Optimisation in Machine Learning

Lab Assignment - 3

Ayush Abrol B20AI052

```
import pandas as pd
import numpy as np
import cvxopt as cp
import cvxpy as cvx
import matplotlib.pyplot as plt
import yfinance as yf
```

Looking Daily Close Price Historical Data from 30 August, 2021 to 30 August, 2022 for any 10 stocks.

Stocks selected: GOOG, AAPL, MS, AMZN, UBER, TSLA, NVDA, INTC, AMD, BABA

```
data = yf.download("GOOG AAPL MS AMZN UBER TSLA NVDA INTC AMD BABA",
                   start="2021-08-30", end="2022-08-30")
```

```
[**********100%**********]  10 of 10 completed
```

```
data.head()
```

	Adj Close						
	AAPL	AMD	AMZN	BABA	GOOG	INTC	MS

```
df = data['Adj Close']
```

2021-	152.266739	111.320000	171.078506	162.289993	145.469498	52.218403	100.829178

```
df.head()
```

Date	AAPL	AMD	AMZN	BABA	GOOG	INTC	MS
2021-08-30	152.266739	111.320000	171.078506	162.289993	145.469498	52.218403	100.829178
2021-08-31	150.983948	110.720001	173.539505	166.990005	145.462006	52.334572	101.158539
2021-09-01	151.660172	109.989998	173.949997	173.279999	145.841995	51.957020	101.042290
2021-							

```
B = df.values
B.shape
```

```
(252, 10)
```

```
df.isnull().sum()
```

```
AAPL    0
AMD     0
AMZN    0
BABA    0
GOOG    0
INTC    0
MS      0
NVDA    0
TSLA    0
UBER    0
dtype: int64
```

```
Q = np.cov(B.T)      # Creating a covariance matrix
```

```
Q
```

```
array([[ 157.74377628,  125.09336857,  121.01896287,  -74.42082694,
          64.39092963,   13.44375875,   40.54330095,  335.77141544,
         391.54866965,   23.5291484 ],
       [ 125.09336857,  439.04994113,  377.71547605,  250.72765389,
         229.59182499,   75.66226151,  130.22328946,  969.71226696,
         742.57837965,  121.74834565],
       [ 121.01896287,  377.71547605,  540.13457818,  420.63647601,
         296.01481426,  103.04531264,  168.84568087,  943.04805374,
         722.52832314,  173.67593213],
```

```
              [ -74.42082694,  250.72765389,  420.63647601,  711.69523388,
                253.5054141 ,   98.30137001,  156.78364348,  557.42906608,
                267.38458803,  169.15405922],
              [  64.39092963,  229.59182499,  296.01481426,  253.5054141 ,
                178.65881525,   63.05593838,   96.33548689,  570.1690946 ,
                406.11457314,   97.11347355],
              [  13.44375875,   75.66226151,  103.04531264,   98.30137001,
                 63.05593838,   32.01402523,   36.17833517,  191.5244935 ,
                110.80221224,   37.48980898],
              [  40.54330095,  130.22328946,  168.84568087,  156.78364348,
                 96.33548689,   36.17833517,   74.20824867,  291.59415384,
                200.25522591,   61.13232898],
              [ 335.77141544,  969.71226696,  943.04805374,  557.42906608,
                570.1690946 ,  191.5244935 ,  291.59415384, 2366.6687094 ,
               1870.79552595,  282.71169375],
              [ 391.54866965,  742.57837965,  722.52832314,  267.38458803,
                406.11457314,  110.80221224,  200.25522591, 1870.79552595,
               2279.17983157,  218.12060348],
              [  23.5291484 ,  121.74834565,  173.67593213,  169.15405922,
                 97.11347355,   37.48980898,   61.13232898,  282.71169375,
                218.12060348,   66.56793673]])
```

```
c = np.array([[0] for i in range (10)])
m = -np.log(np.array([B[-1][i]/B[0][i] for i in range (10)]))
m
```

```
array([-0.05812802,  0.22951941,  0.27620485,  0.51880609,  0.27639995,
        0.46074729,  0.16666735,  0.3610824 , -0.15617936,  0.32028669])
```

```
R = 52 #B20AI052
val = R/1000
```

```
A = np.column_stack((m, np.negative(np.identity(10, dtype = 'int'))))
b = np.array([-val]+[0]*10)
Aeq = np.array([[1]*10])
beq = np.array([[1]])
A.shape
```

```
(10, 11)
```

```
A = A.T
A.shape
```

```
(11, 10)
```

```
sol = cp.solvers.qp(cp.matrix(Q, tc='d'), cp.matrix(c, tc='d'), cp.matrix(A, tc='d'),
                    cp.matrix(b, tc='d'), cp.matrix(Aeq, tc='d'), cp.matrix(beq, tc='d'))
print(sol['x'])
```

```
     pcost       dcost       gap    pres   dres
 0:  3.7802e+00  3.5056e+00  2e+01  5e+00  4e+00
 1:  3.8295e+00  4.4471e+00  6e+00  1e+00  1e+00
 2:  1.1823e+01  1.9949e+01  2e+01  1e+00  9e-01
 3:  1.7238e+01  3.3746e+01  2e+01  6e-01  5e-01
```

```
 4:  2.1496e+01  3.8895e+01  2e+01  4e-01  3e-01
 5:  2.9321e+01  5.0953e+01  1e+01  2e-01  2e-01
 6:  4.0262e+01  6.3981e+01  1e+01  1e-01  1e-01
 7:  7.5309e+01  7.5476e+01  7e+00  2e-02  1e-02
 8:  7.4515e+01  7.4930e+01  5e+00  9e-03  8e-03
 9:  7.5854e+01  7.5598e+01  4e-01  2e-04  2e-04
10:  7.5735e+01  7.5732e+01  5e-03  3e-06  2e-06
11:  7.5733e+01  7.5733e+01  5e-05  3e-08  2e-08
Optimal solution found.
[ 9.73e-01]
[ 6.64e-08]
[ 2.17e-08]
[ 3.31e-09]
[ 4.97e-08]
[ 2.55e-08]
[ 2.73e-02]
[ 3.32e-09]
[-2.36e-09]
[ 7.52e-08]
```

```
print("After rounding upto 5 decimal places: ")
for i in range(10):
    print(round(sol['x'][i], 5))

print("The minimum variance portfolio is: ", sol['primal objective'])
```

```
After rounding upto 5 decimal places:
0.97274
0.0
0.0
0.0
0.0
0.0
0.02726
0.0
-0.0
0.0
The minimum variance portfolio is:  75.73303129867864
```

Colab paid products - Cancel contracts here