FTEC2101/ESTR2520 Optimization Methods **Spring 2023**

Project Specification – Portfolio Optimization

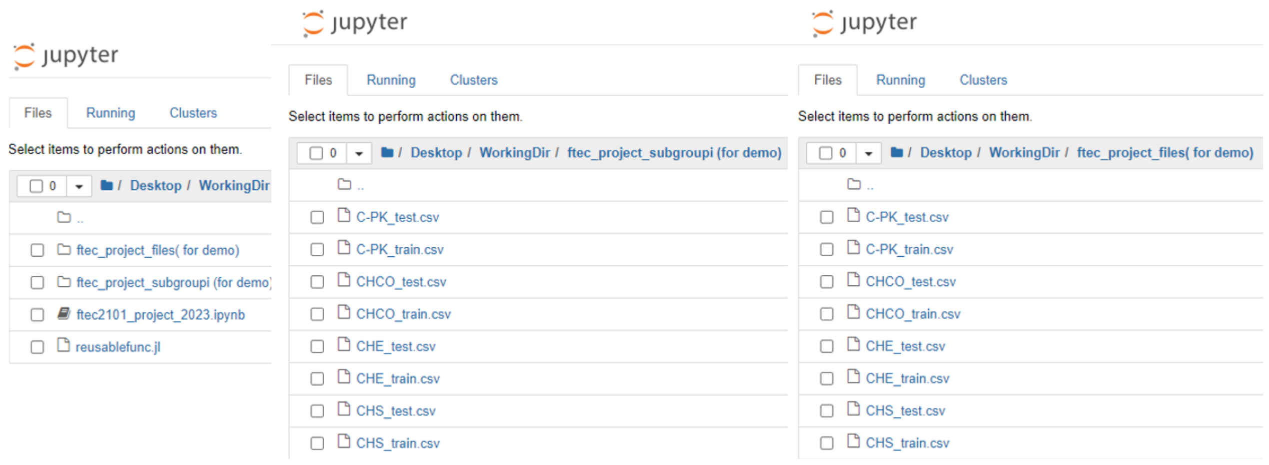
Last Updated: April 30, 2023, Deadline: May 15, 2023, 23:59 (HKT)

So far in FTEC2101/ESTR2520, we have learnt (or will learn) a number of theories about optimization methods, ranging from the simplex method for linear programming, modeling techniques for integer programming, convex optimization, gradient/Newton methods for unconstrained optimization, KKT conditions, SOCP formulation, etc.. Besides the theories, it is important to remember that optimization methods are practical tools for solving real world problems. The aim of the current project is to highlight on such practical aspects of optimization.

This project implements practical solutions for portfolio optimization using `Julia`[1]. We will explore various aspects of portfolio design and more importantly, *implement* these ideas on a real-world dataset.

Dataset & Folder Setting In this project, we will use a dataset gathered from the NYSE market with stock prices of 699 stocks between 2018 and 2022. To prepare your working environment, please retrieve from Blackboard the Jupyter notebook template `ftec2101_project_2023.ipynb`, a helper function `reusablefunc.jl`, and the dataset archives `ftec_project_files.zip`, `ftec_project_subgroup`i`.zip` (where $i \in \{1, ..., 5\}$ – depending on your assigned subgroup).

As usual, it is a good habit to move the files to a working directory of your choice, and unzip the archives. Your working directory shall contain the following content:



```
jupyter
Files   Running   Clusters
Select items to perform actions on them.
0  / Desktop / WorkingDir
..
ftec_project_files( for demo)
ftec_project_subgroupi (for demo)
ftec2101_project_2023.ipynb
reusablefunc.jl
```

```
jupyter
Files   Running   Clusters
Select items to perform actions on them.
0  / Desktop / WorkingDir / ftec_project_subgroupi (for demo)
..
C-PK_test.csv
C-PK_train.csv
CHCO_test.csv
CHCO_train.csv
CHE_test.csv
CHE_train.csv
CHS_test.csv
CHS_train.csv
```

```
jupyter
Files   Running   Clusters
Select items to perform actions on them.
0  / Desktop / WorkingDir / ftec_project_files( for demo)
..
C-PK_test.csv
C-PK_train.csv
CHCO_test.csv
CHCO_train.csv
CHE_test.csv
CHE_train.csv
CHS_test.csv
CHS_train.csv
```

The folders `ftec_project_subgroup`i, `ftec_project_files` contain essentially the same content of the stock data of 20 stocks and 699 stocks, respectively. The latter contains the complete dataset that includes the former, which will be used in the second part of the project. Furthermore, for each stock, the dataset is split a training dataset and testing dataset, as follows:

- For compulsory task, `*_train.csv` – stock prices for *training*, taken from Aug, 2019 to Oct, 2019.

[1] As an alternative, you are welcomed to use `Python` with optimization modeling packages supporting SOCP and MI-NLP such as `cvxpy`. However, you have to notify the instructor about the choice and the plan on how you wish to accomplish the project in the latter case on or before May 1, 2023, i.e., two weeks before the deadline.

- For compulsory task, `*_test.csv` – stock prices for *testing*, taken from Oct, 2019 to Dec, 2022.
- For competitive task, `*_train.csv` – stock prices for *training*, taken from Feb, 2018 to May, 2021.
- For competitive task, `*_test.csv` – stock prices for *testing*, taken from May, 2021 to Dec, 2022.

