Financial Math Mini-Project

Ahmer Nadeem Khan

20 February 2025

1 Introduction

In this mini-project, we want to take values of a small set of major stocks on the US Stock Market over a short past period and compute the Global Minimum Variance Portfolio given by this historical data and the results of Portfolio theory. We then want to verify theoretical predictions about the portfolio and discuss briefly some of its properties. We make remarks about our methodology and the results.

2 Objectives and Methodology

Our objectives are the following:

1. Consider the following major stocks (ticker symbols), the first 7 of which are listed on the NASDAQ and the last 2 on the NYSE:

   AAPL, MSFT, AMZN, NVDA, GOOGL, TSLA, META, BRK.B, UNH

2. We obtain weekly closing prices of each stock for the past 27 weeks, starting from the current date. This means getting the closing prices starting on 02/14/2025 ("Today") and retrieving data for the preceding 26 weeks. Note that "Today" corresponds to market close on last Friday. Then, close prices are obtained for the required Fridays before. The first date for the data is 08/16/2024.

3. We then compute the weekly returns on the i-th stock by the formula

$$R^{(i)}_{weekly} = \frac{P^{(i)}_{t+1} - P^{(i)}_{t}}{P^{(i)}_{t}}.$$

4. We then obtain an estimate for the weekly risk-free return, and compute the excess returns on each stock in each period by the formula

$$R_{excess} = R_{total} - R_{risk-free}.$$

This can also be done after the computations; henceforth, return refers to excess return.

5. We compute the covariance matrix, the holdings vector for the Portfolio C (GMV Portfolio), the return, variance, and the standard deviation (risk) of the portfolio using the following formulas:

$$V = \frac{(\mathbf{r} - \mathbf{f})(\mathbf{r} - \mathbf{f})^\top}{n} \quad \mathbf{h}_c = \frac{\mathbf{V}^{-1}\,\mathbf{e}}{\mathbf{e}^\top\,\mathbf{V}^{-1}\,\mathbf{e}}$$

$$f_c = \mathbf{h}_c^\top \mathbf{f} \quad \sigma_c^2 = \frac{1}{\mathbf{e}^\top\,\mathbf{V}^{-1}\,\mathbf{e}}$$

where $\mathbf{e} = \mathbf{1}$ is a vector of ones, $\mathbf{V}$ is the covariance matrix, $\mathbf{h}_c$ is the holdings vector of the Portfolio C, $\mathbf{r}$ is the matrix of returns, and $\mathbf{f} = \mathbb{E}[\mathbf{r}]$.

6. We then present and compare our results and verify properties of the portfolio C.

Please note the following observations:

- We have $p = 9$, the number of assets i.e. 9 stocks.
- We have $n = 26$, the number of time periods over which we have returns, i.e. 26 weeks. The data for closing prices has to contain 27 data points so that the returns can have 26 data points (1 less, by construction).
- The size of the matrix of returns, $\mathbf{r}$, is $p \times n$, where each row corresponds to the returns of the i-th asset over the specified period.
- There are many ways to obtain an estimate for the weekly risk-free return
- $\mathbf{f} = \mathbb{E}[\mathbf{r}]$ is computed by averaging across each row to get an *estimate* of the expected (weekly) return of the particular stock. We choose a simple arithmetic mean as our expected return. Note that this implicitly uses historical returns as an estimate of independent realizations of returns in the current period, and also assumes a uniform distribution for this random variable. This approach is unsophisticated and brutal, but provides the simplicity we need for our tasks, which is to simply verify results from Portfolio theory (Portfolio theory assumes $\mathbf{f}$ is given apriori).
- We will not verify the formulas presented here, but simply state that $\mathbf{h}_c$ is the characteristic portfolio of the vector of ones, $\mathbf{e}$, and solves the following optimization problem:

$$\begin{aligned} \min_{\mathbf{h} \in \mathbb{R}^p} \quad & f(\mathbf{h}) = \mathbf{h}^\top \mathbf{V} \mathbf{h} \quad \text{(min risk)} \\ \text{subject to} \quad & g(\mathbf{h}) = \mathbf{h}^\top \mathbf{e} = 1 \ \text{(fully invested)}, \end{aligned}$$

which is a simple convex, quadratic minimization problem, easily solved by the method of Lagrange multipliers:

$$\nabla f = \mu \nabla g \implies 2\mathbf{V}\mathbf{h} = \mu\,\mathbf{e} \implies \mathbf{h} = \frac{\mu}{2}\,\mathbf{V}^{-1}\,\mathbf{e} = k\,\mathbf{h}_c$$

and the constraint gives us that

$$1 = \mathbf{h}^\top \mathbf{e} = (k\,\mathbf{h}_c)^\top \mathbf{e} = k(\mathbf{h}_c^\top \mathbf{e}) = k$$

giving us $\mathbf{h} = \mathbf{h}_c$. Note that we have used the properties of characteristic portfolios and basic vector calculus above.

