Proceedings of the 3rd International Conference on Business and Policy Studies
DOI: 10.54254/2754-1169/78/20241730

Markowitz Model and Python Applied in Financial Risk Management

Zekun He[1,a,*]

[1]Henan University, Kaifeng, Henan, 475000, China
a. hzk19138038888@163.com
**corresponding author*

Abstract: With the gradual development of financial technology, the improvement of the financial system, and the advent of the big data era, new technologies have been applied in the field of finance. This article aims to explore the integration of the Markowitz model and Python to achieve portfolio optimization and asset allocation. The Markowitz model is a classic portfolio theory that constructs an efficient portfolio by balancing different risks and expected returns. Python, as a powerful programming language and data analysis tool, provides many libraries and methods that can help apply the Markowitz model in practice. In this article, one stock was selected from each of the 13 different popular industries with the help of Python tools for portfolio analysis. The optimal portfolio with the maximum Sharpe ratio and the optimal portfolio with the minimum variance were empirically obtained, and their expected returns, standard deviations, and Sharpe ratios were compared. Besides, the efficient frontier of the asset portfolio was also presented. Through empirical analysis, the author further illustrated the importance of the Markowitz portfolio theory in financial risk management.

Keywords: Markowitz model, Python, Sharpe ratio, Portfolio optimization, Efficient frontier

1. Introduction

The Markowitz model, also known as the mean-variance model or portfolio theory, was proposed by American economist Harry Markowitz in 1952 and is one of the foundations of the modern portfolio theory [1]. Key concepts in the Markowitz model include expected returns, risk, covariance matrix, and efficient frontier. Investors can choose the optimal portfolio on the efficient frontier based on their risk preferences. In the Markowitz model, there are two parts: the mean-variance model and the efficient frontier. The variance matrix is used to represent the covariance between different assets in the portfolio. The efficient frontier is a set of optimal portfolios that offer the highest expected return for a given level of risk or the lowest risk for a given level of expected return. Investors can choose the best investment portfolio on the efficient frontier based on their risk preferences. Against this background, this article first introduced the basic principles and core concepts of the Markowitz model. Then, commonly used data processing and analysis libraries in Python, such as NumPy and Pandas, were presented for processing and preprocessing historical data of investment portfolios. Next, the author studied the use of Pandas in Python_ Datareader and Matplotlib libraries to achieve visualization of the Markowitz model. Finally, the combined application of the Markowitz model and Python was demonstrated through

DOI: 10.54254/2754-1169/78/20241730

practical case studies. This article proved the effectiveness of combining the Markowitz model with Python in portfolio optimization and asset allocation. Through Python's data processing and analysis capabilities, investors can more scientifically allocate assets and achieve better investment results.

2. The Markowitz Model

In the Markowitz model, it is assumed that a portfolio consists of n assets, where ω_i represents the weight of asset i, and r_i represents the average return of asset i. The Markowitz portfolio theory defines portfolio risk and return using variance and mean, expressed as:

$$\text{Risk level: } \sigma_p{}^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\omega_i\omega_j \mathrm{cov}(r_i, r_j) = \omega^T \mathrm{cov}(r_i, r_j)\omega \tag{1}$$

$$\text{Average return: } E(r_i)=\frac{\sum_{i=1}^{m} r_i}{m} \tag{2}$$

$$\text{Total return: } \rho = \sum_{i=1}^{n}\omega_i\, r_i = r^T\omega \tag{3}$$

The covariance between asset i and asset j, denoted asσ_{ij}, measures the degree of correlation between the returns of the two assets. The correlation coefficient between asset i and asset j, denoted as ρ_{ij}, can be used to compare the magnitude of the correlation between two assets. σ_i and σ_j represent the standard deviations of asset i and asset j, respectively.

According to Formula (1), it can be seen that the risk of a portfolio mainly depends on the investment weights of each asset, the correlation coefficients between different securities, and the standard deviations of each asset. Therefore, assets with lower correlation coefficients are selected to reduce the non-systematic risk in this investment.

Selecting a portfolio involves a large amount of data collection and processing. This paper utilizes Python language techniques to apply the Markowitz model in the stock market. Python language can effectively solve the difficulties in calculating covariance and correlation coefficients. Moreover, Python language has comprehensive packages for accessing financial data from databases, which is convenient for operation.

