FIN 558 Project: Evaluating the Optimal Portfolio's Constituent and Weight Using S&P500 Index Constituent

Group Member: Xiaolei Chang, Yijun Luo, Xinyu Ye

1 Introduction

This project studies S&P500 constituent data, S&P500 index data, and 30-day treasury bill data, which are downloaded from CRSP in the WRDS. The objective of this project is to find out the optimal portfolio's constituent weights among 1000 randomly-generated portfolios. In order to make calculations more convenient and efficient, this research constructs custom functions to calculate the Sharpe ratio of each random portfolio. The result of this project has been exported to a separate MS Excel to display the optimal portfolio's constituent weights.

2 Description of Data

3 datasets downloaded from CRSP in the WRDS are applied to this study: S&P500 constituent data, S&P500 index data, and 30-day treasury bill data. S&P500 constituent data are the monthly returns of S&P500 constituent during 2010 and 2022. S&P500 index data are the monthly returns of S&P500 index during 2010 and 2022. 30-day treasury bill data are the monthly returns of treasury bill during 2010 and 2022, which serve as risk free rate of return in this study.

3 Methodology

(1) Data Preparation

This study first imports the S&P500 constituent data and 30-day treasury bill data to python, then processes the data (drops the row missing and duplicate numbers) and convert data into pivot table. In addition, we use descriptive statistics of numpy array to calculate average monthly returns and covariance matrix of S&P500 constituent.

(2) Custom Functions

This study defines 3 functions: F_PortRtn(r_i , w_i), F_PortStd(cov, w_i), and F_Sharpe(r_p, r_f, s_p). F_PortRtn(r_i , w_i) inputs average returns of stocks and their weights in a portfolio and outputs expected return of a portfolio. F_PortStd(cov, w_i) inputs covariance matrix of stock returns and their weights in a portfolio and outputs the standard deviation of a portfolio. F_Sharpe(r_p, r_f, s_p) inputs expected return, standard deviation, and risk free rate of return and outputs the Sharpe ratio of a portfolio.

(3) Simulation

This study set seed to generate 1,000 sets of N=435 random numbers following uniform distribution between 0 and 1 and compute the random weights of stocks. Then we use the custom functions defined before to compute Sharpe ratio of each simulated portfolio and use descriptive statistics of numpy array to locate the maximum Sharpe ratio portfolio. Finally, we combine the names, permno, and weights of the optimal portfolio's constituent by using panda data frame calling operation.

4 Result

After processing the S&P500 constituent monthly returns data (drop the row missing and duplicate numbers), there remains N=435 constituent in the datasets. The max Sharpe ratio of 1000 randomly-generated portfolios is 0.305 under the seed 653282893. The result of this study, which is the optimal portfolio's constituent weights with max Sharpe ratio, are exported to a MS excel spreadsheet named 'Optimal portfolio's constituent'.

5 Conclusion

The Sharpe ratio of optimal portfolio is 0.305, indicating that for 1 unit of risk, the return is 0.305 times higher than risk-free asset. And this means that the average growth rate of the optimal portfolio's net worth from 2010 to 2022 exceeded the risk-free rate, suggesting that investors should invest in this portfolio during this observed period, which can generate the highest profits than any other portfolio.

However, we should also consider the accuracy of simulation in practice. Since the simulation progress requires variables to obey uniform probability distribution while the distribution of the actual data may not necessarily fit this distribution law exactly, we may get the wrong conclusion that fund investment of this optimal portfolio is better than risk free investment and fund investment of other portfolios with different weight of constituent.

To solve this, we can obtain as much and as accurate initial data as possible when conducting market research, so that when the data are initially processed, a more accurately fitted probability distribution can be obtained, thus improving the efficiency of simulation.