PROJECT 2
PORTFOLIO OPTIMIZATION

PORTFOLIO ANALYSIS AND MANAGEMENT (FIN 653) - Prof. Natalia Gershun

GROUP - 3

Anisha Sharma

Huy Nguyen

Kaustubh Khadpe

Mary Gade

Tulsi Dalsania

Objective:

This project aims to apply the mean-variance optimization (MVO) approach to construct an optimal portfolio of risky securities that aligns with an investor's specified risk tolerance level. By analyzing historical price data, estimating expected returns and covariance matrices, and implementing the MVO algorithm, the project will determine the optimal portfolio weights that maximize expected returns for a given risk level or minimize portfolio variance for a given target return. The performance of the constructed portfolio will be evaluated using various risk-adjusted return metrics, and the impact of varying risk tolerance levels on portfolio composition will be analyzed.

Deliverable 1: Briefly describe why you selected each of 20 stocks to include in your portfolio. Try to be creative and visually organize this information in a way that it is easily digested.

We selected the 20 stocks shown in the tables below in an attempt to construct a diversified portfolio with the following characteristics:

Exposure to different sectors of the economy: The stocks in the portfolio are spread across seven different sectors, including consumer discretionary, technology, healthcare, industrials, energy, financial services, and consumer staples. This helps to reduce the portfolio's risk sensitivity to any one sector.

Mix of growth and value stocks: The portfolio includes a mix of growth stocks and value stocks. Growth stocks are expected to grow at a faster rate than the overall market, while value stocks are trading below their intrinsic value. This mix helps to balance the portfolio's risk and return potential.

Mix of high-quality and low-quality stocks: The portfolio includes a mix of high-quality stocks and low-quality stocks. High-quality stocks are typically characterized by strong profitability, low debt levels, and experienced management teams. Low-quality stocks may have some of these characteristics, but they may also be riskier. This mix helps to enhance the portfolio's return potential while still managing risk.

Mix of high-beta and low-beta stocks: The portfolio includes a mix of high-beta stocks and low-beta stocks. Beta is a measure of a stock's volatility relative to the overall market. High-beta stocks are more volatile, while low-beta stocks are less volatile. This mix helps to reduce the portfolio's overall volatility without sacrificing too much potential return.

Table 1	Sector - Consumer Discretionary					
Stock	**Size**	**Growth**	**Quality**	**Beta**	**Momentum**	**Description**
Tesla (TSLA)	Large	Growth	High	High	High	The leading electric vehicle manufacturer, with a strong brand and loyal customer base.
Nike (NKE)	Large	Value	High	Low	Moderate momentum	A leading sportswear company with a strong brand and global reach.
Public Storage (PSA)	Large	Value	High	Low	Moderate momentum	A leading self-storage company with a strong market position.
Starbucks (SBUX)	Large	Growth	High	High	Moderate momentum	A leading coffeehouse chain with a strong brand and global reach.
Walmart (WMT)	Large	Value	High	Low	Moderate momentum	The largest retailer in the world.
Costco (COST)	Large	Value	High	Low	Moderate momentum	A leading membership-only warehouse club.

Table 2	Sector - Healthcare					
Stock	**Size**	**Growth**	**Quality**	**Beta**	**Momentum**	**Description**
Eli Lilly and Company (LLY)	Large	Growth	High	High	Moderate momentum	A leading pharmaceutical company with a strong pipeline of new drugs in development.
UnitedHealth Group	Large	Value	High	Low	Moderate momentum	The largest health insurance company in the United States.

(UNH)						

Table 3	**Sector - Industrials**					
Stock	**Size**	**Growth**	**Quality**	**Beta**	**Momentum**	**Description**
Caterpillar (CAT)	Large	Value	High	Low	Moderate momentum	A leading construction and mining equipment manufacturer.
United Parcel Services (UPS)	Large	Value	High	Low	Moderate momentum	A leading global shipping and logistics company.
Boeing (BA)	Large	Value	High	Low	Moderate momentum	A leading aerospace manufacturer with a strong global presence.

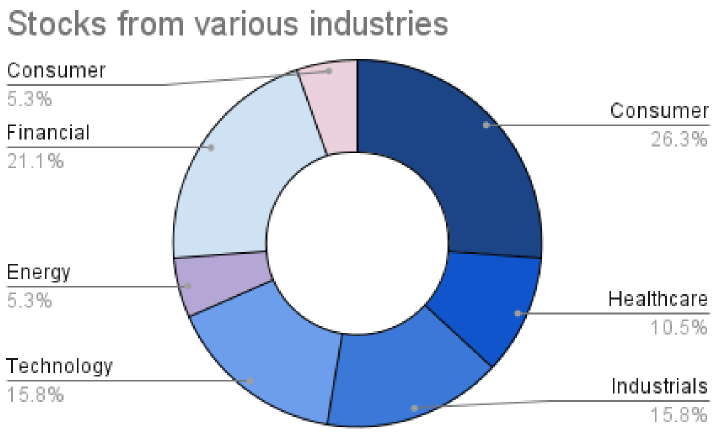
Table 4	**Sector - Technology**					
Stock	**Size**	**Growth**	**Quality**	**Beta**	**Momentum**	**Description**
Nvidia (NVDA)	Large	Growth	High	High	Strong momentum	A leading semiconductor company with a strong position in the gaming and data center markets.
Apple (AAPL)	Large	Growth	High	High	Strong	One of the world's largest and most innovative companies, with a strong track record of growth and profitability.
Advanced Micro Devices (AMD)	Large	Growth	High	High	Strong	A leading semiconductor company with a strong position in the data center and PC markets.

Table 5	Sector - Energy					
Stock	**Size**	**Growth**	**Quality**	**Beta**	**Momentum**	**Description**
Exxon Mobil (XOM)	Large	Value	High	Low	Moderate momentum	One of the world's largest oil and gas companies.

Table 6	Sector - Financial Services					
Stock	**Size**	**Growth**	**Quality**	**Beta**	**Momentum**	**Description**
JPMorgan Chase (JPM)	Large	Value	High	Low	Moderate momentum	One of the world's largest banks.
Bank of America (BOFA)	Large	Value	High	Low	Moderate momentum	One of the world's largest banks.
iShares Gold Trust Micro (IAUM)	Large	Value	High	Low	Moderate momentum	A leading asset manager.

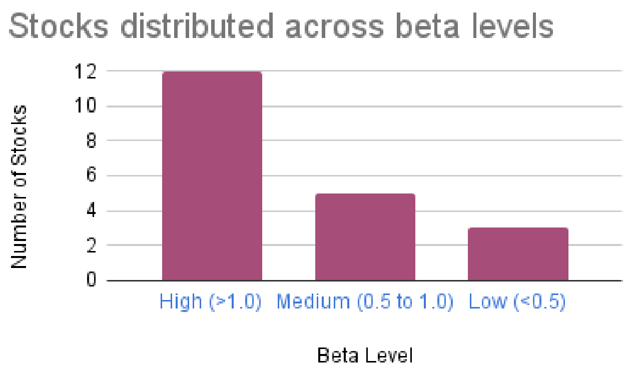
Table 5	ETF					
Stock	**Size**	**Growth**	**Quality**	**Beta**	**Momentum**	**Description**
Vanguard S&P 500 ETF	Large	Value	High	Low	Moderate momentum	Tracks the performance of the S&P 500 index, which is a broad market index of 500 of the largest U.S. companies.

Table 7	Sector - Consumer Staples					
Stock	**Size**	**Growth**	**Quality**	**Beta**	**Momentum**	**Description**
Coca-Cola (KO)	Large	Value	High	Low	Moderate momentum	The world's leading beverage company, with a strong brand and global reach.



Showcasing stocks from various industries

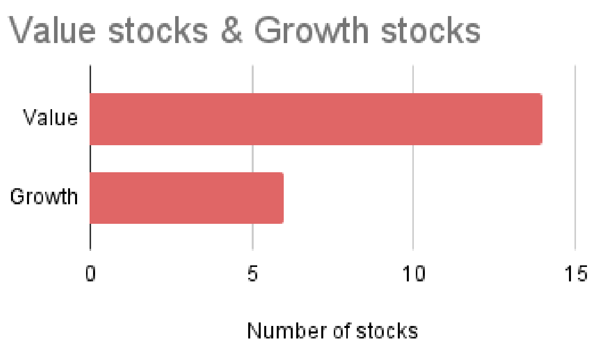
The table below shows how the 20 stocks in the portfolio are distributed across different beta levels



Beta Level	Number of Stocks
High (>1.0)	12
Medium (0.5 to 1.0)	5
Low (<0.5)	3

Stocks distributed across beta levels

This distribution shows that the portfolio has a slight overweighting to high-beta stocks. However, this is balanced by the inclusion of medium-beta and low-beta stocks, which helps to reduce the portfolio's overall volatility.



Characteristic	Value Stocks	Growth Stocks
P/E ratio	Low	High
P/B ratio	Low	High
Dividend yield	High	Low
Earnings growth	Slow	Fast

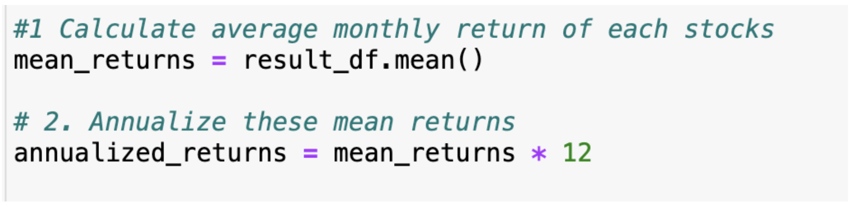
- **Value stocks** are shares of companies that are considered to be undervalued relative to their intrinsic worth. They typically have low price-to-earnings (P/E) ratios and low price-to-book (P/B) ratios. Value stocks are often found in industries that are out of favor with investors, such as energy and manufacturing.
- **Growth stocks** are shares of companies that are expected to grow their earnings at a faster rate than the overall market. They typically have high P/E ratios and high P/B ratios. Growth stocks are often found in industries that are experiencing rapid growth, such as technology and healthcare.