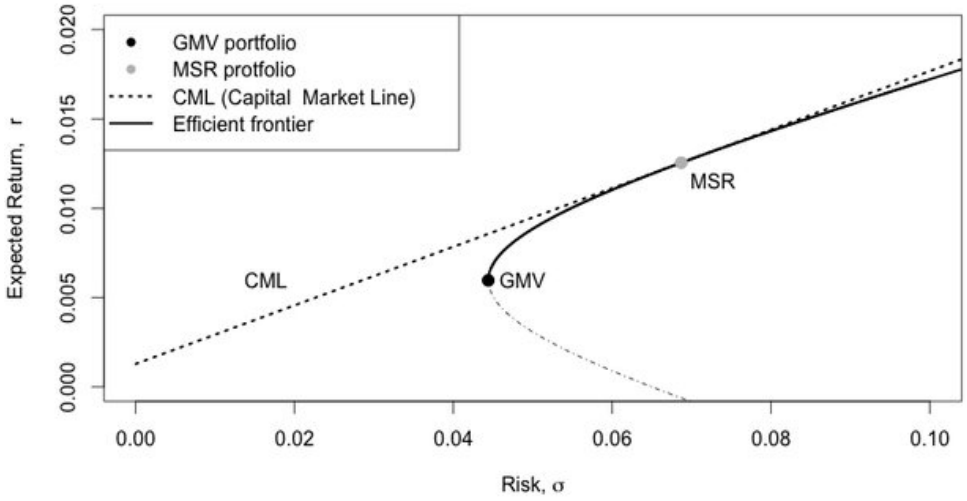

Figure 1: The Efficient Frontier (MSR refers to the Max. Sharpe Ratio) [2]

The values in the above plot are arbitrary.

2.1 Data Collection and Implementation

We need to collect the following data:

- Stock Market price data
- Risk-free rate

Here is how we obtain and pre-process this data:

- We obtain the close price data using GOOGLEFINANCE on Google Sheets using formulas to retrieve data with the following command:

```
=GOOGLEFINANCE("NASDAQ:AAPL", "price", TODAY()-189, TODAY(), "WEEKLY")
```

which is an example with Apple as our stock of interest. The days shift 189 is calculated as $189 = 7 \times 26$. Note the command "price" automatically retrieves the close for the specified time period. See [3] for more details.

- We add the data for each stock on a separate tab in the same Google Sheet (.xlsx format), and publish the sheet on the web, so we can dynamically access the data using the URL (i.e. we can freely edit the sheet, and simply re-run the processing code). The URL is accessed using simple functionality provided by the *pandas* library in Python (see code appendix).

- To obtain the appropriate estimate of the risk-free return, we simply get the annualized return on the 26-week Treasury Bill on the starting day of our data, and scale to get the weekly return on the T-bill using the formula:
$$r_{weekly} = (1 + return_{annual})^{1/52} - 1$$
which is obvious from properties of compounding. The value of the annualized return can be obtained from various sources online. I used the data reported by the U.S. Department of Treasury [4]. The value (4.80%) was hard-coded into the code.

- We present both weekly and annual returns (compounded) and variances wherever appropriate, and use the following formulas to convert between them:
$$r_{annual} = (1 + return_{weekly})^{52} - 1$$
$$\sigma^2_{annual} = 52 \times \sigma^2_{weekly}$$
$$\sigma_{annual} = \sqrt{52} \times \sigma_{weekly}$$

You can download the data from the published Google Sheet at the URL available in the bibliography, and also the code appendix. [1]

We implement our computation in Python using a Jupyter Notebook (.ipynb file) and produce the computations and outputs in different blocks of code for easier reproducibility. We use the basic libraries *pandas* for handling and extracting data, and *numpy* for computations on arrays. For simple mathematical computations we use the library *math*, and *matplotlib* for producing graphs/tables when necessary. For some plotting and latex code generation, I used generative AI (specifically ChatGPT o3-mini-high).

3 Results and Conclusions

I will first simply display the required values and variables obtained from the code.

