To: Professor Sury

From: Pranav Cheruku, Benjamin Kanarick, Nevin Arimilli

Re: Identifying Efficient Portfolios and the Optimal Risky Portfolio Using MV Optimization

Due: 08/08/2022

Introduction:

The goal of this assignment is to take price data for FB, GOOG, and JPM and perform analysis of varying combinations of these equities with respect to the Modern Portfolio Theory by Harry Markowitz. We have written a Python code that queries the user for their current portfolio weights of this 3-asset portfolio, and from there we have computed the optimal risky portfolio, the efficient frontier, and the CAL given the risk-free rate as well. At the end of this assignment, we will have presented a graph consisting of the following:

- The user's current 3-asset portfolio
- The efficient frontier of portfolios for this set of 3 assets
- Suggested portfolios for the user to invest in given their preference to risk / return
- The CAL, which was calculated using a given risk-free rate
- The optimal risky portfolio, also known as the tangency portfolio
    1. *This is the portfolio that lies on the intersection of the EF and the CAL*

We will also present a table with the following information with respect to the user's initial portfolio, EF risky portfolio's 1 and 2, as well as Optimal portfolio's 1 and 2:

- Expected Return
- Standard Deviation
- Sharpe Ratio
- Weights of FB, GOOG, JPM, and the Risk-Free Rate

Methods and Procedures:

As for methods and procedures to reach a final output, a brief flow of logic of how the entire code works is as follows:

- Import the stock prices for the 3 assets of FB, GOOG, JPM
- Convert each stock in to a percent change value and calculate average return, variance, and standard deviation for each
- Create a covariance matrix given the variances from the previous step
- Query the user for their current weight allocation for each asset in the portfolio

- Use matrix notation to find the weights, expected return, variance, and standard deviation of the user's portfolio, global minimum variance portfolio, maximum return portfolio, and the optimal risk portfolio
- Use gradation of 1% to find the set of portfolios that lies between the GMV and max return portfolio (this is the efficient frontier)
- Find the slope of the CML (the Sharpe ratio of the ORP) given the risk-free rate, standard deviation of a given portfolio on the CML, and ORP mrp
- Plot :
    - ORP
    - EF
    - CML
    - User's portfolio
    - Two suggested portfolios on the EF given the closest risk and return to the user's current portfolio
    - Two Optimal Portfolio's on the CAL given the ORP
- Then we will present a table providing the ER, SD, Sharpe, and weights of the equities for each of the above stated portfolios

<u>Results:</u>

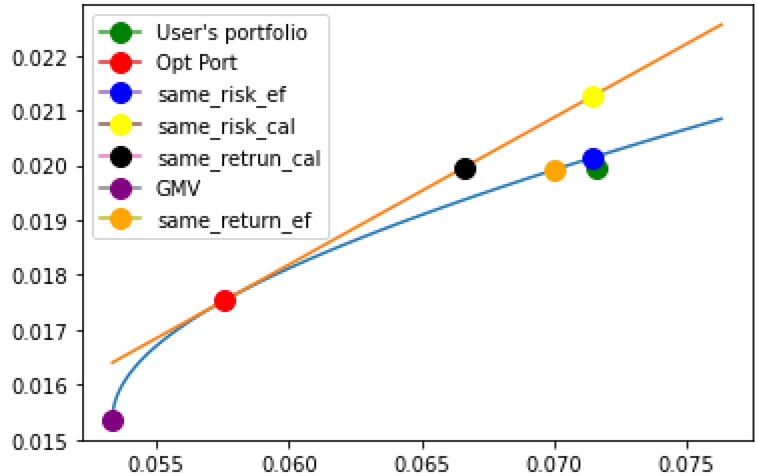
For the results of this project our final output consists of a plot with all the aforementioned portfolios and as well as a table displayed the attributes of each said portfolio.