1.1 Compulsory Tasks (50%)

Suppose that there are n stocks in the market that can be invested. We begin our study by considering the following simplified version of mean-variance Portfolio Optimization problem[2]:

$$\begin{aligned} \min_{\boldsymbol{p} \in \mathbb{R}^n} \quad & \lambda \boldsymbol{p}^\top \boldsymbol{\Sigma} \boldsymbol{p} - \boldsymbol{p}^\top \bar{\boldsymbol{r}} \\ \text{s.t.} \quad & \mathbf{1}^\top \boldsymbol{p} = B, \end{aligned} \tag{1.1}$$

where $\lambda > 0$ is a user-defined parameter (for regulating the risk of the selected portfolio), $B > 0$ is a fixed budget, and $\mathbf{1}$ is an n-dimensional all-one vector. We have defined the covariance matrix and expected return vector for each stock respectively as:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & \rho_{22} & & \rho_{2n} \\ \vdots & & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \cdots & \rho_{nn} \end{pmatrix}, \quad \bar{\boldsymbol{r}} = \begin{pmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \vdots \\ \bar{r}_n \end{pmatrix} \tag{1.2}$$

such that for any $i, j = 1, ..., n$, it holds

$$\bar{r}_i = \mathbb{E}_t[R_i(t)], \quad \rho_{ij} = \mathbb{E}_t[(R_i(t) - \bar{r}_i)(R_j(t) - \bar{r}_j)], \quad R_i(t) = \frac{\text{closing price on day } t - \text{opening price on day } t}{\text{opening price on day } t}, \tag{1.3}$$

i.e., $R_i(t)$ is the return of stock i on day t as they were defined during the lectures. Notice that (1.1) is a constrained optimization problem, which can be shown to be a convex optimization since $\boldsymbol{\Sigma}$ must be a PSD matrix. Throughout this project, we assume that $\boldsymbol{\Sigma}$ is also non-singular.

Theory & Modeling

Task 1: (5%)

Consider the portfolio optimization problem given in (1.1). Answer the following questions:

(a) Derive the KKT conditions for (1.1) and show that the optimal solution for (1.1) is:

$$\boldsymbol{p}_\lambda^\star = \frac{1}{2\lambda} \boldsymbol{\Sigma}^{-1} \big(\bar{\boldsymbol{r}} + \frac{2\lambda B - \mathrm{r}_1}{\mathrm{r}_2} \mathbf{1} \big),$$

where

$$\mathrm{r}_1 = \mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{r}}, \quad \mathrm{r}_2 = \mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \mathbf{1}.$$

(b) Define a fixed desired return as R_{d}. Calculate the range of λ such that the expected return satisfies

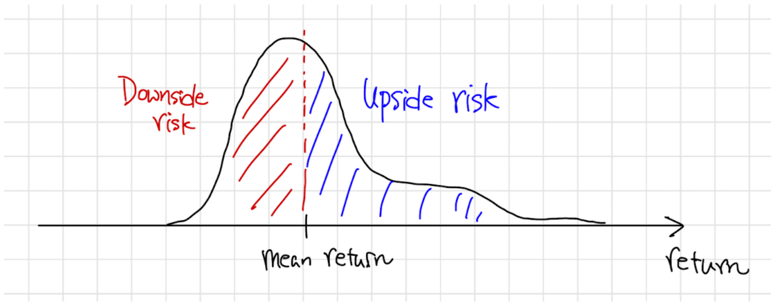
$$\bar{\boldsymbol{r}}^\top \boldsymbol{p}_\lambda^\star \geq R_{\mathsf{d}}.$$

[2]Note that unlike the examples given in the lectures, we do not constraint $\boldsymbol{p}$ to be non-negative in this simplified task.

One-sided Risk. The main aim of this project is to extend portfolio optimization to cover some realistic aspects of the problem. We first recall the following Markowitz's mean-variance Portfolio Optimization problem that we have studied during the class: let $\mathbf{1}$ denotes the all-one vector:

$$\begin{aligned} \min_{\boldsymbol{p}\in\mathbb{R}^n} \quad & \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{\Sigma}\boldsymbol{p} \\ \text{s.t.} \quad & \mathbf{1}^\top \boldsymbol{p} = B,\ \bar{\boldsymbol{r}}^\top \boldsymbol{p} \geq R_{\mathsf{d}},\ \boldsymbol{p} \geq \mathbf{0}. \end{aligned} \tag{1.4}$$

The above portfolio optimization problem (as well as (1.1)) considers the *risk* of a portfolio as the *variance* of the return given by the portfolio. Notice that the variance equally measures the *upside risk* and *downside risk*[3]. The objective function penalizes *any deviation* of the selected portfolio from the expected return.



This possibly deviate from the practical needs since an upside risk (e.g., the actual return is higher than the expected one) is not harmful. Instead, ideally one may focus on reducing the downside risk only.

The aim of this project is to investigate different modifications to the portfolio optimization problem that focus on the downside risk. To this end, the first idea we shall implement is to minimize the probability of the event when the actual return is smaller than the expected return, i.e.,

$$\mathbb{P}\{\boldsymbol{p}^\top \bar{\boldsymbol{r}} > \boldsymbol{p}^\top \boldsymbol{R}_t\}$$

where $\mathbb{P}\{\mathcal{A}\}$ denotes the probability of event $\mathcal{A}$, $\boldsymbol{R}_t = [R_1(t), R_2(t), \ldots, R_n(t)]^\top$, and $\boldsymbol{R}_t \in \mathbb{R}^n$ is the actual (random) return of n stocks on day t. With a sufficient number of historical records for the stocks, the above probability can be *approximated* by

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{p}^\top \bar{\boldsymbol{r}} - \boldsymbol{p}^\top \boldsymbol{R}_t > 0\}, \quad \text{where} \quad \mathbb{1}\{x>0\} = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases} \tag{1.5}$$

is the indicator function. Now, complete the following task.