The (weekly) covariance matrix is (up to 2 decimal places):

$$\mathbf{V} = 10^{-5} \begin{bmatrix} 89.40 & 4.54 & 17.75 & -9.86 & 34.27 & 60.01 & 26.44 & 12.43 & -14.83 \\ 4.54 & 81.63 & 61.23 & 119.28 & 37.78 & 135.93 & 34.78 & -4.56 & -12.61 \\ 17.75 & 61.23 & 119.75 & 72.73 & 82.57 & 90.34 & 65.61 & -1.78 & -1.20 \\ -9.86 & 119.28 & 72.73 & 472.28 & 20.86 & 119.63 & 60.12 & 0.68 & 10.50 \\ 34.27 & 37.78 & 82.57 & 20.86 & 134.75 & 146.98 & 45.41 & -4.98 & -2.55 \\ 60.01 & 135.93 & 90.34 & 119.63 & 146.98 & 881.73 & 57.84 & -3.69 & 4.49 \\ 26.44 & 34.78 & 65.61 & 60.12 & 45.41 & 57.84 & 134.50 & -1.62 & 14.62 \\ 12.43 & -4.56 & -1.78 & 0.68 & -4.98 & -3.69 & -1.62 & 48.92 & 16.71 \\ -14.83 & -12.61 & -1.20 & 10.50 & -2.55 & 4.49 & 14.62 & 16.71 & 124.37 \end{bmatrix}$$

Note: Multiply the entries in the matrix by $10^{-}5$ to obtain the exact values.

The holdings vector i.e. the propotion of wealth in each asset is (up to 3 decimal places):

$$\mathbf{h}_c = \begin{bmatrix} 0.163 \\ 0.453 \\ -0.126 \\ -0.053 \\ 0.115 \\ -0.062 \\ 0.039 \\ 0.304 \\ 0.168 \end{bmatrix}$$

In terms of approximate percentage of wealth, we get

$$\mathbf{h}_c = \begin{bmatrix} \text{AAPL} : 16.3 \\ \text{MSFT} : 45.3 \\ \text{AMZN} : -12.6 \\ \text{NVDA} : -5.3 \\ \text{GOOGL} : 11.5 \\ \text{TSLA} : -6.2 \\ \text{META} : 3.9 \\ \text{BRK.B} : 30.4 \\ \text{UNH} : 16.8 \end{bmatrix} \%$$

Note: The negative sign indicates short positions.

Statistic	Weekly	Annualized
Return	-0.002098	-0.103454
Variance	0.000174	0.009048
Standard Deviation	0.013191	0.095122

Table 1: Portfolio C: Weekly and Annualized Statistics

Stock	Weekly Variance	Annual Variance
AAPL	0.000894	0.046489
MSFT	0.000816	0.042447
AMZN	0.001198	0.062271
NVDA	0.004723	0.245586
GOOGL	0.001348	0.070071
TSLA	0.008817	0.458500
META	0.001345	0.069938
BRK.B	0.000489	0.025439
UNH	0.001244	0.064672

