Mini Project Part 1

Jenny Petrova

February 2025

1 Executive Summary

This project examines a sample Portfolio P invested in the following $p = 9$ stocks: AAPL, MSFT, AMZN, NVDA, GOOGL, TSLA, META, BRK.B, UNH. The weekly closing prices for each stock are observed over a time period of $n = 26$ weeks, from 2024-08-02 to 2025-01-24. The portfolio's excess returns r_P are determined with respect to a risk-free rate r_F, the (weekly) sample covariance matrix Y are computed, and the holdings vector h_c is calculated for the minimum risk, fully invested portfolio C. The feasibility of this portfolio is assessed and comments are made regarding the level of risk of these investments.

2 Description of the Mathematics

1. Let P_t be the time t value of portfolio P. Then the weekly return of a stock i is given by

$$r_t^i = \frac{P_t^i - P_{t-1}^i}{P_{t-1}^i}, \quad 1 \leq i \leq 9, \quad 1 \leq t \leq 26 \tag{1}$$

   The weekly excess returns are then calculated by

$$r_P^i(t) = r_t^i - r_F, \tag{2}$$

   where r_F is the risk-free rate (set to a constant rate for ease of computation). A $(p \times n)$ matrix X of weekly excess returns such that $r_P^i(t) \in X$.

2. Let $f_P^i = \frac{1}{26}\sum_{t=1}^{26} r_P^i(t)$ be the expected (average) weekly excess returns for each stock i. The de-meaned excess returns for each stock i are found by subtracting f_P^i from each weekly excess return, such that

$$\tilde{r}_P^i(t) = r_P^i(t) - f_P^i, \quad 1 \leq i \leq 9, \quad 1 \leq t \leq 26 \tag{3}$$

   for all $r_P^i(t) \in X$, where $\tilde{r}_P^i(t) \in Y$. The (weekly) sample covariance matrix is given by

$$V = \frac{YY^T}{n} \tag{4}$$

3. We define the holdings vector for the minimum risk, fully invested portfolio C by

$$h_C = \frac{V^{-1}e}{e^T V^{-1} e} \tag{5}$$

where $e^T = [1\,1\,\ldots 1]$. The value of the expected excess returns of the fully invested portfolio C is

$$f_C = h_C^T \mathbf{f} \tag{6}$$

where $\mathbf{f} = [f_P^1, f_P^2, \ldots, f_P^9]$ is the expected excess returns vector for all stocks in portfolio P. We compute the variance of portfolio C by

$$\sigma_C^2 = h_C^T V h_C, \tag{7}$$

and the standard deviation (i.e. risk) is simply

$$\sigma_C = \sqrt{h_C^T V h_C}. \tag{8}$$

The diagonal entries of the covariance matrix V are simply the variances of each asset i. Hence

$$[\sigma_1^2, \sigma_2^2, \ldots, \sigma_9^2] = diag(V). \tag{9}$$

Note that this portfolio is fully invested in risky assets (i.e. no cash assets). Any fully invested portfolio P will have $\sigma_P = \sigma_C$, so in context of this portfolio we may use these variable notations interchangeably.

3 Numerical Results

All stock (closing) price data is accessed using the 'GOOGLEFINANCE' command in Google Sheets. The portfolio is constructed in Google Sheets and exported as a .xlsx file into a Python program. All computations are performed using the NumPy and Pandas Python programming libraries on the PyCharm integrated development environment. Weekly stock price data for each stock is obtained by:

=GOOGLEFINANCE("AAPL", "CLOSE", DATE(2024,7,25), DATE(2024,1,25), "DAILY")

where the data for each stock is retrieved by changing the stock's corresponding ticker symbol. The data is obtained for the weekly closing prices of stocks from week 2024-07-26 (time $t = 0$) to week 2025-01-24 (time $t = 26$). The initial portfolio P is given by:

Time	AAPL	MSFT	AMZN	NVDA	GOOGL	TSLA	META	BRK.B	UNH
0	217.96	425.27	182.50	113.06	167.00	219.80	465.70	437.66	569.72
1	219.86	408.49	167.90	107.27	166.66	207.67	488.14	428.36	589.83
2	216.24	406.02	166.94	104.75	163.67	200.00	517.77	431.67	558.76
3	226.05	418.47	177.06	124.58	162.96	216.12	527.42	444.51	577.68
4	226.84	416.79	177.04	129.37	165.62	220.32	528.00	453.38	584.51
5	229.00	417.14	178.50	119.37	163.38	214.11	521.31	475.92	590.20
6	220.82	401.70	171.39	102.83	150.92	210.73	500.27	459.42	596.88
7	222.50	430.59	186.49	119.10	157.46	230.29	524.62	447.61	594.32
8	228.20	435.27	191.60	116.00	163.59	238.25	561.35	455.31	575.00
9	227.79	428.02	187.97	121.40	163.95	260.46	567.36	457.47	581.85
10	226.80	416.06	186.51	124.92	167.06	250.08	595.94	461.97	591.20
11	227.55	416.32	188.82	134.80	163.24	217.80	589.95	460.21	598.05
12	235.00	418.16	188.99	138.00	163.42	220.70	576.47	464.80	569.61
13	231.41	428.15	187.83	141.54	165.27	269.19	573.25	454.01	564.56
14	222.91	410.37	197.93	135.40	171.29	248.98	567.16	452.14	567.56
15	226.96	422.54	208.18	147.63	178.35	321.22	589.34	463.41	615.81
16	225.00	415.00	202.61	141.98	172.49	320.72	554.08	470.28	592.23
17	229.87	417.00	197.12	141.95	164.76	352.56	559.14	476.57	590.87
18	237.33	423.46	207.89	138.25	168.95	345.16	574.32	483.02	610.20
19	242.84	443.57	227.03	142.44	174.71	389.22	623.77	470.50	549.62
20	248.13	447.27	227.46	134.25	189.82	436.23	620.35	457.90	520.48
21	254.49	436.60	224.92	134.70	191.41	421.06	585.25	453.20	500.13
22	255.59	430.53	223.75	137.01	192.76	431.66	599.81	456.51	509.99
23	243.36	423.35	224.19	144.47	191.79	410.44	604.63	453.56	513.00
24	236.85	418.95	218.94	135.91	192.04	394.74	615.86	442.66	520.69
25	229.98	429.03	225.94	137.71	196.00	426.50	612.77	467.95	509.76
26	222.78	444.06	234.85	142.62	200.21	406.58	647.49	463.19	532.51

We calculate the returns from week $t = 1$ (2024-08-02) to week $t = 26$ (2025-01-24) and subtract the risk-free rate r_F from each cell to generate a (9×26) matrix P of excess returns r_P^i for each asset i at each time t (1). The following Python commands are executed:

```python
prices = pd.read_excel("portfolioP.xlsx", index_col=0)

returns = prices.pct_change().dropna()

excess_returns = returns - 0.00087
```

We set the risk-free rate to be $r_F = 0.00087$. This value is determined by taking the average of daily 3-Month Treasury Bill (T-Bill) rates from 2024-08-02 to 2025-01-24, obtained from public U.S. Department of Treasury records. These rates are often used by U.S.-based investors as a proxy for the true risk-free rate, since we expect there to be no risk of default on obligations made by the government. We found the average annualized 3-Month T-Bill rate to be

$r_{F_{\text{annual}}} = .045$. The average weekly rate is found by $.045/52 \approx 0.00087$. The excess returns of the portfolio P are:

Time	**AAPL**	**MSFT**	**AMZN**	...	**META**	**BRK.B**	**UNH**
1	0.007847	-0.040327	-0.080870	...	0.047316	-0.022119	0.034428
2	-0.017335	-0.006917	-0.006588	...	0.059830	0.006857	-0.053546
3	0.044496	0.029794	0.059751	...	0.017768	0.028875	0.032991
4	0.002625	-0.004885	-0.000983	...	0.000230	0.019085	0.010953
5	0.008652	-0.000030	0.007377	...	-0.013540	0.048845	0.008865
6	-0.036591	-0.037884	-0.040702	...	-0.041230	-0.035540	0.010448
7	0.006738	0.071049	0.087233	...	0.047804	-0.026576	-0.005159
8	0.024748	0.009999	0.026531	...	0.069143	0.016332	-0.033378
9	-0.002667	-0.017526	-0.019816	...	0.009836	0.003874	0.011043
10	-0.005216	-0.028813	-0.008637	...	0.049504	0.008967	0.015199
11	0.002437	-0.000245	0.011515	...	-0.010921	-0.004680	0.010717
12	0.031870	0.003550	0.000030	...	-0.023719	0.009104	-0.048425
13	-0.016147	0.023020	-0.007008	...	-0.006456	-0.024084	-0.009736
14	-0.037601	-0.042398	0.052902	...	-0.011494	-0.004989	0.004444
15	0.017299	0.028786	0.050916	...	0.038237	0.024056	0.084143
16	-0.009506	-0.018714	-0.027626	...	-0.060700	0.013955	-0.039161
17	0.020774	0.003949	-0.027966	...	0.008262	0.012505	-0.003166
18	0.031583	0.014622	0.053767	...	0.026279	0.012664	0.031844
19	0.022347	0.046620	0.091198	...	0.085232	-0.026790	-0.100149
20	0.020914	0.007471	0.001024	...	-0.006353	-0.027650	-0.053888
21	0.024762	-0.024726	-0.012037	...	-0.057451	-0.011134	-0.039969
22	0.003452	-0.014773	-0.006072	...	0.024008	0.006434	0.018845
23	-0.048720	-0.017547	0.001096	...	0.007166	-0.007332	0.005032
24	-0.027620	-0.011263	-0.024288	...	0.017703	-0.024902	0.014120
25	-0.029876	0.023190	0.031102	...	-0.005887	0.056262	-0.021861
26	-0.032177	0.034163	0.038565	...	0.055791	-0.011042	0.043759