In the Markowitz model, the correlation coefficients between different assets in the portfolio are controlled to be low, while trade-offs are made between portfolio variance and expected return, corresponding to portfolio risk and return. Under this controlled variable setting, three situations are encountered:

Minimum Variance Portfolio: This portfolio aims to minimize the portfolio variance while achieving a certain level of expected return. It represents the portfolio with the lowest risk among all possible portfolios.

$$\min \omega^T \mathrm{cov}(r_i, r_j)\omega \tag{4}$$

$$\text{S.t. } \sum_i \omega_i=1 \tag{5}$$

$$r^T\omega \geq \rho \tag{6}$$

$$\omega_1, \omega_{2,\cdots,}\omega_n \geq 0 \tag{7}$$

DOI: 10.54254/2754-1169/78/20241730

Tangency Portfolio: This portfolio is located on the efficient frontier and represents the optimal portfolio that maximizes the Sharpe ratio. The Sharpe ratio measures the excess return per unit of risk and helps investors evaluate the risk-adjusted performance of a portfolio.

$$\max \mathrm{r}^{\mathrm{T}}\omega \tag{8}$$

$$\mathrm{S.t.}\ \sum_{i} \omega_i = 1 \tag{9}$$

$$\omega^{\mathrm{T}}\mathrm{cov}(\mathrm{r_i}, \mathrm{r_j})\omega \leq \beta \tag{10}$$

$$\omega_1, \omega_{2,\dots,}\omega_n \geq 0 \tag{11}$$

Maximum Return Portfolio: This portfolio aims to maximize the expected return while accepting a certain level of risk. It represents the portfolio with the highest return among all possible portfolios.

$$\min -\lambda \mathrm{r}^{\mathrm{T}}\omega + \omega^{\mathrm{T}}\mathrm{cov}(\mathrm{r_i}, \mathrm{r_j})\omega \tag{12}$$

$$\mathrm{S.t.}\ \sum_{i} \omega_i = 1 \tag{13}$$

$$\omega_1, \omega_{2,\dots,}\omega_n \geq 0 \tag{14}$$

This article parameterizes the ratio of risk and reward and uses the Sharpe ratio to select the most cost-effective portfolio.

3. Methods

In this study, 20 stocks were selected from 20 different industries in the stock market that were popular and had good performance in the past two years [2]. Then, the selection was narrowed down to 13 representative stocks based on their historical data from the first half of the year. These 13 stocks formed a portfolio. Two optimization methods, maximizing the Sharpe ratio and minimizing the variance, were used to find the optimal portfolio weights. The expected returns, expected volatility, and Sharpe ratio for these two optimal portfolios were calculated. Finally, the efficient frontier of the portfolio was visualized.

3.1. Importing the Necessary Database

First, the author created a Python file and imported the database. The code below shows how to import the database:

```
import pandas as pd
import numpy as np
import statsmodels, api as sm
import scipy.stats as scs
import matplotlib.pyplot as plt
mport pandas_datareader as pdrimport matplotlib.pyplot as plt
stocks = ['603918.SS', '600030.SS', '000002.SZ', '601012.SS', '601598.SS', '601949.SS',
'002415.SZ', '002508.SZ', '601398.SS', '601919.SS', '000725.SZ', '300394.SZ']
start_date = '2021-01-01'
end_date = '2022-12-31'
```

```
data = pdr.get_data_yahoo(stocks, start=start_date, end=end_date)['Close']
plt.figure(figsize=(14,7))for c in data.columns.values:
plt.plot(data.index, data[c], lw=3, alpha=0.8,label=c)
plt.legend(loc='upper left', fontsize=12)
plt.ylabel('price in $')
plt.show()
```

Then, 12 popular stocks that had good performance and obtained their historical data were selected from January 1, 2021 to December 31, 2021, which consisted of 250 trading days. Python functions were used to calculate the annualized returns of each stock and the covariance matrix of stock returns, as shown in Table 1 and Table 2 [3].

Table 1: Annualized returns (%) of 13 stocks.