PYTHON:

Deliverable 2:

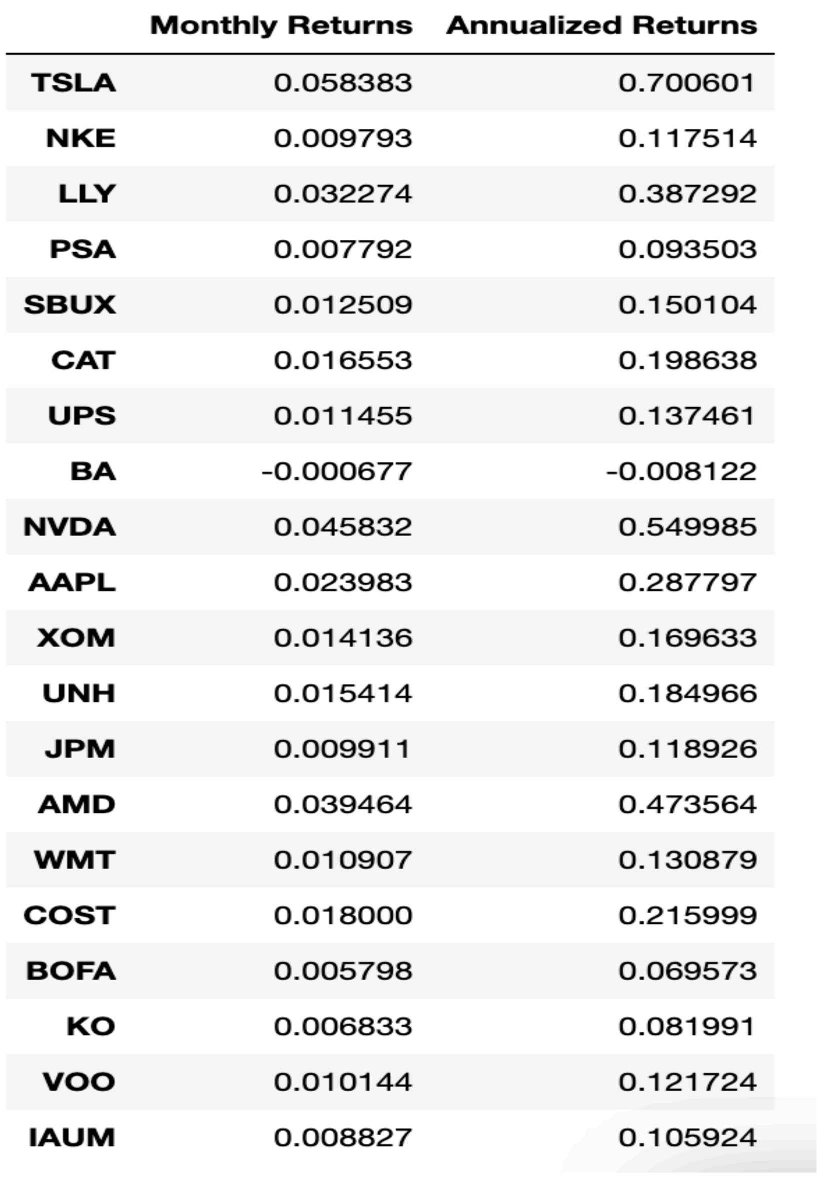
1,

Code:



```
#1 Calculate average monthly return of each stocks
mean_returns = result_df.mean()

# 2. Annualize these mean returns
annualized_returns = mean_returns * 12
```

Result:



	Monthly Returns	Annualized Returns
TSLA	0.058383	0.700601
NKE	0.009793	0.117514
LLY	0.032274	0.387292
PSA	0.007792	0.093503
SBUX	0.012509	0.150104
CAT	0.016553	0.198638
UPS	0.011455	0.137461
BA	-0.000677	-0.008122
NVDA	0.045832	0.549985
AAPL	0.023983	0.287797
XOM	0.014136	0.169633
UNH	0.015414	0.184966
JPM	0.009911	0.118926
AMD	0.039464	0.473564
WMT	0.010907	0.130879
COST	0.018000	0.215999
BOFA	0.005798	0.069573
KO	0.006833	0.081991
VOO	0.010144	0.121724
IAUM	0.008827	0.105924

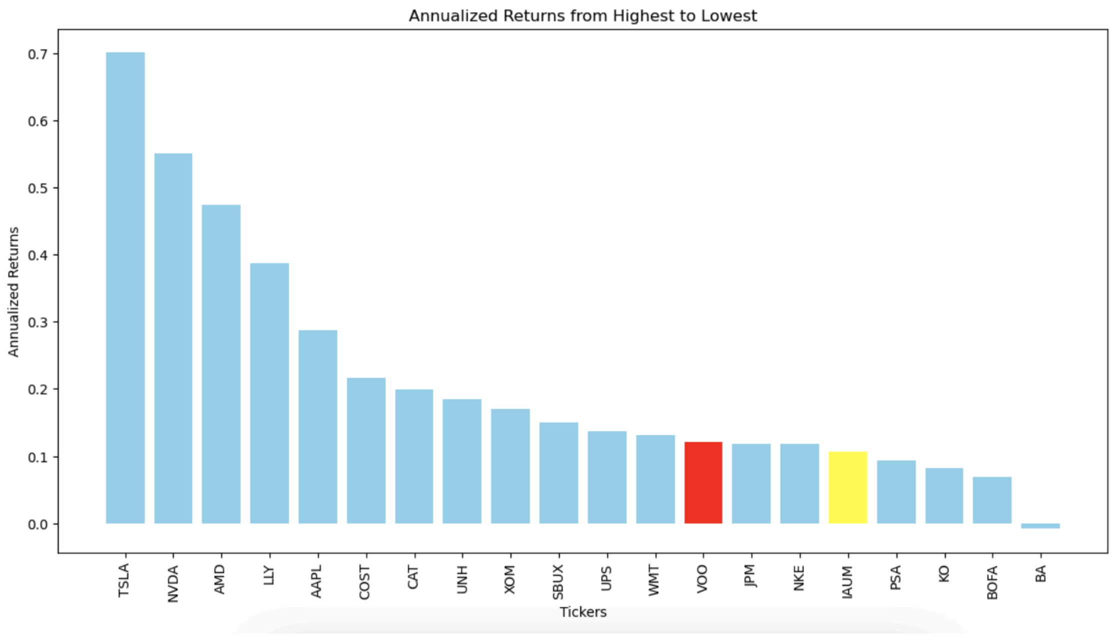


Fig 1. Annualized Returns from Highest to Lowest (Appendix 1)

Observations based on Annualized returns:

- TSLA is the best-performing stock, with an expected annualized return of 69.38% while the most underperforming stock is BA, with an expected annualized return of 0.53%.
- VOO's expected annualized monthly return is slightly higher than IAUM's, implying that the return of the stock market is anticipated to outperform gold over the long term.
- VOO, which tracks the S&P 500 index, represents a broad market ETF. The higher expected return of VOO could be attributed to the broader market's tendency to generate higher returns over extended periods compared to specific sectors or themes.
- Market conditions and individual stock performances can fluctuate, potentially impacting returns. Diversification across various asset classes and sectors can help mitigate risk and potentially enhance overall portfolio returns.

Deliverable 3: Variance-Covariance

Assume markets are efficient at least in a weak form. Market efficiency means that monthly returns are independent; that is, the correlation coefficient between returns in one month and returns in any other month is 0. We can multiply the entire variance-covariance matrix by 12 to get yearly covariance