Task 2: (5%)

Formulate the portfolio optimization problem to determine $\boldsymbol{p} \in \mathbb{R}^n$ with the following requirements:

- The objective is to minimize the downside risk associated with the new portfolio, i.e.,

$$\sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{p}^\top(\bar{\boldsymbol{r}} - \boldsymbol{R}_t) > 0\}$$

[3]Upside risk is the sum of excessive **actual return** compared to the expected one; downside risk is the sum of insufficient **actual return** compared to the expected one. Usually, upside risk is not a risk as you actually make more profit than expected; see `https://en.wikipedia.org/wiki/Upside_risk`.

- It is required that the expected return is above or equal a given threshold $\$R_{\sf d}$, i.e.,

$$\textstyle\sum_{i=1}^n p_i \bar{r}_i \geq R_{\sf d}.$$

- The sum of *net change in portfolio value* must be less than or equal to the budget $\$B$, i.e.,

$$\textstyle\sum_{i=1}^n p_i \leq B.$$

- The portfolio must be non-negative, i.e.,

$$p_i \geq 0,\ \ i = 1, ..., n.$$

You may assume that for any $t = 1, \ldots, T$, the quantity $\boldsymbol{p}^\top(\bar{\boldsymbol{r}} - \boldsymbol{R}_t)$ is always bounded by M. The formulated problem shall be a mixed-integer program.

As we have learnt from the class, the MIP problem can be very challenging to solve when T is large. As an alternative, we approximate the objective function (1.5) with the following *ridge function*:

$$\frac{1}{T}\sum_{t=1}^{T} \max\{\boldsymbol{p}^\top \bar{\boldsymbol{r}} - \boldsymbol{p}^\top \boldsymbol{R}_t, 0\} \tag{1.6}$$

where

$$\max\{x, 0\} = \begin{cases} x, & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases}$$

Now, complete the following task.

Task 3: (5%)

(a) Similar to Task 2, formulate a optimization problem to determine the portfolio $\boldsymbol{p} \in \mathbb{R}^n$ with the following requirements:
  - The objective is to minimize the ridge-function risk (1.6) associated with the portfolio.
  - It is required that the expected return is above or equal a given threshold as $\$R_{\sf d}$.
  - The sum of *net change in portfolio value* must be less than or equal to the budget $\$B$.
  - The new portfolio must be non-negative.

(b) Rewrite the above optimization problem as a standard form LP.

Computational

The optimization problems formulated in Task 2 & 3 are MIPs/LPs which **do not** admit a closed form solution. We next focus on solving these portfolio optimization problems numerically on computers and evaluate their performance subsequently.

Data Preprocessing. The required data – expected return $\bar{r}_i$, and covariance between stocks $\boldsymbol{\Sigma} = [\rho_{ij}]_{n\times n}$ – can be *approximated* from historical data as:

$$\bar{r}_i \approx \frac{1}{T}\sum_{t=1}^{T} R_i(t), \quad \rho_{ij} \approx \frac{1}{T}\sum_{t=1}^{T}\Big(R_i(t) - \frac{1}{T}\sum_{t'=1}^{T} R_i(t')\Big)\Big(R_j(t) - \frac{1}{T}\sum_{t'=1}^{T} R_j(t')\Big), \quad \forall\, i, j = 1, ..., n. \tag{1.7}$$

With a slight abuse of notations, we shall refer the approximated expected return and covariance by the same symbols $\bar{r}_i = \frac{1}{T}\sum_{t=1}^{T} R_i(t)$ and $\boldsymbol{\Sigma} = [\rho_{ij}]_{n\times n}$.

In the following tasks, we focus on a small set of $n = 20$ stocks (from `ftec_project_subgroup` i) as our first exploration:

Task 4: Warmup Exercise (5%)

(a) Inspect the dataset, e.g., by plotting the daily return over time of around 3-4 stocks of your choice.

Remark: The program template has provided the relevant helper codes for this task.

Portfolio Optimization. In the following tasks, you will implement the different portfolio optimization formulations that you have completed in Task 1–3. They will be implemented into Julia and their performances will be evaluated.

We first implement the closed form solution found in Task 1:

Task 5: Closed Form Solution with Simple Portfolio Optimization (7.5%)

(a) Implement the closed form solution found in Task 1 with the given training data. You may take the parameters

$$B = 20,\ R_{\sf d} = \max\{2, \textstyle\sum_{i=1}^{n} \bar{r}_i\},\ \lambda = 0.5.$$

Comment on the portfolio you found using the above parameters.

(b) Plot the portfolio across different stocks and compare the portfolio profile with (i) the expected returns $\bar{r}_i$, (ii) the variance of return for each stock $\hat{\rho}_{i,i}$. Comment on the trends you found using the above parameters. *Hint: You may need to scale up/down the expected returns and variances to make the plots more readable.*

(c) Calculate Task 1(b) and try several different values (at least 2) for the parameter λ. Comment on its effects on the optimal portfolio.

(d) Try different values for the desired return $R_{\sf d}$, budget B. Comment on their effects on the optimal portfolio.

Suppose the given portfolio is $w_i = 1,\ i = 1, ..., n$. We now implement the two formulations in Tasks 2–3, as well as the original mean-variance Markowitz portfolio optimization.