Code	Annualized returns
603918.SS	-13.79%
600038.SS	52.59%
000002.SZ	-46.41%
601012.SS	4.32%
601598.SS	35.14%
601949.SS	5.24%
002415.SZ	25.26%
002508.SZ	4.17%
601398.SS	10.47%
601919.SS	44.64%
000725.SZ	-24.16%
300394.SZ	46.28%

Table 2: Covariance matrix of 13 stocks.

0.08172079	0.03700368	-0.0053915	0.02198259	0.02472553	0.01715316	0.02312821	-0.00049342	0.04699158	0.05204474	-0.00579737	0.02323011
0.03700368	0.13848706	-0.03616053	0.03055684	0.05567116	0.02793684	0.04277474	0.01221526	0.05986884	0.05748337	-0.01179789	0.04138526
-0.00539158	-0.03616053	0.08061547	-0.00602505	-0.01320737	-0.00465684	-0.01292716	-0.00218737	-0.02326758	-0.01051821	-0.00206911	-0.003285
0.02198259	0.03055684	-0.00602505	0.02148947	0.01537105	0.01263079	0.01101842	0.00131658	0.02918105	0.02094526	-0.00433716	0.01312395
0.02472553	0.05567116	-0.01320737	0.01537105	0.05867895	0.01995	0.02943474	0.00375921	0.05278921	0.04579474	-0.00856474	0.02704737
0.01715316	0.02793684	-0.00465684	0.01263079	0.01995	0.02749947	0.01363658	0.00369105	0.02440579	0.02111316	-0.00420021	0.01194763
0.02312821	0.04277474	-0.01292716	0.01101842	0.02943474	0.01363658	0.07010368	0.00557237	0.04653053	0.04000816	-0.00561921	0.01932079
-0.00049342	0.01221526	-0.00218737	0.00131658	0.00375921	0.00369105	0.00557237	0.01685474	0.00326379	0.00149316	-0.00017463	0.00043232
0.04699158	0.05986884	-0.02326758	0.02918105	0.05278921	0.02440579	0.04653053	0.00326379	0.09481768	0.05124711	-0.01242921	0.03173474

Table 2: (continued).

0.0520 4474	0.0574 8337	-0.010 51821	0.0209 4526	0.0457 9474	0.0211 1316	0.0400 0816	0.0014 9316	0.0512 4711	0.0968 9158	-0.012 61737	0.0322 6263
-0.005 79737	-0.011 79789	-0.002 06911	-0.004 33716	-0.008 56474	-0.004 20021	-0.005 61921	-0.000 17463	-0.012 42921	-0.012 61737	0.0554 2147	-0.002 06595
0.0232 3011	0.0413 8526	-0.003 285	0.0131 2395	0.0270 4737	0.0119 4763	0.0193 2079	0.0004 3232	0.0317 3474	0.0322 6263	-0.002 06595	0.0577 6505

3.2. Monte Carlo Simulation for Portfolio Optimization

In the case of a single investment portfolio, returns are highly unstable. Therefore, more information about different asset allocation ratios and their corresponding returns is needed to help find the optimal investment portfolio and efficient frontier under different conditions.

First, a Monte Carlo simulation was conducted to generate 20,000 portfolio models and the mean and covariance of each portfolio were recorded. Since short-selling is not allowed in the Chinese stock market, the equity of each asset was constrained to be greater than or equal to 0, and the sum of equity weights to be 1. The `stat` function in Python was used to collect the data, and the `pandas_datareader` and `matplotlib` libraries were applied for data visualization [4].

Next, return to the original intention of the mean-variance theory based on the defined investment risk and return. The theory aims to obtain a set of weights that optimally balance the risk and return of an investment portfolio [5].

4. Result and Analysis

The mean and covariance of each simulated portfolio were vectorized, and they were used to allocate assets to the simulated portfolios. Then, these portfolio points were plotted on a graph with risk on the x-axis and return on the y-axis, as shown in Figure 1.

