Homework 3

Yuting Liang

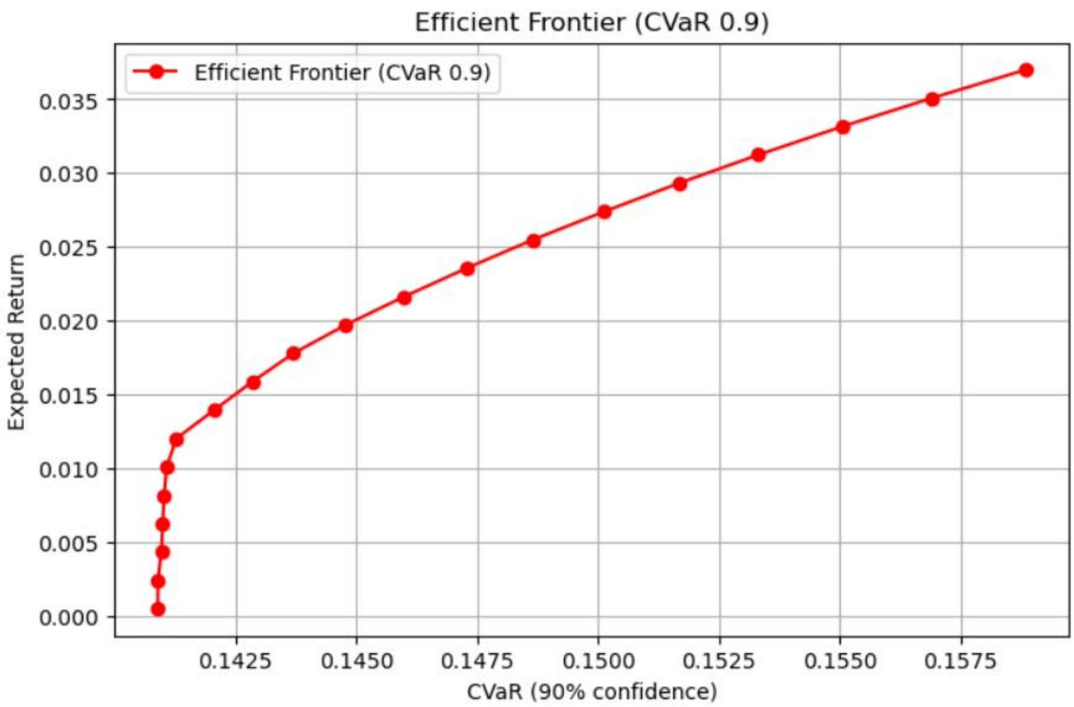
First, after installing PSG, we need to ensure that the Python version is 3.11. Then, we run the PSG interface in Python, allowing us to use PSG within the Python environment.

1、Changing CVaR deviation

We replace cvar_dev (0.99, matrix_scenarios) with cvar_dev(0.90, matrix_scenarios). We can find that the optimized portfolio has an expected return of 1.165% with a 90% CVaR of 13.91%, indicating controlled downside risk. The 99% CVaR is 23.16%, with a maximum potential loss of 26.04%, while the portfolio's volatility is 5.25%.

2、Finding the efficient frontier by changing the expected return

First, we determine the upper and lower limits of the dataset's returns and keep only the positive returns. Then, we randomly generate 20 expected returns within this range and input them into PSG's objective function to compute the corresponding CVaR values. However, when plotting, I couldn't directly store the results as a string, so I manually extracted the results and wrote them into an array before plotting.



3、Maximizing the objective function

In this step, we only need to replace the objective function and constraints in the previous code with those provided in the problem statement. To ensure a fair comparison between the two objective functions, we no longer randomly select 20 expected returns. Instead, we use the same 20 expected returns generated in the second step and input them into PSG for computation. The resulting efficient frontier is as follows, demonstrating that the two methods for calculating CVaR are equivalent.