Table 2: Weekly and Annualized Variances for Individual Stocks

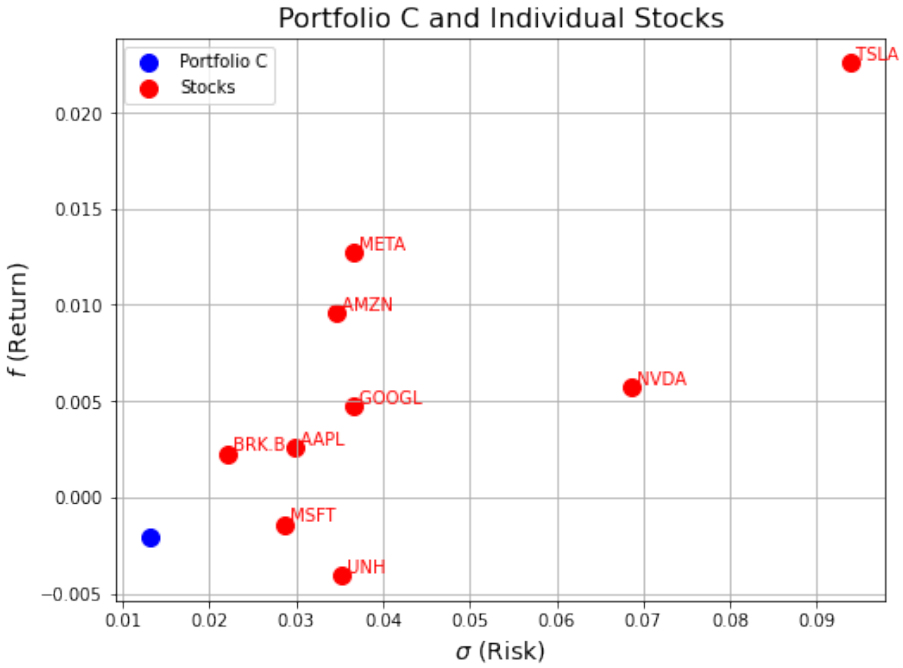

Figure 2: Weekly Risk and Returns

Conclusions: We can first easily verify that the portfolio is fully invested by noticing that the sum equals 1 (this is done in the code). We also have both long and short positions in the portfolio. From the table and the graph, we can immediately deduce that C does indeed have the smallest risk, at least when compared individually to the market (our smaller set of stocks). If any stock had a smaller variance, we would be completely invested in that one stock - as it turns out, we can minimize variance by a combination of stocks. In terms of minimizing risk, this is the optimal fully invested portfolio. About 1.08 of the portfolio positions are in the 4 stocks with the smallest variances (MSFT, BRK.B, UNH, AAPL, in decreasing order of position). We can also tell that there are no constraints on the return, since C has an expected negative return. We also have short positions in AMZN, TSLA, and NVDA.

The short positions in volatile stocks like NVIDIA and Tesla indicate the short horizon of our historical data, since these may be imprudent investments given macroeconomic trends or fundamental analyses - the true, overall volatility of these stocks is not priced in. It also appears from the expected negative return that we are capturing a overall downturn in the market, at least in these major stocks. If the GMV portfolio actually had negative expected (excess) return, it

would not make sense to be invested in it; instead, you would fully invest in the risk-free asset to get guaranteed positive return (0 excess return) at 0 risk. On the diagram, this means moving towards the origin, we reduce risk and increase expected returns. However, theoretically, the properties of C are verified: it is indeed a low-risk portfolio to invest in.

4 Code Appendix

```
import pandas as pd
import numpy as np
import math

# Published XLSX file URL
url = "https://docs.google.com/spreadsheets/d/e/2PACX-1vRDOPYPQnWr
    -BgDRS1wKNFQ3rNFK9VhsICkAHFQs-OQG-2FfJ7CJiGXYo2vjoKTBThCKxsu6jEUKWBK/pub?output=xlsx"

# Load all sheets into a dictionary: keys are sheet names, values are DataFrames
sheets_dict = pd.read_excel(url, sheet_name=None)

# Create a dictionary to hold just the second column from each sheet
data = {}
for stock, df in sheets_dict.items():
    # Extract the second column (index 1) and ignore the first column (dates)
    data[stock] = df.iloc[:, 1]

result = pd.DataFrame(data)

print(result.head())

shape = np.shape(result)
n = shape[0]
p = shape[1]

prices = result.to_numpy()
returns = np.zeros((n-1,p))

for i in range(1, n):
    for j in range(p):
        returns[i - 1, j] = (prices[i, j] - prices[i - 1, j]) / prices[i - 1, j]

risk_free_annual = 4.80/100
risk_free_weekly = (1+risk_free_annual)**(1/52) - 1
excess_returns = (returns - risk_free_weekly).T

f = np.mean(excess_returns, axis=1, keepdims=True)
```

```
demeaned_returns = excess_returns - f
sample_cov = (demeaned_returns @ demeaned_returns.T) / (n-1)

V_inv = np.linalg.inv(sample_cov)
e = np.ones(p)
numer = V_inv @ e
denom = e.T @ numer
h_c = numer / denom

return_c = h_c.T @ f
var_c = 1/denom
std_c = math.sqrt(var_c)

var_stocks = np.diag(sample_cov)

# Annualize portfolio C values:
annual_return_c = (1 + return_c)**52 - 1
annual_var_c    = 52 * var_c
annual_std_c    = np.sqrt(52) * std_c
annual_var_stocks = 52 * var_stocks

print("Holdings Vector (Portfolio C): ", h_c)
print("Return of Portfolio C: ", return_c)
print("Variance and Standard Deviation of Portfolio C: ", var_c, std_c)
print("Variances of the Stocks: ", var_stocks)
print("Fully Invested?: ", np.sum(h_c))
```

References

[1] Published google sheets data. https://docs.google.com/spreadsheets/d/e/2PACX-1vRD0PYPQnWr-BgDRS1wKNFQ3rNFK9VhsICkAHFQs-0QG-2FfJ7CJiGXYo2vjoKTBThCKxsu6jEUKWBK/pub?output=xlsx, n.d. Accessed: Febrary 20, 2025.

[2] J.F. Monge, Mercedes Landete, and José Ruiz. Sharpe portfolio using a cross-efficiency evaluation. 10 2016.

[3] Google Support. Google docs help – answer 3093281. https://support.google.com/docs/answer/3093281?hl=en, n.d. Accessed: Febrary 20, 2025.

[4] U.S. Department of the Treasury. Daily treasury bill rates. https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_bill_rates&field_tdr_date_value=2024, 2024. Accessed: February 20, 2025.