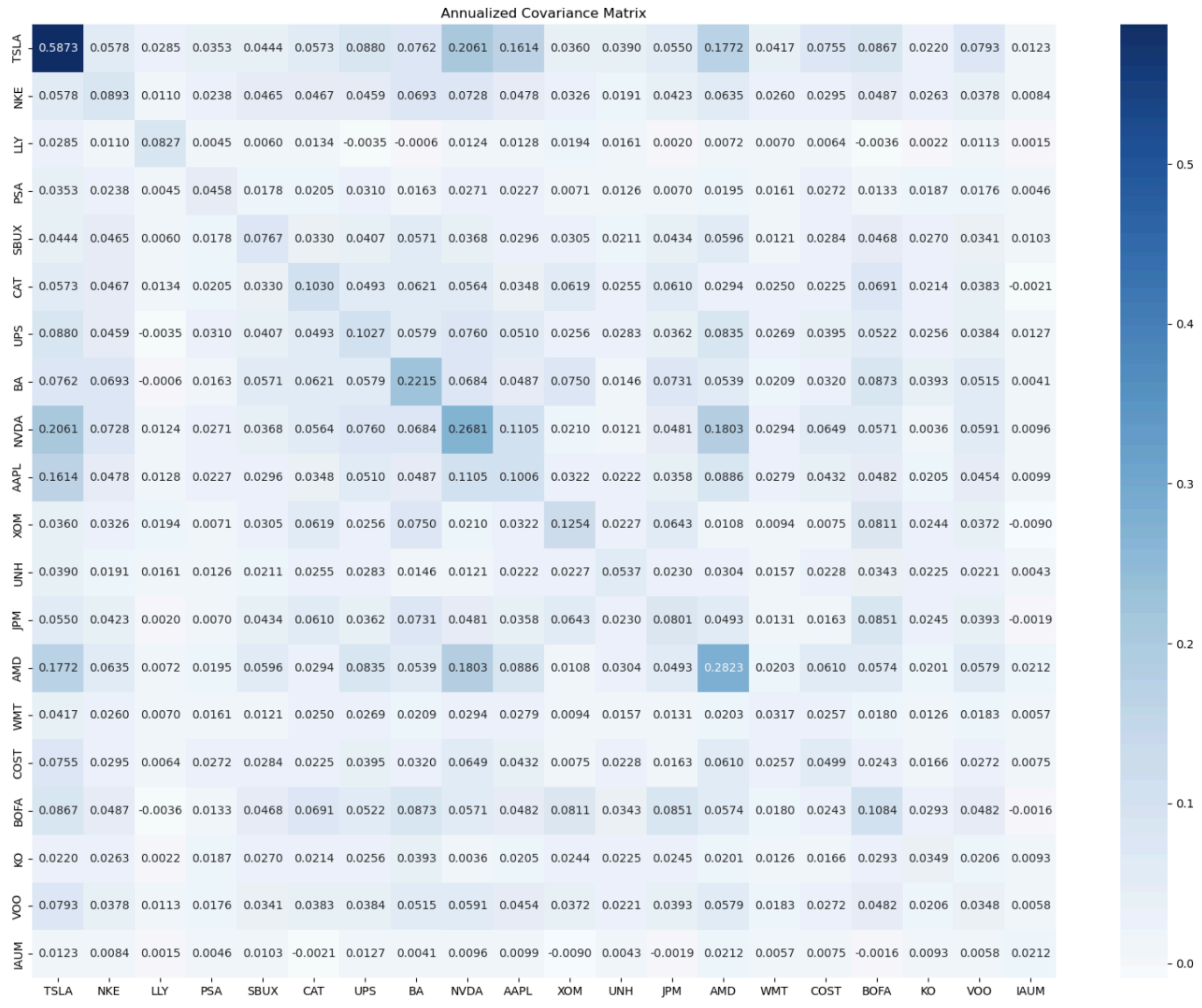
The covariance of a stock with itself is essentially the variance of that stock's returns

Code

```
# 1: Calculate covariance matrix
covariance_table = result_df.cov()
```

```
# 2 Annualized
cov_matrix = result_df.cov() * 12
```

Result:



	TSLA	NKE	LLY	PSA	SBUX	CAT	UPS	BA	NVDA	AAPL	XOM	UNH	JPM	AMD	WMT	COST	BOFA	KO	VOO	IAUM
TSLA	0.5873	0.0578	0.0285	0.0353	0.0444	0.0573	0.0880	0.0762	0.2061	0.1614	0.0360	0.0390	0.0550	0.1772	0.0417	0.0755	0.0867	0.0220	0.0793	0.0123
NKE	0.0578	0.0893	0.0110	0.0238	0.0465	0.0467	0.0459	0.0693	0.0728	0.0478	0.0326	0.0191	0.0423	0.0635	0.0260	0.0295	0.0487	0.0263	0.0378	0.0084
LLY	0.0285	0.0110	0.0827	0.0045	0.0060	0.0134	-0.0035	-0.0006	0.0124	0.0128	0.0194	0.0161	0.0020	0.0072	0.0070	0.0064	-0.0036	0.0022	0.0113	0.0015
PSA	0.0353	0.0238	0.0045	0.0458	0.0178	0.0205	0.0310	0.0163	0.0271	0.0227	0.0071	0.0126	0.0070	0.0195	0.0161	0.0272	0.0133	0.0187	0.0176	0.0046
SBUX	0.0444	0.0465	0.0060	0.0178	0.0767	0.0330	0.0407	0.0571	0.0368	0.0296	0.0305	0.0211	0.0434	0.0596	0.0121	0.0284	0.0468	0.0270	0.0341	0.0103
CAT	0.0573	0.0467	0.0134	0.0205	0.0330	0.1030	0.0493	0.0621	0.0564	0.0348	0.0619	0.0255	0.0610	0.0294	0.0250	0.0225	0.0691	0.0214	0.0383	-0.0021
UPS	0.0880	0.0459	-0.0035	0.0310	0.0407	0.0493	0.1027	0.0579	0.0760	0.0510	0.0256	0.0283	0.0362	0.0835	0.0269	0.0395	0.0522	0.0256	0.0384	0.0127
BA	0.0762	0.0693	-0.0006	0.0163	0.0571	0.0621	0.0579	0.2215	0.0684	0.0487	0.0750	0.0146	0.0731	0.0539	0.0209	0.0320	0.0873	0.0393	0.0515	0.0041
NVDA	0.2061	0.0728	0.0124	0.0271	0.0368	0.0564	0.0760	0.0684	0.2681	0.1105	0.0210	0.0121	0.0481	0.1803	0.0294	0.0649	0.0571	0.0036	0.0591	0.0096
AAPL	0.1614	0.0478	0.0128	0.0227	0.0296	0.0348	0.0510	0.0487	0.1105	0.1006	0.0322	0.0222	0.0358	0.0886	0.0279	0.0432	0.0482	0.0205	0.0454	0.0099
XOM	0.0360	0.0326	0.0194	0.0071	0.0305	0.0619	0.0256	0.0750	0.0210	0.0322	0.1254	0.0227	0.0643	0.0108	0.0094	0.0075	0.0811	0.0244	0.0372	-0.0090
UNH	0.0390	0.0191	0.0161	0.0126	0.0211	0.0255	0.0283	0.0146	0.0121	0.0222	0.0227	0.0537	0.0230	0.0304	0.0157	0.0228	0.0343	0.0225	0.0221	0.0043
JPM	0.0550	0.0423	0.0020	0.0070	0.0434	0.0610	0.0362	0.0731	0.0481	0.0358	0.0643	0.0230	0.0801	0.0493	0.0131	0.0163	0.0851	0.0245	0.0393	-0.0019
AMD	0.1772	0.0635	0.0072	0.0195	0.0596	0.0294	0.0835	0.0539	0.1803	0.0886	0.0108	0.0304	0.0493	0.2823	0.0203	0.0610	0.0574	0.0201	0.0579	0.0212
WMT	0.0417	0.0260	0.0070	0.0161	0.0121	0.0250	0.0269	0.0209	0.0294	0.0279	0.0094	0.0157	0.0131	0.0203	0.0317	0.0257	0.0180	0.0126	0.0183	0.0057
COST	0.0755	0.0295	0.0064	0.0272	0.0284	0.0225	0.0395	0.0320	0.0649	0.0432	0.0075	0.0228	0.0163	0.0610	0.0257	0.0499	0.0243	0.0166	0.0272	0.0075
BOFA	0.0867	0.0487	-0.0036	0.0133	0.0468	0.0691	0.0522	0.0873	0.0571	0.0482	0.0811	0.0343	0.0851	0.0574	0.0180	0.0243	0.1084	0.0293	0.0482	-0.0016
KO	0.0220	0.0263	0.0022	0.0187	0.0270	0.0214	0.0256	0.0393	0.0036	0.0205	0.0244	0.0225	0.0245	0.0201	0.0126	0.0166	0.0293	0.0349	0.0206	0.0093
VOO	0.0793	0.0378	0.0113	0.0176	0.0341	0.0383	0.0384	0.0515	0.0591	0.0454	0.0372	0.0221	0.0393	0.0579	0.0183	0.0272	0.0482	0.0206	0.0348	0.0058
IAUM	0.0123	0.0084	0.0015	0.0046	0.0103	-0.0021	0.0127	0.0041	0.0096	0.0099	-0.0090	0.0043	-0.0019	0.0212	0.0057	0.0075	-0.0016	0.0093	0.0058	0.0212

Fig 2. Covariance Matrix Heatmap (Appendix 2)

Observations from Annualized Variance-Covariance matrix:

The **variance** of TSLA is the highest, followed by AMD, NVDA, and BA. This suggests that these stocks are more volatile than the other stocks in the portfolio.