Task 6: Optimization-based Formulation (17.5%)

(a) Implement an SOCP program for minimizing the risk $\frac{1}{2}\boldsymbol{p}^\top \boldsymbol{\Sigma} \boldsymbol{p}$ as in (1.4); check out Lecture Note 19 to see how to write the SOCP program. Use the following parameters:

$$B = 20,\ R_{\sf d} = \max\{2, \textstyle\sum_{i=1}^{n} \bar{r}_i\}$$

You may use the solver `ECOS` in `JuMP` for the SOCP.

(b) Implement the mixed-integer program from Task 2 with the following parameters:

$$B = 20,\ M = 40,\ R_{\sf d} = \max\{2, \textstyle\sum_{i=1}^{n} \bar{r}_i\}$$

You may use the solver `GLPK` in `JuMP` for the MI-LP.

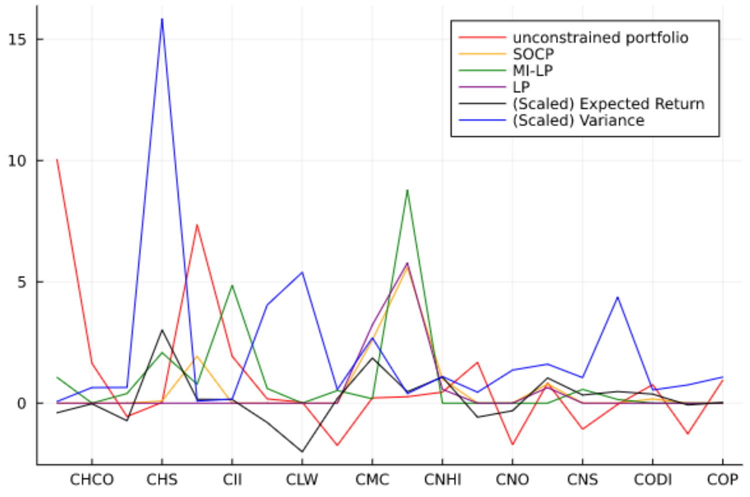
(c) Implement the LP from Task 3 with the following parameters:

$$B = 20,\;\; R_{\sf d} = \max\{2, \textstyle\sum_{i=1}^n \bar{r}_i\}$$

You may use the solver `GLPK` in `JuMP` for the LP.

(d) Plot and compare the portfolios found using the approaches from Task 1, 2, and 3. Comment on the differences between the solutions obtained.

Notice that the plot in part d) of Task 6 may look like:



which shows the portfolio optimized for each stock, compared with the expected return and variance/risk.

Lastly, we shall introduce several metrics for measuring the performance of a portfolio $\boldsymbol{p}$. Here, these performance metric shall be calculated from the test dataset with $T_{\sf test}$ days of record. We first formally define the downside risk in two different favors:

$$\text{Prob. of Downside Risk Violation} = \frac{1}{T_{\sf test}} \sum_{t=1}^{T_{\sf test}} \mathbb{1}\{\boldsymbol{p}^\top \bar{\boldsymbol{r}}_{\sf train} - \boldsymbol{p}^\top \boldsymbol{R}_t > 0\}$$

$$\text{Amt. of Downside Risk Violation} = \frac{1}{T_{\sf test}} \sum_{t=1}^{T_{\sf test}} \max\{0, \boldsymbol{p}^\top \bar{\boldsymbol{r}}_{\sf train} - \boldsymbol{p}^\top \boldsymbol{R}_t > 0\}$$

where $\bar{\boldsymbol{r}}_{\sf train}$ indicates the average risk vector from the *training dataset*, i.e., it was simply called $\bar{\boldsymbol{r}}$ in the previous tasks. Another commonly used metric is the *Sharpe ratio*[4] [Fabozzi et al., 2007]:

$$\text{Sharpe ratio} = \frac{\bar{\boldsymbol{r}}_{\sf test}^\top \boldsymbol{p}}{\sqrt{\boldsymbol{p}^\top \boldsymbol{\Sigma}_{\sf test} \boldsymbol{p}}}$$

notice that the expected return $\bar{\boldsymbol{r}}_{\sf test}$ and covariance $\boldsymbol{\Sigma}_{\sf test}$ shall be calculated using similar formula as (1.7) but with the *testing* data that are not seen during the portfolio optimization. Notice that if $\boldsymbol{p} = \boldsymbol{0}$, then we define Sharpe ratio $= 0$. For your convenience, we have prepared a function that are included with the preloaded file `reusablefunc.jl` and it can be called as `performance_testdata(path, p,T,r)`, with the inputs as

[4]Assuming that the risk free asset has zero return.

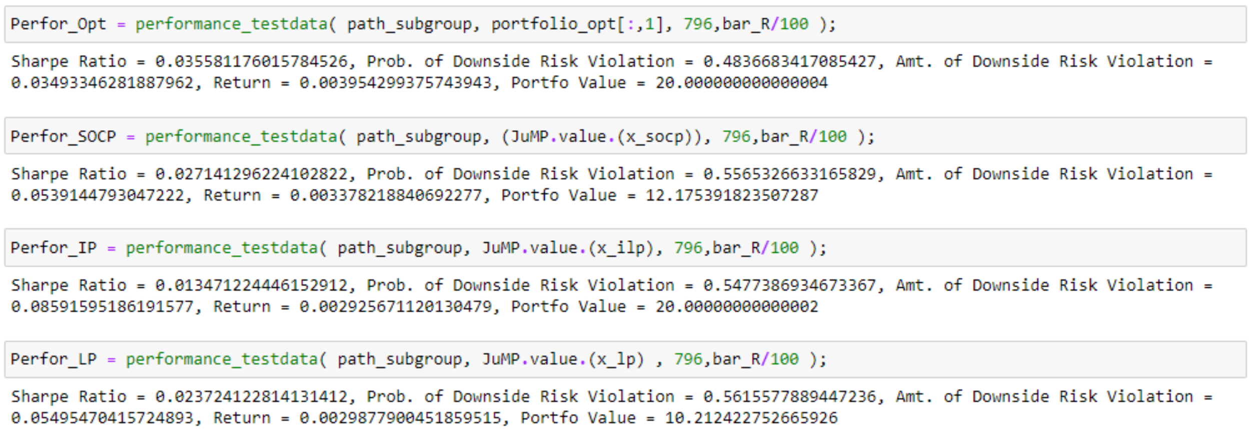
- `path` – the relative path in which the testing data are stored.
- `p` – the portfolio $\boldsymbol{p}$.
- `T` – the number of records.
- `r` – the average risk vector for evaluating downside risk violation.