We compute the expected excess returns f_P^i of each asset i (2) by transposing the above table (i.e. matrix) and calculate the average (*expected* excess returns) for each row i in the matrix P. The expected excess return for stock each i is subtracted from the stock's excess return value at each time t (3). This transforms matrix P into the de-meaned excess returns matrix Y. The following commands are executed:

```
excess_returns = excess_returns.T

expected_returns = excess_returns.mean(axis=1)
Y = excess_returns.sub(expected_returns, axis=0)
```

and matrix Y is displayed:

Time	1	2	3	...	24	25	26
AAPL	0.007575	-0.017608	0.044224	...	-0.027893	-0.030148	-0.032450
MSFT	-0.041487	-0.008077	0.028633	...	-0.012423	0.022030	0.033002
AMZN	-0.090501	-0.016219	0.050119	...	-0.033919	0.021471	0.028934
NVDA	-0.062489	-0.034770	0.178030	...	-0.070529	0.001966	0.024377
GOOGL	-0.009536	-0.025440	-0.011838	...	-0.006196	0.013121	0.013980
TSLA	-0.083095	-0.064842	0.052692	...	-0.066160	0.052550	-0.074614
META	0.034763	0.047278	0.005215	...	0.005151	-0.018440	0.043239
BRK.B	-0.023695	0.005281	0.027299	...	-0.026478	0.054686	-0.012618
UNH	0.037182	-0.050792	0.035745	...	0.016874	-0.019107	0.046513

Then the weekly sample covariance matrix V is calculated using matrix-matrix multiplication (4) such that the following code:

```
V = Y @ Y.T / 26
```

yields a (9×9) matrix:

Assets	AAPL	MSFT	AMZN	...	META	BRK.B	UNH
AAPL	0.000600	0.000226	0.000214	...	0.000121	0.000116	-0.000115
MSFT	0.000226	0.000739	0.000792	...	0.000432	0.000049	-0.000090
AMZN	0.000214	0.000792	0.001530	...	0.000563	0.000130	-0.000092
NVDA	0.000512	0.001124	0.001473	...	0.000719	0.000258	0.000551
GOOGL	0.000190	0.000364	0.000671	...	0.000464	-0.000037	-0.000115
TSLA	0.000707	0.001411	0.001076	...	0.000574	0.000142	0.000011
META	0.000121	0.000432	0.000563	...	0.001346	-0.000065	0.000097
BRK.B	0.000116	0.000049	0.000130	...	-0.000065	0.000531	0.000161
UNH	-0.000115	-0.000090	-0.000092	...	0.000097	0.000161	0.001402

We then create a (9×1) vector of 1's (denoted e) and find the holdings vector of excess returns h_C (5), using the following commands:

```
ones = np.ones(9)
V_inv = np.linalg.inv(V)
h_C = V_inv @ ones / (ones.T @ V_inv @ ones)
```

We find h_C to be

Asset	h_C
AAPL	0.21231822
MSFT	0.4233215
AMZN	-0.11839153
NVDA	-0.08360823
GOOGL	0.16983762
TSLA	-0.06012498
META	0.02560046
BRK.B	0.26545223
UNH	0.16559472

The value of the expected excess returns f_c (6) is computed by taking the inner product of the holdings vector h_C and the excess returns vector $\mathbf{f}$:

```
portfolio_expected_returns = h_C.T @ expected_returns
```

yielding

f_C	-0.001677491788890427

Lastly, we are interested in the variance and standard deviation of the fully invested portfolio C (7), as well as the variance of the excess returns for each individual stock i in our original portfolio P, stored in the diagonal entries of the sample covariance matrix V (9). We compute

```
portfolio_var = h_C.T @ V @ h_C
portfolio_std_dev = np.sqrt(portfolio_var)

stock_var = np.diag(V)
```

The resulting output is combined into the following table:

Variable	Values
σ_C^2	0.00015940
σ_C	0.01262555
σ_{APPL}	0.00059977
σ_{MSFT}	0.00073915
σ_{AMZN}	0.00153034
σ_{NVDA}	0.00473809
σ_{GOOGL}	0.00099978
σ_{TSLA}	0.00856369
σ_{META}	0.00134573
$\sigma_{\text{BRK.B}}$	0.00053061
σ_{UNH}	0.00140234

The above results for the portfolio return, variance, and standard deviation, along with the individual stock variances, are presented as *weekly* quantities. The annualized values are found by scaling the values by 52 weeks/year. Hence we compute

```
annualized_return = portfolio_expected_returns * 52
annualized_var = portfolio_var * 52
annualized_std_dev = portfolio_std_dev * np.sqrt(52)
annualized_stock_var = stock_var * 52
```

and the resulting values are

Variable	Values
$f_{P_{\text{annualized}}}$	-0.08722957
$\sigma^2_{C_{\text{annualized}}}$	0.00828903
$\sigma_{C_{\text{annualized}}}$	0.09104413
$\sigma_{\text{APPLannualized}}$	0.03118808
$\sigma_{\text{MSFTannualized}}$	0.03843605
$\sigma_{\text{AMZNannualized}}$	0.07957792
$\sigma_{\text{NVDAannualized}}$	0.24638079
$\sigma_{\text{GOOGLannualized}}$	0.05198849
$\sigma_{\text{TSLAannualized}}$	0.44531198
$\sigma_{\text{METAannualized}}$	0.06997805
$\sigma_{\text{BRK.Bannualized}}$	0.02759162
$\sigma_{\text{UNHannualized}}$	0.07292186

4 Analysis of Results and Conclusion

Assessing the risk of a portfolio depends largely on the investor's risk-tolerance, and the feasibility of a portfolio must be considered in the context of a risk-reward trade-off. We know the portfolio C is *efficient*, since it is the minimum risk, fully invested portfolio for our chosen (risky) assets. Looking at our portfolio's standard deviation of excess returns, we assess that this portfolio is a low-risk investment, as we found the value of the standard deviation to be $\sigma_C \approx 0.0126 = 1.26\%$ for the weekly excess returns and $\sigma_C \approx 0.0910 = 9.10\%$ for the annualized excess returns. However, with such low risk we also see negative expected excess returns, where $f_C \approx -0.00168 = -0.168\%$ are the weekly expected excess returns and $f_C \approx -0.0872 = -8.72\%$ are the annualized expected excess returns for our portfolio. With this information, we cannot conclude that this portfolio is reasonable, as we do *not* expect to make a positive return on our investment.

5 Appendix: Complete Code

```python
import numpy as np
import pandas as pd

# Get weekly closing price data
prices = pd.read_excel("portfolioP.xlsx", index_col=0)
print("\nPortfolio Closing Prices:\n",prices)

# Compute weekly returns
# Drop first row (t = 0)
returns = prices.pct_change().dropna()

# Compute excess returns
excess_returns = returns - 0.00087
print("\nPortfolio Excess Returns\n",excess_returns)

# Transpose excess returns matrix
excess_returns = excess_returns.T # (p x n) matrix of p=9 stocks and n=26 weeks

# Compute de-meaned excess returns matrix Y
expected_returns = excess_returns.mean(axis=1)
Y = excess_returns.sub(expected_returns, axis=0)
print("\nY:\n", Y)

# Compute sample covariance matrix V
V = Y @ Y.T / 26
print("\nV:\n", V)

# Compute holdings vector h_C for minimum variance, fully invested portfolio C
ones = np.ones(9)
V_inv = np.linalg.inv(V)
h_C = V_inv @ ones / (ones.T @ V_inv @ ones)
print("\nh_C:\n", h_C)

# Compute portfolio expected excess return, variance, and standard deviation
portfolio_expected_returns = h_C.T @ expected_returns
print("\nPortfolio Expected Returns\n", portfolio_expected_returns)
portfolio_var = h_C.T @ V @ h_C
print("\nPortfolio Variance:\n", portfolio_var)
portfolio_std_dev = np.sqrt(portfolio_var)
```

```python
print("\nPortfolio Standard Deviation:\n",
    portfolio_std_dev)
stock_var = np.diag(V)
print("\nStock Variance:\n", stock_var)

# Scale by 52 for annualized results
annualized_return = portfolio_expected_returns * 52
print("\nAnnualized Return:\n", annualized_return)
annualized_var = portfolio_var * 52
print("\nAnnualized Variance:\n", annualized_var)
annualized_std_dev = portfolio_std_dev * np.sqrt(52)
print("\nAnnualized Standard Deviation:\n",
    annualized_std_dev)
annualized_stock_var = stock_var * 52
print("\nAnnualized Stock Variance:\n",
    annualized_stock_var)
```