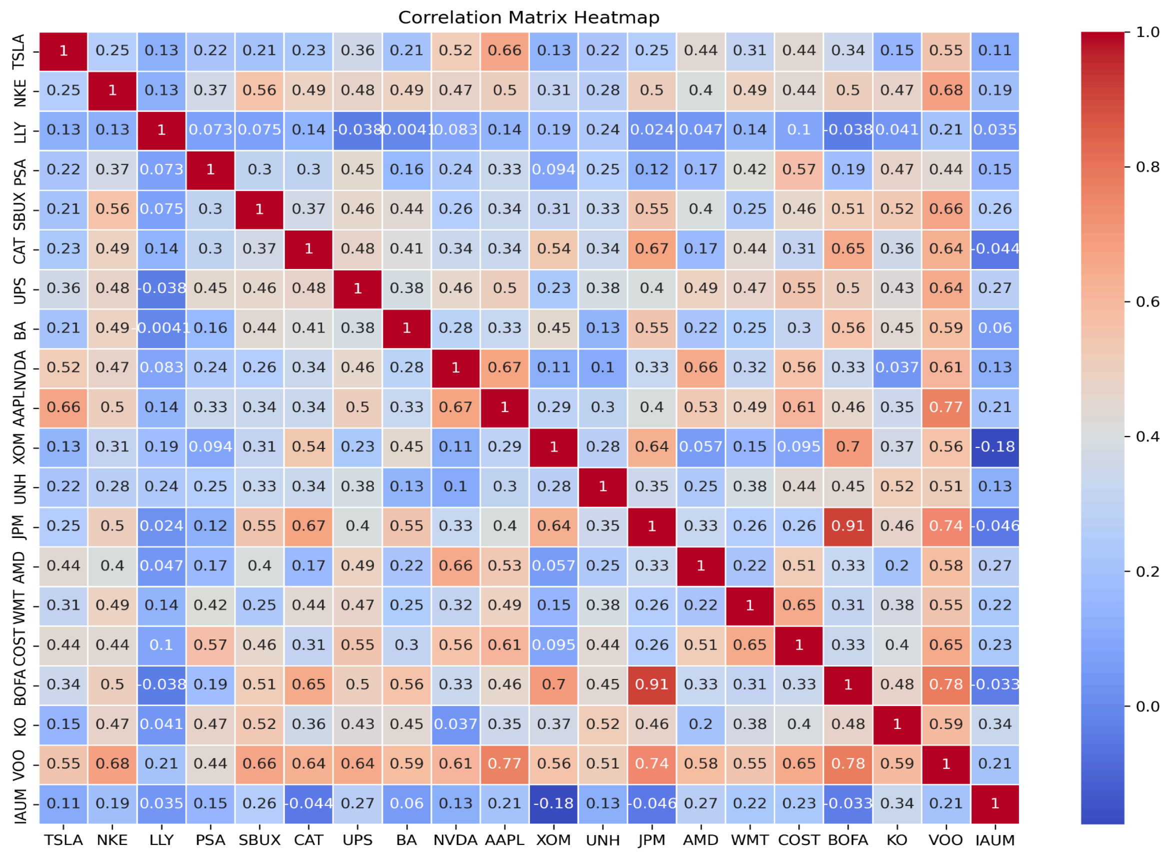
The **variance** of IAUM is the lowest. IAUM (which is gold) is still the best safe asset with its low variance, followed by Walmart, and VOO, and KO

The **covariance** between NVDA and TSLA is positive followed by AMD-AAPL, AMD-TSLA, and AAPL-TSLA, suggesting that these two stocks tend to move in the same direction.

The **covariance** between LLY and BA is negative, suggesting that these two stocks tend to move in opposite directions. This is likely due to the fact that LLY is a pharmaceutical company and BA is an airline. Similarly, IAUM has a negative correlation with CAT, XOM, and JPM.

<u>Deliverable 4: Correlation matrix</u>

Code:
```
correlation_table = result_df.corr()
```



	TSLA	NKE	LLY	PSA	SBUX	CAT	UPS	BA	NVDA	AAPL	XOM	UNH	JPM	AMD	WMT	COST	BOFA	KO	VOO	IAUM
TSLA	1	0.25	0.13	0.22	0.21	0.23	0.36	0.21	0.52	0.66	0.13	0.22	0.25	0.44	0.31	0.44	0.34	0.15	0.55	0.11
NKE	0.25	1	0.13	0.37	0.56	0.49	0.48	0.49	0.47	0.5	0.31	0.28	0.5	0.4	0.49	0.44	0.5	0.47	0.68	0.19
LLY	0.13	0.13	1	0.073	0.075	0.14	-0.038	-0.0041	0.083	0.14	0.19	0.24	0.024	0.047	0.14	0.1	-0.038	0.041	0.21	0.035
PSA	0.22	0.37	0.073	1	0.3	0.3	0.45	0.16	0.24	0.33	0.094	0.25	0.12	0.17	0.42	0.57	0.19	0.47	0.44	0.15
SBUX	0.21	0.56	0.075	0.3	1	0.37	0.46	0.44	0.26	0.34	0.31	0.33	0.55	0.4	0.25	0.46	0.51	0.52	0.66	0.26
CAT	0.23	0.49	0.14	0.3	0.37	1	0.48	0.41	0.34	0.34	0.54	0.34	0.67	0.17	0.44	0.31	0.65	0.36	0.64	-0.044
UPS	0.36	0.48	-0.038	0.45	0.46	0.48	1	0.38	0.46	0.5	0.23	0.38	0.4	0.49	0.47	0.55	0.5	0.43	0.64	0.27
BA	0.21	0.49	-0.0041	0.16	0.44	0.41	0.38	1	0.28	0.33	0.45	0.13	0.55	0.22	0.25	0.3	0.56	0.45	0.59	0.06
NVDA	0.52	0.47	0.083	0.24	0.26	0.34	0.46	0.28	1	0.67	0.11	0.1	0.33	0.66	0.32	0.56	0.33	0.037	0.61	0.13
AAPL	0.66	0.5	0.14	0.33	0.34	0.34	0.5	0.33	0.67	1	0.29	0.3	0.4	0.53	0.49	0.61	0.46	0.35	0.77	0.21
XOM	0.13	0.31	0.19	0.094	0.31	0.54	0.23	0.45	0.11	0.29	1	0.28	0.64	0.057	0.15	0.095	0.7	0.37	0.56	-0.18
UNH	0.22	0.28	0.24	0.25	0.33	0.34	0.38	0.13	0.1	0.3	0.28	1	0.35	0.25	0.38	0.44	0.45	0.52	0.51	0.13
JPM	0.25	0.5	0.024	0.12	0.55	0.67	0.4	0.55	0.33	0.4	0.64	0.35	1	0.33	0.26	0.26	0.91	0.46	0.74	-0.046
AMD	0.44	0.4	0.047	0.17	0.4	0.17	0.49	0.22	0.66	0.53	0.057	0.25	0.33	1	0.22	0.51	0.33	0.2	0.58	0.27
WMT	0.31	0.49	0.14	0.42	0.25	0.44	0.47	0.25	0.32	0.49	0.15	0.38	0.26	0.22	1	0.65	0.31	0.38	0.55	0.22
COST	0.44	0.44	0.1	0.57	0.46	0.31	0.55	0.3	0.56	0.61	0.095	0.44	0.26	0.51	0.65	1	0.33	0.4	0.65	0.23
BOFA	0.34	0.5	-0.038	0.19	0.51	0.65	0.5	0.56	0.33	0.46	0.7	0.45	0.91	0.33	0.31	0.33	1	0.48	0.78	-0.033
KO	0.15	0.47	0.041	0.47	0.52	0.36	0.43	0.45	0.037	0.35	0.37	0.52	0.46	0.2	0.38	0.4	0.48	1	0.59	0.34
VOO	0.55	0.68	0.21	0.44	0.66	0.64	0.64	0.59	0.61	0.77	0.56	0.51	0.74	0.58	0.55	0.65	0.78	0.59	1	0.21
IAUM	0.11	0.19	0.035	0.15	0.26	-0.044	0.27	0.06	0.13	0.21	-0.18	0.13	-0.046	0.27	0.22	0.23	-0.033	0.34	0.21	1

Fig 3. Correlation Matrix Heatmap (Appendix 3)

Interpreting Correlation Coefficients

Correlation coefficients range from -1 to 1, indicating the degree of association between two securities. A value of 1 indicates a perfect positive correlation, meaning that as one stock increases, the other also increases proportionally. Conversely, a value of -1 indicates a perfect negative correlation, meaning that as one stock increases, the other decreases proportionally. A value of 0 indicates no correlation, implying that the two variables are not linearly related.

Observations from the Correlation Matrix

- **Strong Positive Correlations:**

JPM - BOFA and exhibit a strong positive correlation, suggesting that banking has a closely related correlation. VOO has a strong positive correlation with BOFA, JPM, and AAPL, possibly because of their huge market capitalization, or their link to the overall economy health

- **Weak Positive Correlations:**

NKE has a very low positive correlation with BOFA, AMD, JPM, and AAPL, suggesting that NKE can be used to diversify against those stocks. Similarly, COSTCO with LLY and BA has weak positive correlations.

- **Strong Negative Correlations:**

XOM has a strong negative correlation with IAUM, implying that their movements tend to diverge. Gold is the best safe haven for XOM

- **Weak Negative Correlations:**

LLY exhibits weak negative correlations with both BOFA and UPS, while IAUM exhibits weak negative correlations with BOFA, JPM, and CAT. Weak negative correlation offer a better chance to diversify portfolios than positive correlation

4. Optimal Portfolio Selection

Deliverable 5: Construct the Minimum Variance Efficient Portfolio, trace the efficient frontier, and present it on the graph in the standard deviation vs. expected return space.