The function will report (i) the Sharpe ratio of the portfolio $\boldsymbol{p}$, (ii) downside risk violation probability, (iii) the average downside risk violation, (iv) the expected return, and (v) the total portfolio value.

Task 7: Evaluate the Portfolio Performance on Testing Set (5%)

Compare the solutions found in Task 5 and 6 under the given parameters, (i) the Sharpe ratio of the portfolio $\boldsymbol{p}$, (ii) downside risk violation probability, (iii) the average downside risk violation, (iv) the expected return, and (v) the total portfolio value.

For example, you may observe the following output using the helper code provided:



```
Perfor_Opt = performance_testdata( path_subgroup, portfolio_opt[:,1], 796,bar_R/100 );

Sharpe Ratio = 0.035581176015784526, Prob. of Downside Risk Violation = 0.4836683417085427, Amt. of Downside Risk Violation =
0.03493346281887962, Return = 0.003954299375743943, Portfo Value = 20.000000000000004

Perfor_SOCP = performance_testdata( path_subgroup, (JuMP.value.(x_socp)), 796,bar_R/100 );

Sharpe Ratio = 0.027141296224102822, Prob. of Downside Risk Violation = 0.5565326633165829, Amt. of Downside Risk Violation =
0.0539144793047222, Return = 0.003378218840692277, Portfo Value = 12.175391823507287

Perfor_IP = performance_testdata( path_subgroup, JuMP.value.(x_ilp), 796,bar_R/100 );

Sharpe Ratio = 0.013471224446152912, Prob. of Downside Risk Violation = 0.5477386934673367, Amt. of Downside Risk Violation =
0.08591595186191577, Return = 0.002925671120130479, Portfo Value = 20.00000000000002

Perfor_LP = performance_testdata( path_subgroup, JuMP.value.(x_lp) , 796,bar_R/100 );

Sharpe Ratio = 0.023724122814131412, Prob. of Downside Risk Violation = 0.5615577889447236, Amt. of Downside Risk Violation =
0.05495470415724893, Return = 0.0029877900451859515, Portfo Value = 10.212422752665926
```

Note that the testing set contains data that have not been seen by the optimization problem. Don't be surprised if they display worse performance than you expected.

1.2 Competitive Tasks (30%)

The goal of this competitive task is to implement your own (approximate) solver to the portfolio optimization problem, without relying on JuMP and its optimizers such as GLPK, ECOS. To motivate, we observe that while optimization packages such as JuMP are very convenient to use, yet these solvers are limited by their high complexity for large-scale problems when $n, T \gg 1$.

To prepare for this task, we adopt two modifications to search for a portfolio with the *best performance* and *lowest cost*, as follows:

- In Task 3, you have begun with a portfolio optimization problem in the following form (before you transform the problem into a LP):

$$\begin{aligned} \min_{\boldsymbol{p}\in\mathbb{R}^n} \quad & \sum_{t=1}^{T} \max\{\boldsymbol{p}^\top \bar{\boldsymbol{r}} - \boldsymbol{p}^\top \boldsymbol{R}_{\rm t}, 0\} \\ \text{s.t.} \quad & f_1(\boldsymbol{p}) \le 0, \quad f_2(\boldsymbol{p}) \le 0, \\ & 0 \le p_i, \;\; i = 1, ..., n, \end{aligned}$$

where f_1, f_2 are some functions of the portfolio $\boldsymbol{p} \in \mathbb{R}^n$. A heuristic for solving the above problem is to consider the following approximation:

$$\begin{aligned} \min_{\boldsymbol{p}\in\mathbb{R}^n} \quad & \sum_{t=1}^{T} \max\{\boldsymbol{p}^\top \bar{\boldsymbol{r}} - \boldsymbol{p}^\top \boldsymbol{R}_t, 0\} + \gamma f_1(\boldsymbol{p}) + \upsilon f_2(\boldsymbol{p}) \\ \text{s.t.} \quad & 0 \le p_i \le 1, \;\; i = 1, ..., n, \end{aligned} \tag{1.8}$$

for some $\gamma, \upsilon > 0$. Note we have included an upper bound on each p_i.

For your information, this method is called the penalty method which adds constraints to the objective function by imposing a cost to infeasible points [cf. This idea is related to the Big-M simplex method]. For instance, since we desired that $f_1(\boldsymbol{p}) \le 0$ in the original formulation, the approximation shall impose a positive cost if $f_1(\boldsymbol{p}) > 0$ [recall that $\gamma > 0$]. Moreover, intuitively, increasing γ (resp. υ) ensures that the constraint $f_1(\boldsymbol{p}) \le 0$ (resp. $f_2(\boldsymbol{p}) \le 0$) is more strictly enforced.

