Instructions:

Try to solve all problems on your own. If you have difficulties, ask the instructor or TAs.

Please follow the instructions given below to prepare your solution notebooks:

- Please use different notebooks for solving different Exercise problems.
- The notebook name for Exercise 1 should be `ROLLNUMBER-labLL-ex1.ipynb`. `ROLLNUMBER-labLL-ex2.ipynb`, etc for others. 'LL' is the two digit lab number (lab-3 is 03, etc).
- Please ask your doubts to TAs or instructors or post in Moodle Discussion Forum channel.
- You should upload on the .ipynb files on Moodle (one per exercise).

Only the questions marked [**R**] need to be answered on paper. Write legible and to-the-point explanations. The work-sheet on which you write needs to be submitted before leaving the session.

Some other questions require plotting graphs (histograms, trajectories, level-sets etc) or tables. Please make sure that the plots are present in the submitted ipython notebooks.

Submission Time: Please check the submission deadline as show on the assignment web-page in Moodle. Late submissions will be accepted upto 24 hours from the deadline. All late submissions will have a penalty of 3 marks. Submissions later than 24 hours after the deadline will not be accepted.

The fourth laboratory exercise aims to helps us understand decision making under uncertainty.

Exercise 1[10 marks] Suppose you are responsible for investing ₹1 crore in stocks of a given set of companies. If you knew beforehand the rate of return for each stock, you would invest the whole amount in the stock with highest returns. However, stock prices can fluctuate, and it is not obvious how to select good stocks to invest in. One may want to consider past information about stocks to make a decision, like we will do today.

Suppose for our example, we know prices (per share) of 30 stocks over the past 120 days. See 'Stock_Closing_Price.csv' on Moodle for the data. Further suppose that we want to invest available amount for a 5 days period today, distributed amongst these 30 stocks. How much money will you invest in each? Let us start with the following exercises.

1. First let us estimate the average rate of return of each stock over a 5-day period. You can assume that the expected return is just the average rate of return in all 5-day periods available in the past data. Compute this average for all 30 stocks.
2. Next compute the standard deviation of the rate of return of each stock separately.
3. Are any stocks co-related? For this, you can compute covariance of rate of returns. The 'cov' function in numpy package provides this functionality. Check if the standard deviation computed in the step above is consistent with the covariances.
4. [**R**] Report the stock with the highest average rate of return. What is its standard deviation? Also find 3 pairs of stocks that are highly corelated to each other (positively or negatively).
5. Suppose we want to attain a target rate of return (say, for example 0.00025) with minimum possible risk. This can be modeled as the following quadratic optimization model:

$$\begin{aligned} \min\, & x^T \sigma x (\text{risk}) \\ \text{subject to: } & \sum_i x_i = 1,00,00,000 \\ & \sum_i \mu_i x_i \geq 0.00025 \times (??)(\text{return-rate}) \\ & x \geq 0 \end{aligned}$$

6. Solve the above model using Pyomo and a suitable choice of a solver. Decide how you will provide the parameters (μ_i and the (??) in the rhs). Solve the model for several values of expected return-rates. Draw a graph of risk vs return over carefully chosen values of return-rates.
7. Now suppose you would like to use a different approach where the risk is constrained and returns are maximized. Write an optimization model and solve it for carefully chosen parameters. Plot the maximum rate of return as a function of risk. Check if this is consistent with the previous model.
8. If you spend equal amount of money in all 30 stocks, what is your expected rate of return and the risk? Would you recommend this strategy?