- Step 1: Establish initial weights and define functions to get portfolio variance. Define constraint for portfolios: total weight to equal 1. Calculate initial portfolio variance, standard deviation, and return.

$$\sum_{i=1}^{N} w_i = 1$$

$$Variance\ of\ Portfolio$$

$$\sigma_{pf}^2 = \omega \Omega \omega'$$

$$= [W_1\ ,\ \ W_2, \dots \dots \dots \dots \dots W_{20}] \times \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1,20}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{20,1}^2 & \cdots & \sigma_{20}^2 \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ \dots \\ \dots \\ \dots \\ W_{20} \end{bmatrix}$$

Initially we are assuming equal weights, i.e. $W_1, W_{2,} W_3, , , , , , , , , W_{20}$= 0.5

$$= [0.5\ \ , 0.5 \qquad \dots \dots \dots \dots \dots 0.5] \times \begin{bmatrix} \sigma_{TSLA}^2 & \cdots & \sigma_{IAUM,TSLA}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{TSLA,IAUM}^2 & \cdots & \sigma_{IAUM}^2 \end{bmatrix} \times \begin{bmatrix} 0.5 \\ 0.5 \\ \dots \\ \dots \\ \dots \\ 0.5 \end{bmatrix}$$

σ_{pf}^2 = 0.0401

Standard Deviation

$$\sigma_{pf} = \sqrt{\sigma_{pf}^2}$$

$$\sigma_{pf} = \sqrt{0.0401} = 0.2$$

Expected Returns

$$E[r_{pf}] = R\omega'$$

$$= [R_1\ ,\ \ R_2, \dots \dots \dots \dots \dots R_{20}] \times \begin{bmatrix} W_1 \\ W_2 \\ \cdots \\ \cdots \\ \cdots \\ W_{20} \end{bmatrix}$$

$$= [R_{TSLA}\ ,\ \ R_{NKE}, \dots \dots \dots \dots \dots R_{IAUM}] \times \begin{bmatrix} 0.5 \\ 0.5 \\ \cdots \\ \cdots \\ \cdots \\ 0.5 \end{bmatrix}$$

=0.2143

Step 2: Construct MVE portfolio

- Optimize portfolio to minimize portfolio variance, extract MVE weights, and calculate MVE expected return (E_RMVE). Mve_weights variable is the allocation that minimize MVE

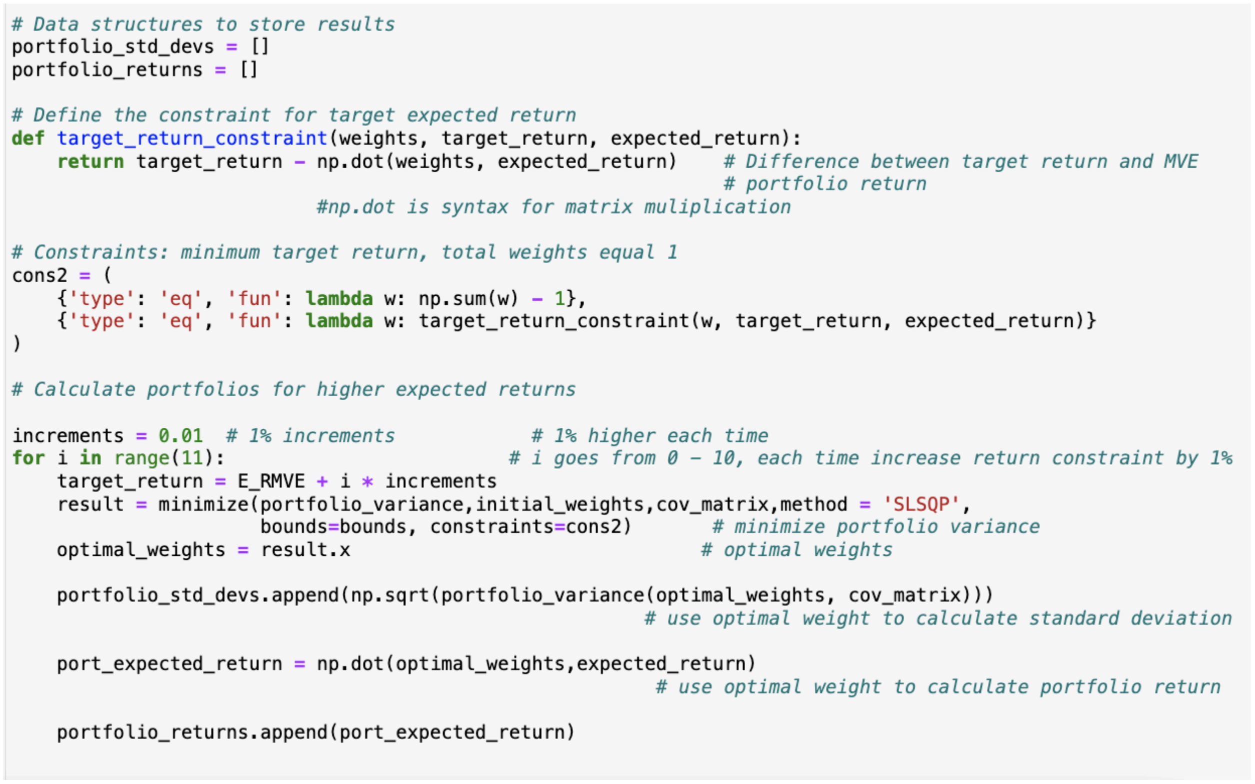
```
# Number of assets
N = len(expected_return)

# Initial guess equal weights
initial_weights = np.repeat(1/N, N)

# Define function to get portfolio variance
def portfolio_variance(weights, cov_matrix):
    return weights @ cov_matrix @ weights

# Constraints: sum of weight = 1
cons = ({'type': 'eq', 'fun': lambda w: np.sum(w) - 1})

# No short-selling constraints
bounds = [(-np.inf,np.inf) for i in range(N)]


# Optimization: MVE portfolio
solution = minimize(portfolio_variance, initial_weights, args = cov_matrix,
                    method='SLSQP', bounds=bounds, constraints=cons)

mve_weights = solution.x

# MVE portfolio expected return
E_RMVE = np.dot(mve_weights, expected_return)
```

Step 2: Adding return restraint and creating loop

- The loop is designed to incrementally construct portfolios with higher expected returns than the Minimum Variance Edge (MVE) portfolio. Starting from the return of the MVE, the loop increases the target return by 1% at each iteration (E_RMVE + 1*increment), creating a new portfolio at each step that is optimized to achieve this higher return with the lowest possible risk. By solving a constrained optimization problem to minimize variance, the script calculates the optimal weights for each new portfolio. Repeating this process over a range of expected returns traces out the efficient frontier, which represents the trade-off between risk and return for an investor's optimal portfolio choices
- This is commonly known as Mean-Variance Optimization (MVO) or the Markowitz optimization model in finance.
- Added expected return and standard deviation of each portfolio into data structure (potfolio_std_devs and portfolio_returns)
- This code goes from E_RMVE + 0% to E_RMVE + 10%

$$\sum_{i=1}^{N} w_i = 1$$

$$E[r_{pf}] = A,$$



```python
# Data structures to store results
portfolio_std_devs = []
portfolio_returns = []

# Define the constraint for target expected return
def target_return_constraint(weights, target_return, expected_return):
    return target_return - np.dot(weights, expected_return)    # Difference between target return and MVE
                                                               # portfolio return
                           #np.dot is syntax for matrix muliplication

# Constraints: minimum target return, total weights equal 1
cons2 = (
    {'type': 'eq', 'fun': lambda w: np.sum(w) - 1},
    {'type': 'eq', 'fun': lambda w: target_return_constraint(w, target_return, expected_return)}
)

# Calculate portfolios for higher expected returns

increments = 0.01  # 1% increments            # 1% higher each time
for i in range(11):                         # i goes from 0 - 10, each time increase return constraint by 1%
    target_return = E_RMVE + i * increments
    result = minimize(portfolio_variance,initial_weights,cov_matrix,method = 'SLSQP',
                      bounds=bounds, constraints=cons2)       # minimize portfolio variance
    optimal_weights = result.x                               # optimal weights

    portfolio_std_devs.append(np.sqrt(portfolio_variance(optimal_weights, cov_matrix)))
                                                        # use optimal weight to calculate standard deviation

    port_expected_return = np.dot(optimal_weights,expected_return)
                                                         # use optimal weight to calculate portfolio return

    portfolio_returns.append(port_expected_return)
```

<u>Deliverable 6:</u> Construct the tangency portfolio and add it to your graph of the efficient frontier.

- Function and constraints. We take the 10-year T-bonds yield as the risk-free rate, which is 5%
- Sharpe Ratio for the initial portfolio (where all weights are 0.05)