- An issue with $\max\{x, 0\}$ function is that the latter is not differentiable at $x = 0$. As such, the second modification is to approximate the max function with the Huber function, parameterized by $\delta > 0$, as:

$$h_\delta(x) = \begin{cases} \dfrac{x^2}{2\delta}, & \text{if } 0 \le x \le \delta, \\ x - \frac{1}{2}\delta, & \text{if } x > \delta, \\ 0, & \text{if } x < 0, \end{cases} \tag{1.9}$$

and we replace the optimization problem (1.8) by

$$\begin{aligned} \min_{\boldsymbol{p}\in\mathbb{R}^n} \quad & \sum_{t=1}^{T} h_\delta(\boldsymbol{p}^\top \bar{\boldsymbol{r}} - \boldsymbol{p}^\top \boldsymbol{R}_t) + \gamma f_1(\boldsymbol{p}) + \upsilon f_2(\boldsymbol{p}) \\ \text{s.t.} \quad & 0 \le p_i \le 1, \;\; i = 1, ..., n, \end{aligned} \tag{1.10}$$

Finally, observe that (1.8) is now an optimization problem with a 'simple' constraint:

$$X = \{\boldsymbol{p} \in \mathbb{R}^n : 1 \ge p_i \ge 0, \;\; i = 1, ..., n\},$$

where X is a convex set, and the projection into this set can be easily calculated. In other words, with a differentiable and convex f_1, f_2, it is easy to apply methods such as projected gradient, conditional gradient methods to solve (1.8); see Appendix A. This is the goal of the main task in this section:

Task 8: Customized Solver for Portfolio Optimization (30%)

Using the complete dataset with the training data from $n = 699$ stocks stored in `/ftec_project_files/`, recorded over $T = 799$ days. Implement a projection-based iterative algorithm to tackle (1.8) with a choice of approximation (e.g., Huber function (1.9)) approximating the downside risk minimization problem in Task 3 (which approximated Task 2). Notice that the budget constraint B and expected return goal $R_{\sf d}$ are irrelevant in this task as the additive constants in the constraints shall have no effects on (1.8). As a recommendation, you can apply the projected gradient descent method using a constant step size.

While we have provided some suggested values in the helper code. You may need to tune the parameters υ, γ, δ, as well as the step size, to achieve the best result.

For the iterative algorithm applied, you are required to initialize by $\boldsymbol{p}^0 = \boldsymbol{0}$ and the algorithm has to run for at least 5000 iterations. You are encouraged to try more than one algorithm (which will be counted towards the 'innovation' section in the Report assessment).

Assessment You will receive a maximum of **9%** for implementing at least one numerical algorithm (e.g., projected gradient) **correctly**, together with (i) plotting the trajectory of the algorithm is show that the objective value in (1.8) is decreasing and providing comments on the algorithm(s) implemented, (ii) showing the derivations on why the implemented algorithm can be used to solve/tackle (1.8).

The remainder of your marks will be calculated according to the following formula:

$$\begin{aligned}
\mathsf{Score} &= \text{(Score on the Sharpe Ratio, see below)} \\
&+ 5\% \times \frac{\exp\Big(10 \cdot \max\{0.45, \text{Lowest Probability of Downside Risk Violation (training data)}\}\Big)}{\exp\Big(10 \cdot \max\{0.45, \text{Your Probability of Downside Risk Violation (training data)}\}\Big)} \\
&+ 3\% \times \frac{\max\{0.1, \text{Lowest Amount of Downside Risk Violation (training data)}\}}{\max\{0.1, \text{Your Amount of Downside Risk Violation (training data)}\}} \\
&+ 5\% \times \frac{\exp\Big(10 \cdot \max\{0.45, \text{Lowest Probability of Downside Risk Violation (test data)}\}\Big)}{\exp\Big(10 \cdot \max\{0.45, \text{Your Probability of Downside Risk Violation (test data)}\}\Big)} \\
&+ 3\% \times \frac{\max\{0.1, \text{Lowest Amount of Downside Risk Violation (test data)}\}}{\max\{0.1, \text{Your Amount of Downside Risk Violation (test data)}\}}.
\end{aligned} \tag{1.11}$$

where the score on Sharpe Ratio will be calculated as follows:

$$\text{if Your Sharpe Ratio (testing data)} > 0, \text{ then } \; 5\% \times \frac{\min\{0.01, \text{Your Sharpe Ratio (testing data)}\}}{\min\{0.01, \text{Highest Sharpe Ratio (testing data)}\}}$$

$$\text{otherwise, it is } \; 5\% \times \frac{\text{Your Sharpe Ratio (testing data)}\}}{\text{Most Negative Sharpe Ratio (testing data) in class}}$$

Note that this part of the score can be **negative** if the Sharpe ratio of your portfolio is negative. For example, the lowest amt. of downside risk violation (training data) are calculated as the lowest one among the class of FTEC2101[5].

If you have tried more than one algorithm and/or more than one set of parameters, you can select **only one set of portfolio** for the competition. Please indicate which set of portfolio is selected in your report and include that in the submission of your program files. Moreover, please observe the following rule:

- You are **not allowed** to directly optimize on the testing set data. In other words, your iterative algorithm should not 'touch' any data file with the suffix `_test.csv` as you are not supposed to see the 'future' when investing. Your score in (1.11) will be set to zero if we detect such 'cheating' behavior. However, you can use the function `total_performance` to evaluate your portfolio as many times as you like.

[5]These constants will be calculated separately among the ESTR2520 students.

- Your selected algorithm for the competition must be **deterministic**. In other words, you may not use stochastic algorithm such as stochastic gradient descent (SGD) for the competition. That being said, you are still encouraged to try such algorithms as an additional task which may be counted towards the 'innovation' section.

