFaceColor', 'r');

% Punto portafoglio equally weighted
plot(sigma_p_eq_ann*100, mu_p_eq_ann*100, ...
     'kd', 'MarkerSize', 8, 'MarkerFaceColor', 'k');

xlabel('Volatilità annua (%)');
ylabel('Rendimento atteso annuo (%)');
title('Confronto rischio-rendimento: Portafogli ottimizzati vs IBM');
legend({'Portafogli target \sigma', 'IBM 100%', 'Portafoglio 1/3-1/3-1/3'},
...
       'Location', 'best');

hold off;
```



==========

==========

Funzione di vincolo non lineare per fmincon

```
==========================================================================
```

```
function [c, ceq] = port_sigma_constraint(w, Sigma, sigma_target)
    % Forzo w a vettore colonna
    w = w(:);
    sigma_p = sqrt(w' * Sigma * w);
    c   = [];                       % nessun vincolo di tipo <=
    ceq = sigma_p - sigma_target;   % vincolo di uguaglianza
end
```

```
 Iter  Func-count            Fval   Feasibility   Step Length       Norm of
First-order

step    optimality
    0           4   -8.410322e-02     4.465e-03     1.000e+00
0.000e+00     9.711e-02
    1           8   -8.509312e-02     1.386e-05     1.000e+00
3.821e-02     3.111e-02
    2          12   -8.509953e-02     2.192e-07     1.000e+00
3.078e-03     2.257e-03
    3          16   -8.514574e-02     5.368e-06     1.000e+00
1.524e-02     2.096e-03
    4          20   -8.535291e-02     1.208e-04     1.000e+00
7.229e-02     1.532e-03
    5          25   -8.554304e-02     2.823e-05     1.000e+00
2.022e-01     7.270e-04
    6          29   -8.554949e-02     2.467e-07     1.000e+00
3.283e-03     1.840e-05
```

```
    7          33   -8.554943e-02     1.278e-09     1.000e+00
2.300e-04     2.763e-06
    8          37   -8.554943e-02     1.348e-11     1.000e+00
2.369e-05     2.531e-08

Feasible point with lower objective function value found, but optimality
criteria not satisfied. See output.bestfeasible..


Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

--- Punto 3: Portafoglio ottimale con target sigma = 10% ---
Pesi ottimali (S&P, MSCI, LehmanAgg): [0.551  0.128  0.321]
Rendimento atteso annuo portafoglio:  8.55%
Volatilità annua portafoglio:         10.00%
```

Published with MATLAB® R2025b