$$\frac{E[rpf] - rf}{\sigma_{pf}}$$

$$= \frac{0.2143 - 0.05}{0.2}$$

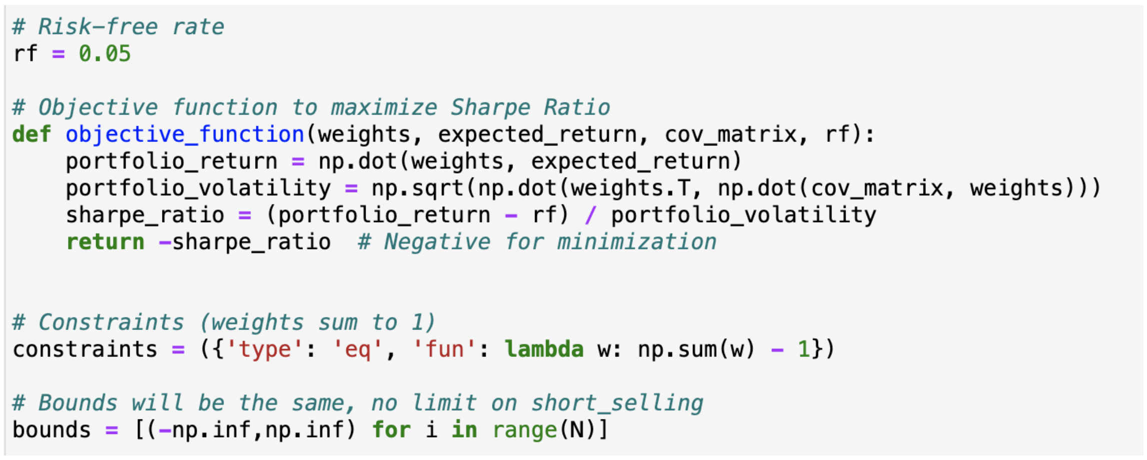
$$=0.82029$$

- Optimize to find tangent portfolio

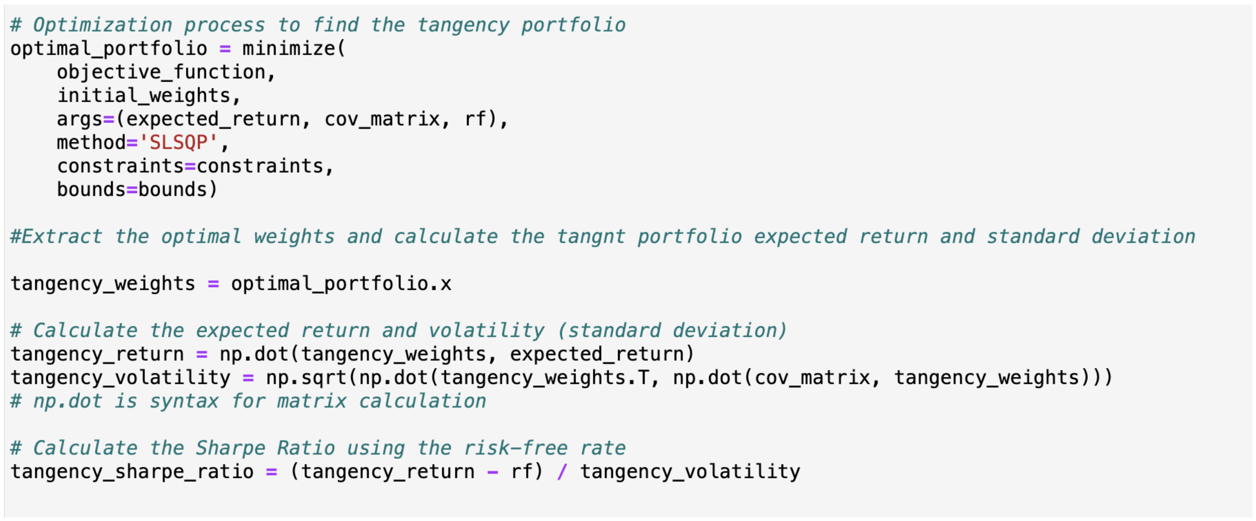
$$\max_{\omega} \frac{E[r_{pf}]-r_f}{\sigma_{pf}}$$

subject to:

$$\sum_{i=1}^{N} w_i = 1$$



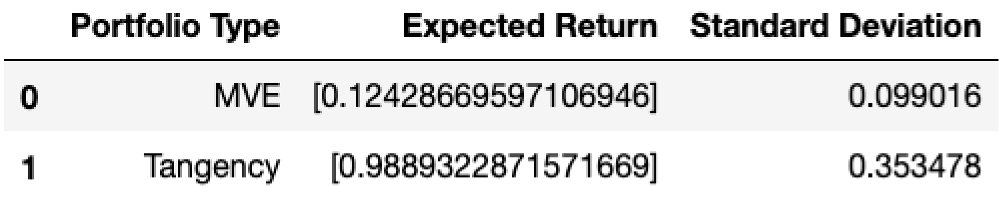
```python
# Risk-free rate
rf = 0.05

# Objective function to maximize Sharpe Ratio
def objective_function(weights, expected_return, cov_matrix, rf):
    portfolio_return = np.dot(weights, expected_return)
    portfolio_volatility = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))
    sharpe_ratio = (portfolio_return - rf) / portfolio_volatility
    return -sharpe_ratio  # Negative for minimization


# Constraints (weights sum to 1)
constraints = ({'type': 'eq', 'fun': lambda w: np.sum(w) - 1})

# Bounds will be the same, no limit on short_selling
bounds = [(-np.inf,np.inf) for i in range(N)]
```



```python
# Optimization process to find the tangency portfolio
optimal_portfolio = minimize(
    objective_function,
    initial_weights,
    args=(expected_return, cov_matrix, rf),
    method='SLSQP',
    constraints=constraints,
    bounds=bounds)

#Extract the optimal weights and calculate the tangnt portfolio expected return and standard deviation

tangency_weights = optimal_portfolio.x

# Calculate the expected return and volatility (standard deviation)
tangency_return = np.dot(tangency_weights, expected_return)
tangency_volatility = np.sqrt(np.dot(tangency_weights.T, np.dot(cov_matrix, tangency_weights)))
# np.dot is syntax for matrix calculation

# Calculate the Sharpe Ratio using the risk-free rate
tangency_sharpe_ratio = (tangency_return - rf) / tangency_volatility
```

<u>Deliverable 7:</u>

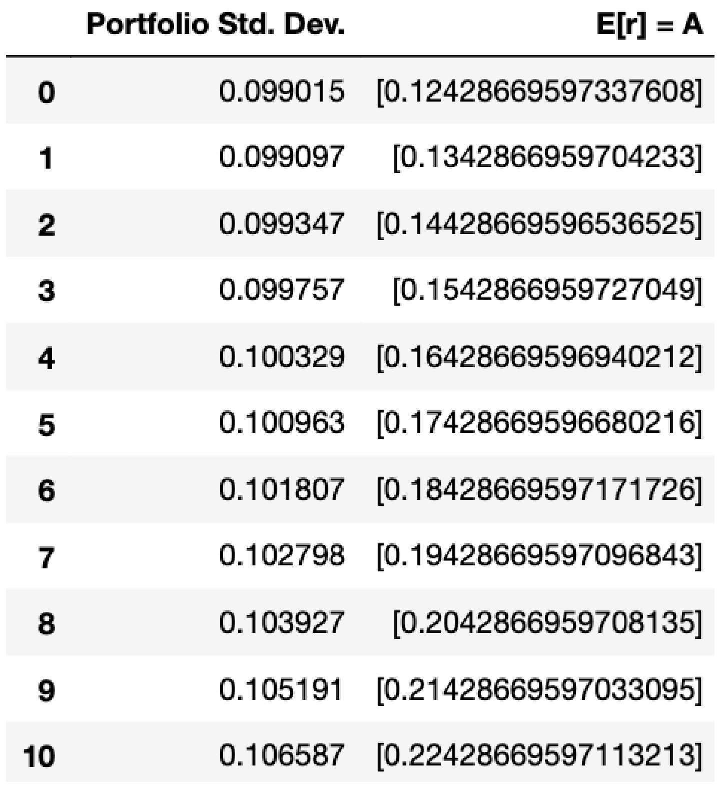
1. Present a table that contains the expected return and standard deviation of the MVE portfolio and the tangency portfolio.



	Portfolio Type	Expected Return	Standard Deviation
0	MVE	[0.12428669597106946]	0.099016
1	Tangency	[0.9889322871571669]	0.353478

2. Present a table that contains expected returns and standard deviation of all efficient portfolios.

The index runs from 0 - 10. At index 0 is the MVE portfolio, the expected return of the MVE portfolio is 12,42%, Index 1 to 10 represents 10 portfolios with return restraint, each time increasing the expected return by 1% relative to the previous portfolio



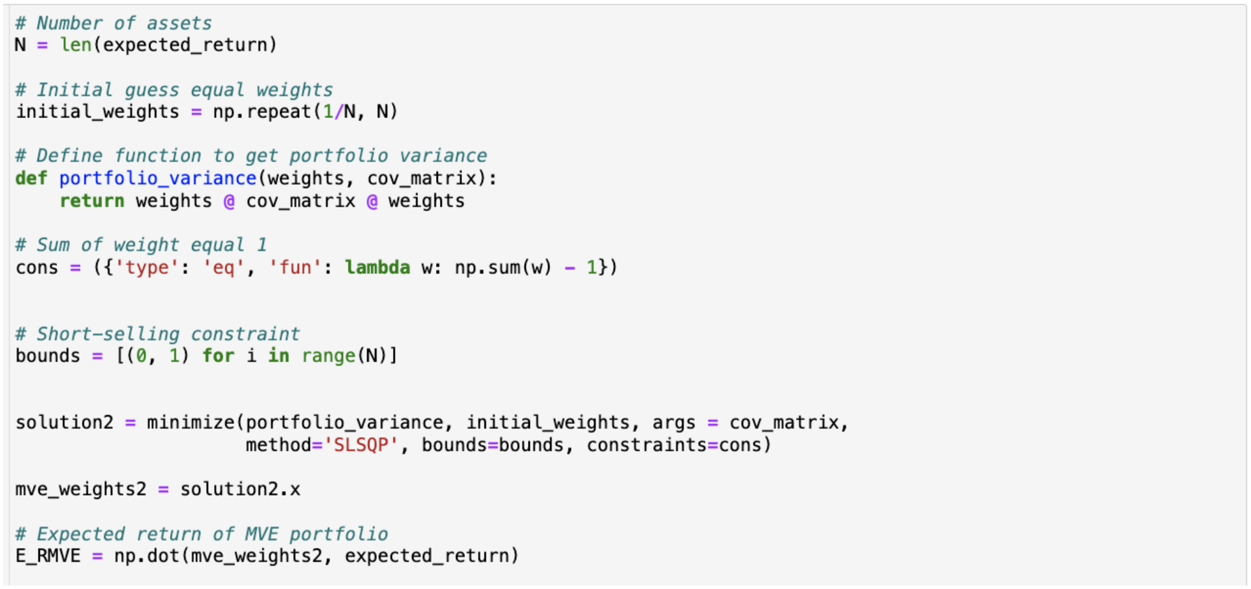
	Portfolio Std. Dev.	E[r] = A
0	0.099015	[0.12428669597337608]
1	0.099097	[0.1342866959704233]
2	0.099347	[0.14428669596536525]
3	0.099757	[0.1542866959727049]
4	0.100329	[0.16428669596940212]
5	0.100963	[0.17428669596680216]
6	0.101807	[0.18428669597171726]
7	0.102798	[0.19428669597096843]
8	0.103927	[0.2042866959708135]
9	0.105191	[0.21428669597033095]
10	0.106587	[0.22428669597113213]