If you have questions about the rules, please do not hesitate to consult the instructor at `htwai@cuhk.edu.hk` or the TA or ask questions on Piazza.

1.3 Report (20%)

You are required to compile a project report with answers to the questions posed in Task 1 to Task 8. You may structure the report according to the order of the tasks, for instance:

1. **Background and Introduction** — In this section, you can briefly introduce the problem, e.g., explaining the goal of portfolio design, discussing the role of optimization methods in portfolio design.
2. **Model and Theory** — In this section, you can discuss how the portfolio design problem is modeled as optimization problems. More specifically,
   – You shall begin by discussing the formulation (1.1) and then answer the questions in **Task 1**.
   – Next, you can describe the optimization models with downside risk and then answer **Tasks 2 & 3**.
3. **Experiments** — In this section, you describe the experiments conducted to test your formulation, i.e.,
   – You shall first describe the dataset by answering **Task 4**. In addition, it is helpful to describe a few meta-data regarding the dataset, e.g., from when to when the stock price data is collected.
   – Then, you can describe the experiments for each of the 3 formulations, by answering **Tasks 5 & 6**.
   – Finally, you can compare the formulations by answering **Task 7**.
4. **Competitive Task** — In this section, you describe the custom solver you built to solve the large-scale portfolio design problem, i.e.,
   – You shall first describe the approximation technique as laid out in the discussion of (1.8), (1.9).
   – Then, you shall describe the iterative algorithm you have derived in **Task 8**.
   – Apply the iterative algorithm on the complete training dataset and show the objective value vs. iteration number. Discuss whether the algorithm converges and report on the performance of the designed portfolio (e.g., the Sharpe ratio, etc..).
5. **Conclusions** — In this section, you shall summarize the findings in the project, and discuss various aspects that can be improved with the formulation, etc..

Throughout the report, please feel free to write your answer which involves equations (e.g., Task 1-3) on a paper and scan it to your Word/PDF report as a figure. For Task 4 to 8, please include all the plots and comments as requested. For Task 8, please indicate the required metrics in the scoring formula (1.11) of **the selected solution**.

The program code has to be submitted separately. However, you are welcomed to use excerpts from the program codes if you find that it is helpful for explaining some of the concepts.

Lastly, you are welcomed to use online resources when preparing the project. However, you must give **proper references** for every sources that are not your original creation.

Assessment Here is a breakdown of the assessment metric for the report writing component.

- (10%) *Report Writing*: A project report shall be readable to a person with knowledge in optimization (e.g., your classmates in FTEC2101/ESTR2520). Make sure that your report is written with clarity, and more importantly, using your own language!
- (10%) *Innovation*: You can get innovation marks if you include extra experiments, presentations, etc.. that are relevant to the project (with sufficient explanations); see Appendix A for some recommendations.

1.4 Submission

This is an individual project. While discussions regarding *how* to solve the problems is encouraged, students should answer the problems on their own (just like your HWs). The deadline of submission is May 15, 2023, 23:59 (HK time). Please submit a single zip file with the following content:

- Your Project Report in PDF format.
- Your Program Codes [either in Jupyter notebook (`.ipynb`), or Julia code (`.jl`)].

In addition, all project reports shall be submitted to VeriGuide for plagiarism check. (The submission to VeriGuide needs not follow the strict deadline of the main submission, e.g., it can be done within a reasonable amount of time like on or before May 16.)

On the Use of Generative AI Tools

The teaching team of FTEC2101/ESTR2520 is aware of the rapid advances of generative AI tools such as ChatGPT from OpenAI, Sydney from Bing, etc. Below we state our policy on their uses in this project assessment.

In short, you are allowed to use generative AI tools to **assist you** in this project, provided that you must give **explicit acknowledgement** to the use of such tools, e.g., you may include a sentence like:

> *The following section has been completed with the aid of ChatGPT.*

Failure to do so will constitute act of academic dishonesty and may result in **failure of the course** and/or other penalties; see `https://www.cuhk.edu.hk/policy/academichonesty/`. In other words, we are roughly following **Approach 3** as listed in the University's Guideline on the matter: `https://www.aqs.cuhk.edu.hk/documents/A-guide-for-students_use-of-AI-tools.pdf`.

Below we list a number of examples of the do's and don'ts for the FTEC2101/ESTR2520 project:

- **DO's**: You may use AI tools such as ChatGPT for polishing your writeups, e.g., to correct grammatical mistakes, typos, or summarizing long/complicated paragraphs, etc. The results are usually quite robust especially for improving the writings from less experienced writers. Of course, you are responsible for the integrity of the edited writing, e.g., check if the AI tools have distorted the meaning of your original writeups or not.

- **DON'Ts**: You should not ask AI tools such as ChatGPT to formulate optimization problems, solve mathematical questions, or write the programming codes for you. Not only this will spoil the purpose of learning, AI tools do a notoriously bad job for tasks involving *facts* and *mathematical/logical reasoning.* Worst still, they tend to produce solutions that *sound* legit but are completely wrong.
- **DON'Ts**: You should not ask AI tools such as ChatGPT to write the entire report for you. Likewise, AI tools are notoriously bad at creating (technical) content. They tend to produce writings that *sound* legit but are completely illogical.

The practical use of generative AI tools is all new to the most of us. We believe that when properly used, they can be helpful in improving students' overall learning experience. In fact, you are even encouraged to try them out at your leisure time. (P.S. It is important to remind ourselves that without advanced optimization methods, these almost-magical generative AI tools would not even exist!)