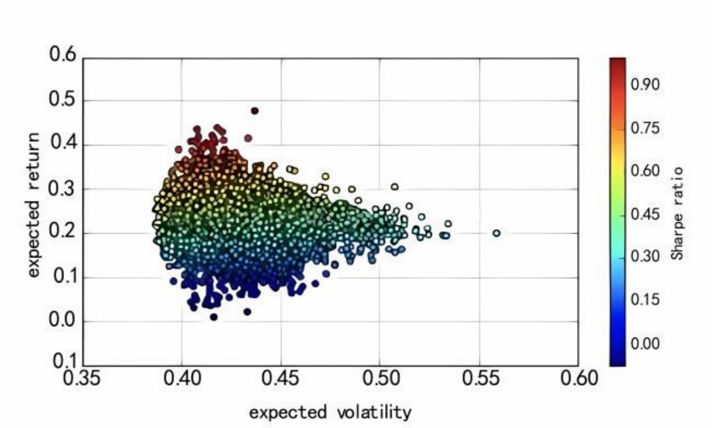

Figure 1: Visualization of all simulated investment portfolios.

Since each randomly simulated asset allocation has a different Sharpe ratio, different colors can be used to represent the Sharpe ratio values, making it easier to observe the portfolios. However, the resulting graph is not the most intuitive representation, so further processing is needed to mark the position of the efficient frontier, as shown in Figure 2.

DOI: 10.54254/2754-1169/78/20241730

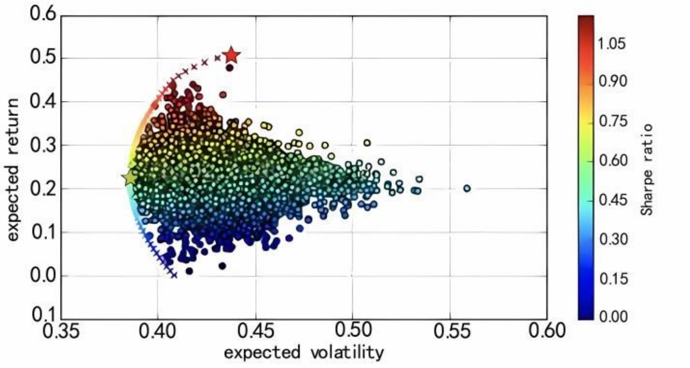

Figure 2: Visualization of the efficient frontier.

In Figure 2, the yellow star represents the minimum variance portfolio, and the red star represents the portfolio with the maximum Sharpe ratio (i.e., the risk-return balance point). The efficient frontier is divided into upper and lower parts by the minimum variance portfolio. The upper part of the efficient frontier represents the optimal portfolios. From the graph, it can be seen that all portfolios have positive Sharpe ratios, and the portfolios on the efficient frontier achieve the goal of minimizing investment risk at the same level of return and maximizing investment returns at the same level of risk [6]. Similarly, the portfolios on the efficient frontier provide options for investors with different risk and return preferences. Additionally, investors can define functions in Python to constrain the asset weights in the portfolio based on their own needs and preferences.

5. Conclusion

The comprehensive and meticulous application of statistical methods by Markowitz to analyze the optimal asset structure and the selection of such structure has resolved the issue of asset portfolio selection. The paper digitizes financial models using the Python language, allowing for the complete visualization of the entire model process and providing a more convenient premise for future model updates and the addition of more conditions. The utilization of the computational advantages of handling and analyzing large amounts of data by computers in the establishment of financial models reflects the advancement of investment theory from qualitative to scientific quantitative analysis, thus offering broader development prospects for asset pricing theory and providing new perspectives for other financial processing methods. Additionally, with the rapid development of artificial intelligence, intelligent financial services will also be applied to various aspects of the financial industry in the future, driving the reform and development of the financial sector. The Markowitz model can be fully presented in the form of Python and can be gradually improved in the future, integrating it with the rapidly developing artificial intelligence of the era and aligning with the theme of future era development.

References

[1] *Markowitz, H. (1952). Portfolio Selection. The Journal of Finance, 7(1), 77-91.*

[2] *Hilpisch, Y. (2018). Python for Finance: Mastering Data-Driven Finance. O'Reilly Media.*

[3] *VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media.*

[4] *Fabozzi, F.J., Gupta, F. and Markowitz, H.M. (2002). The Legacy of Modern Portfolio Theory. The Journal of lnvesting, 11(3), 7-22.*

[5] *Zhang, W. (2023). Research and application based on Python data visualization. Computer Programming Skills and Maintenance, (11), 3-5+12. doi:10.16184/j.cnki.comprg. 2023.11.038.*

[6] *Bell, S. (2016). Quantitative Finance For Dummies. John Wiley & Sons.*