<u>Deliverable 8:</u>

1. Short-sell constraints
    a. Efficient frontier

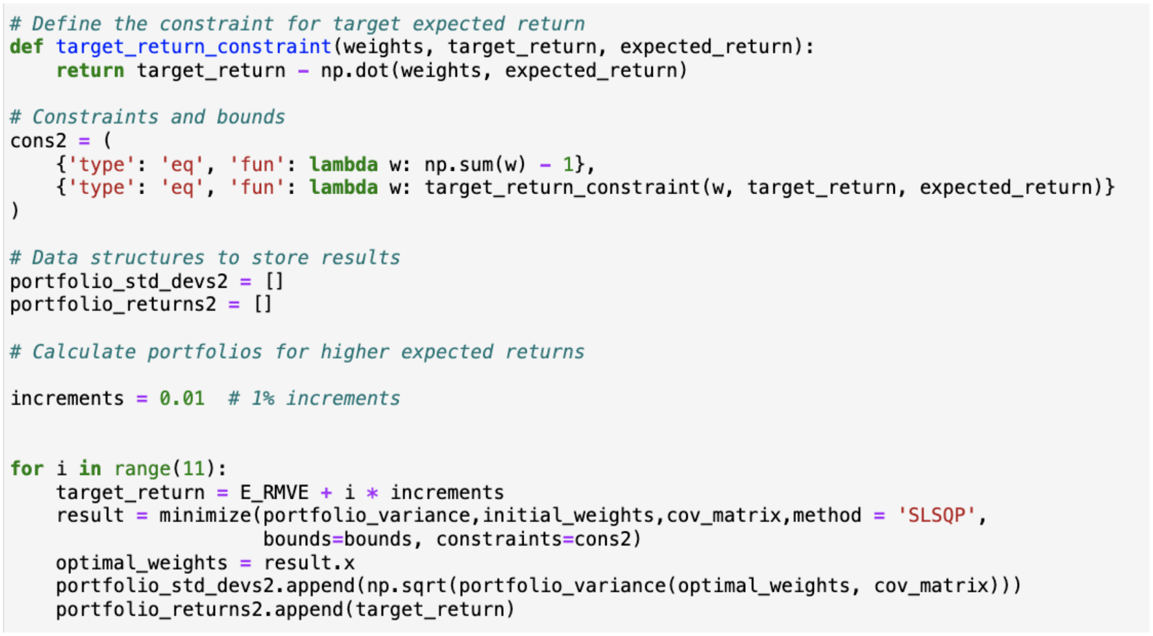
Step 1:

The only thing that differs from the code in deliverable 5 is bound. Here, we restrict the weight to be between 0 and 1, meaning no short sellings.



```python
# Number of assets
N = len(expected_return)

# Initial guess equal weights
initial_weights = np.repeat(1/N, N)

# Define function to get portfolio variance
def portfolio_variance(weights, cov_matrix):
    return weights @ cov_matrix @ weights

# Sum of weight equal 1
cons = ({'type': 'eq', 'fun': lambda w: np.sum(w) - 1})


# Short-selling constraint
bounds = [(0, 1) for i in range(N)]


solution2 = minimize(portfolio_variance, initial_weights, args = cov_matrix,
                     method='SLSQP', bounds=bounds, constraints=cons)

mve_weights2 = solution2.x

# Expected return of MVE portfolio
E_RMVE = np.dot(mve_weights2, expected_return)
```

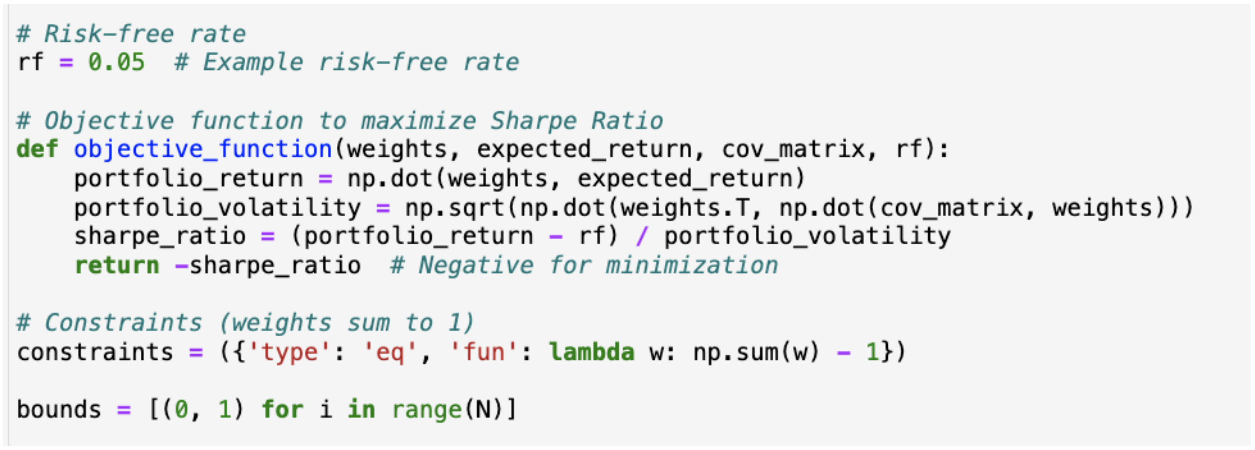
Step 2:



```python
# Define the constraint for target expected return
def target_return_constraint(weights, target_return, expected_return):
    return target_return - np.dot(weights, expected_return)

# Constraints and bounds
cons2 = (
    {'type': 'eq', 'fun': lambda w: np.sum(w) - 1},
    {'type': 'eq', 'fun': lambda w: target_return_constraint(w, target_return, expected_return)}
)

# Data structures to store results
portfolio_std_devs2 = []
portfolio_returns2 = []

# Calculate portfolios for higher expected returns

increments = 0.01  # 1% increments


for i in range(11):
    target_return = E_RMVE + i * increments
    result = minimize(portfolio_variance,initial_weights,cov_matrix,method = 'SLSQP',
                      bounds=bounds, constraints=cons2)
    optimal_weights = result.x
    portfolio_std_devs2.append(np.sqrt(portfolio_variance(optimal_weights, cov_matrix)))
    portfolio_returns2.append(target_return)
```

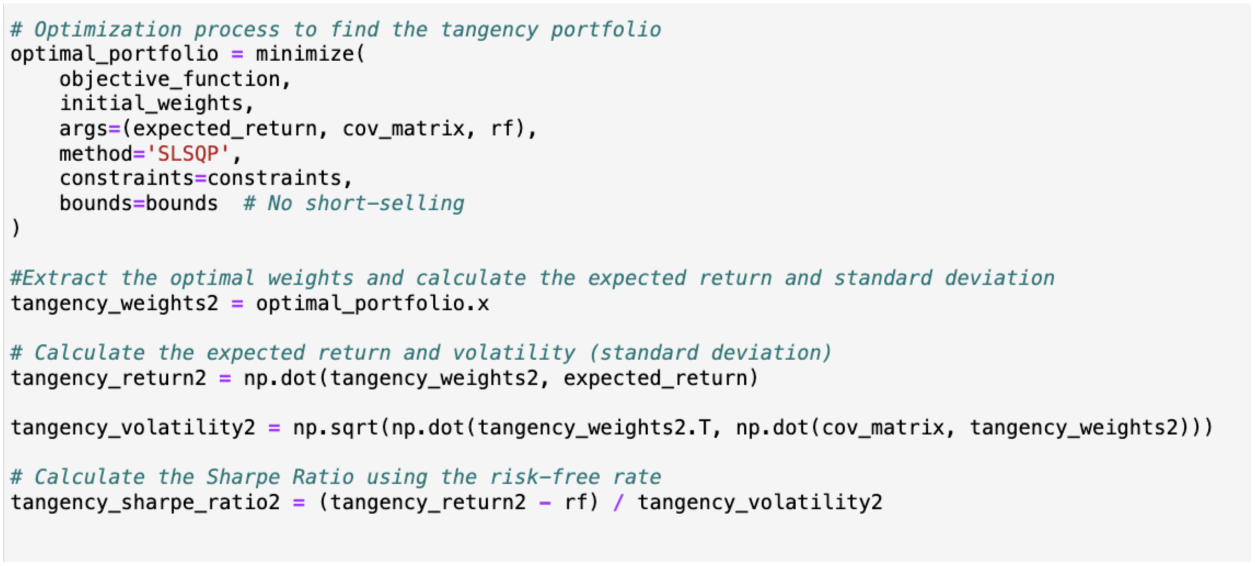
b. Tangent portfolio

- Function and Constraint



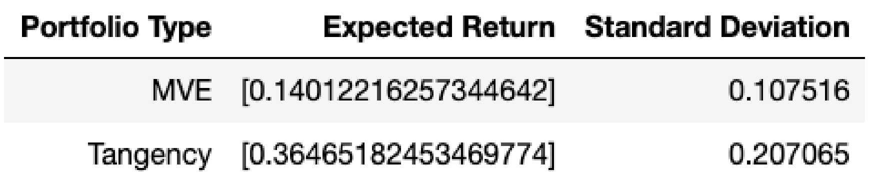
```
# Risk-free rate
rf = 0.05  # Example risk-free rate

# Objective function to maximize Sharpe Ratio
def objective_function(weights, expected_return, cov_matrix, rf):
    portfolio_return = np.dot(weights, expected_return)
    portfolio_volatility = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))
    sharpe_ratio = (portfolio_return - rf) / portfolio_volatility
    return -sharpe_ratio  # Negative for minimization

# Constraints (weights sum to 1)
constraints = ({'type': 'eq', 'fun': lambda w: np.sum(w) - 1})

bounds = [(0, 1) for i in range(N)]
```

- Optimize



```
# Optimization process to find the tangency portfolio
optimal_portfolio = minimize(
    objective_function,
    initial_weights,
    args=(expected_return, cov_matrix, rf),
    method='SLSQP',
    constraints=constraints,
    bounds=bounds  # No short-selling
)

#Extract the optimal weights and calculate the expected return and standard deviation
tangency_weights2 = optimal_portfolio.x

# Calculate the expected return and volatility (standard deviation)
tangency_return2 = np.dot(tangency_weights2, expected_return)

tangency_volatility2 = np.sqrt(np.dot(tangency_weights2.T, np.dot(cov_matrix, tangency_weights2)))

# Calculate the Sharpe Ratio using the risk-free rate
tangency_sharpe_ratio2 = (tangency_return2 - rf) / tangency_volatility2
```

2. Present a table which contains expected returns and standard deviations of the MVE portfolio and the tangency portfolio obtained under the short sale restrictions.