Nevertheless, we must emphasize again that you have to provide **explicit acknowledgement** in your submission if you have used any generative AI tools to assist you in the project.

A Additional Information

Several tips for implementing the convex problems which have been included with the the template program. However, please be reminded that it is not necessary to follow all the tips therein.

Suggestions for Extensions — The below are only suggestions for the extensions. You are more than welcomed to try out new ideas in your project.

- *Formulation Aspect* – For task 8, another model to approximate the max function is to use the sigmoid function, parameterized by $\alpha > 0, \beta$, as:

$$\frac{1}{T}\sum_{i=t}^{T} \sigma_{\mathrm{t},\alpha,\beta}(\boldsymbol{p}) \quad \text{where} \quad \sigma_{\mathrm{t},\alpha,\beta}(\boldsymbol{p}) = \frac{1}{1+\exp\{-\alpha\, \boldsymbol{p}^\top(\bar{\boldsymbol{r}} - \boldsymbol{R}_t) + \beta\}}. \tag{1.12}$$

  which yields a better approximation than $\max\{x, 0\}$ to the desired 0-1 indicator function.

- *Algorithm Aspect* – For Task 8, with the constraint structure given, the recommended algorithms are *projected gradient descent (PGD) method* and *conditional gradient (CG) method*, which are described as follows. For solving a general optimization:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x}) \ \text{ s.t. } \ l_i \le x_i \le u_i,\ i = 1, ..., n. \tag{1.13}$$

  Let $X = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ l_i \le x_i \le u_i,\ i = 1, ..., n\}$, these algorithms can be described as

```
Input: $x^{(0)} \in \mathbb{R}^n$ with $l_i \le x_i^{(0)} \le u_i$ for all $i$, constant step size $\gamma > 0$, max. iteration number $K_{max}$.
For $k = 0, ..., K_{max}$
    $x^{(k+1)} = \mathrm{Proj}_X\{x^{(k)} - \gamma \nabla F(x^{(k)})\}$
End For
```

PGD Method

```
Input: $x^{(0)} \in \mathbb{R}^n$ with $l_i \le x_i^{(0)} \le u_i$ for all $i$, max. iteration number $K_{max}$.
For $k = 0, ..., K_{max}$
    $a^{(k)} = \arg\min_{a \in X} \ a^\top \nabla F(x^{(k)})$
    $x^{(k+1)} = (1 - \frac{2}{k+1})x^{(k)} + \frac{2}{k+1} a^{(k)}$
End For
```

CG Method

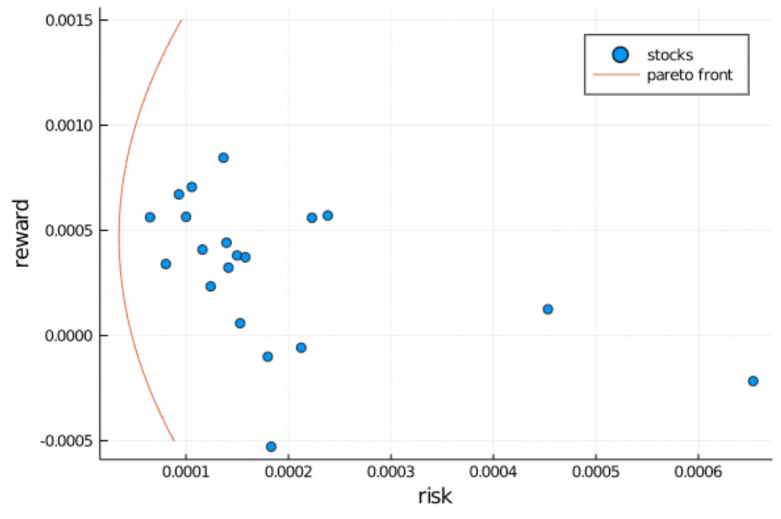
Moreover, the projection/CG operators can be easily computed: for any $\boldsymbol{g} \in \mathbb{R}^n$, we have

$$\boldsymbol{g}^{PG} = \text{Proj}_X(\boldsymbol{g}), \quad \text{where} \quad [\boldsymbol{g}^{PG}]_i = \begin{cases} g_i, & \text{if } g_i \in [l_i, u_i], \\ l_i, & \text{if } g_i < l_i, \\ u_i, & \text{if } g_i > u_i. \end{cases}$$

$$\boldsymbol{g}^{CG} = \arg\min_{\boldsymbol{a} \in X} \ \boldsymbol{a}^\top \boldsymbol{g}, \quad \text{where} \quad [\boldsymbol{g}^{CG}]_i = \begin{cases} u_i, & \text{if } g_i < 0, \\ l_i, & \text{if } g_i \geq 0. \end{cases}$$

It should not be very hard to program the above into a Julia function. Nevertheless, you are more than welcomed to explore the use of other iterative algorithms for solving the optimization at hand.

- *Presentation Aspect* – The comparison of the (unconstrained) optimal portfolio against the return/risk profile as done in Task 5 (b) is related to the *Efficient/Pareto Frontier* of an investment portfolio. To investigate the latter, you can plot the risk of the optimal portfolio (in Task 1) as a function of the desired reward, e.g., as follows



Importantly, this efficient frontier is found to be always 'enclosing' the reward/risk of each stock covered in the portfolio; see `https://en.wikipedia.org/wiki/Efficient_frontier` for further reference.

References

F. J. Fabozzi, P. N. Kolm, D. A. Pachamanova, and S. M. Focardi. *Robust portfolio optimization and management*. John Wiley & Sons, 2007.