Example 1:

In this first example, we will display an output with the input weights of:

- FB: 85%
- GOOG: 10%
- JPM: 5%

The image below displays the plot and legend of all said portfolios and the suggested new portfolios on the CAL and the EF:



The image below displays the table for all the portfolios displayed in the plot as well as attributes such as ER, SD, Sharpe, and weights of each of the 3 equities:

Out[6]:

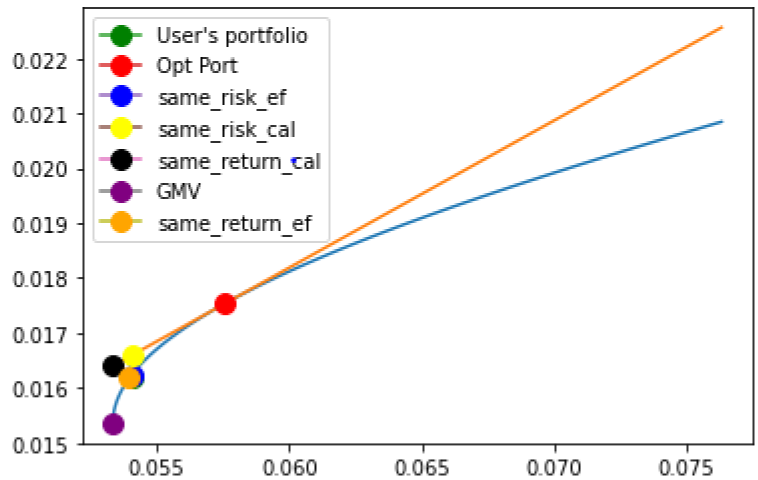
	Initial Portfolio	Efficient Risky Portfolio 1	Efficient Risky Portfolio 2	Optimal Portfolio 1	Optimal Portfolio 2
Expected Return	0.019968	0.019968	0.019915	0.021252	0.019955
Standard Deviation	0.071603	0.071409	0.069976	0.071409	0.066574
Sharpe Ratio	0.249777	0.250458	0.254823	0.268441	0.268441
Weight_FB	85.000000	22.247940	25.115118	51.195771	47.729927
Weight_GOOG	10.000000	46.061067	45.529961	52.801832	49.227261
Weight_JPM	5.000000	31.690992	29.354921	20.120439	18.758328
Weight_RF	0.000000	0.000000	0.000000	-0.241180	-0.157155

Example 2:

In this second example, we will display an output with the input weights of:

- FB: 27%
- GOOG: 39%
- JPM: 34%

The image below displays the plot and legend of all said portfolios and the suggested new portfolios on the CAL and the EF:



The image below displays the table for all the portfolios displayed in the plot as well as attributes such as ER, SD, Sharpe, and weights of each of the 3 equities:

Out[7]:

	Initial Portfolio	Efficient Risky Portfolio 1	Efficient Risky Portfolio 2	Optimal Portfolio 1	Optimal Portfolio 2
Expected Return	0.016206	0.016206	0.016182	0.016593	0.016402
Standard Deviation	0.054110	0.054051	0.053965	0.054051	0.053341
Sharpe Ratio	0.261006	0.261291	0.261259	0.268441	0.268441
Weight_FB	27.000000	73.140348	73.857142	38.751225	37.719097
Weight_GOOG	39.000000	36.633927	36.501151	39.966889	38.902382
Weight_JPM	34.000000	-9.774275	-10.358293	15.229611	14.823974
Weight_RF	0.000000	0.000000	0.000000	0.060523	0.085545

Lessons Learned / Future Actions:

From this assignment, we were exposed to the convenience of using matrix notation in our code when calculating the variance and covariance of a portfolio and how the long-hand formulas can be gruesome if dealing with portfolios larger than 3 equities.

Another pain-point we came across was dealing with the technical differences between matrices, arrays, and the circumstances when which you can alter, add, and multiply each. When performing out calculations in matrix notation in Python, we came across multiple errors when trying to carry out functions on our matrices and arrays previously stated.

Towards the end of the project when calculating the weights of our efficient portfolios, we had to re-store our alpha values as well, for we forgot to assign the list to a variable earlier in the project.

Overall, we found the assignment to be extremely stimulating, challenging, thought-provoking, and undoubtedly beneficial to our understanding of the topics being taught in the classroom.