Portfolio Type	Expected Return	Standard Deviation
MVE	[0.14012216257344642]	0.107516
Tangency	[0.36465182453469774]	0.207065

3. Since we have a loop, we can easily adjust the number of portfolios to any number we want. To better show the effect of short-sell constraints on efficient frontiers, we are creating 100 portfolios, going from E_RMVE - 5% to E_RMVE + 95%

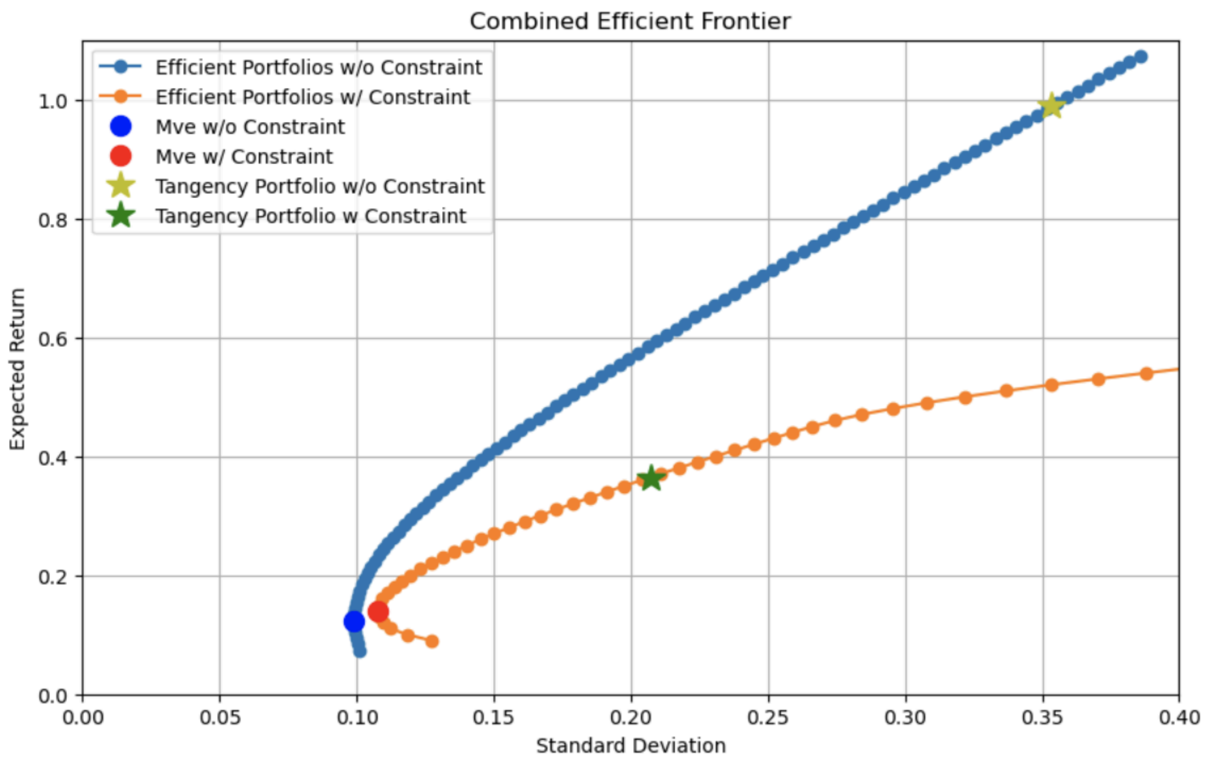


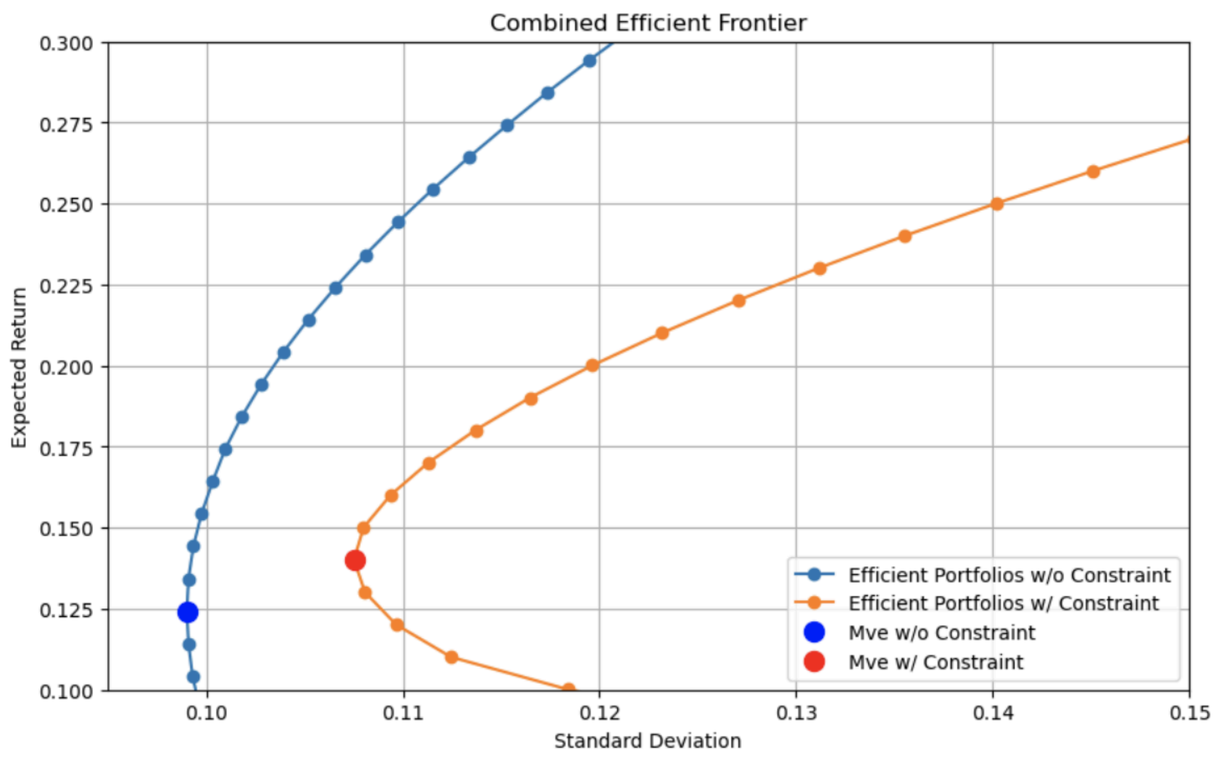
Fig 4. Combined Efficient Frontier (MVE and Tangent Portfolio) (Appendix 4)

a. Efficient Frontiers:

General Observation: The efficient frontier without constraints is positioned higher and to the left of the constrained frontier, indicating that for a given level of risk, an unconstrained portfolio can achieve a higher return.

Impact of Constraints: The imposition of short sale constraints shifts the entire efficient frontier downwards and to the right. This suggests that the investment opportunities are less efficient overall, as investors are unable to leverage the full range of investment strategies, such as taking short positions to hedge or to take advantage of overvaluations.

b. Minimum Variance Portfolios (MVE):



Short-sell Constraint	Expected Return	Standard Deviation
No	[0.12428669597106946]	0.099016
Yes	[0.14012216257344642]	0.107516

Without Short Sale Constraints (Blue Dot): The MVE portfolio without constraints has a lower standard deviation, indicating it is less risky. This suggests that when short sales are allowed, investors can construct a portfolio that is more effective at minimizing risk.

With Short Sale Constraints (Red Dot): The constrained MVE portfolio has a higher standard deviation, which implies that the inability to short-sell securities leads to a less risk-efficient portfolio. Investors who hold only long positions may not optimize their portfolio in terms of risk diversification.

c. Tangent Portfolios:

Short-sell Constraint	Expected Return	Standard Deviation	Sharpe Ratio
No	[0.9889322871571669]	0.353478	[2.6562647632841907]
Yes	[0.36465182453469774]	0.207065	[1.5195793300686804]

A tangency portfolio is a portfolio that touches the efficient frontier at the point with the highest Sharpe ratio

Without Short Sale Constraints (Yellow star): The tangency portfolio is further up the blue line (efficient frontier without constraint) and would offer a higher return for a given level of risk than the constrained case. This is due to the greater flexibility in weight allocation, including the ability to employ hedging positions.

With Short Sale Constraints (Green star): The tangency portfolio on the orange line (efficient frontier with constraint) would be limited in its position sizing and unable to fully capitalize on negative weightings to maximize the Sharpe ratio.

CONCLUSION

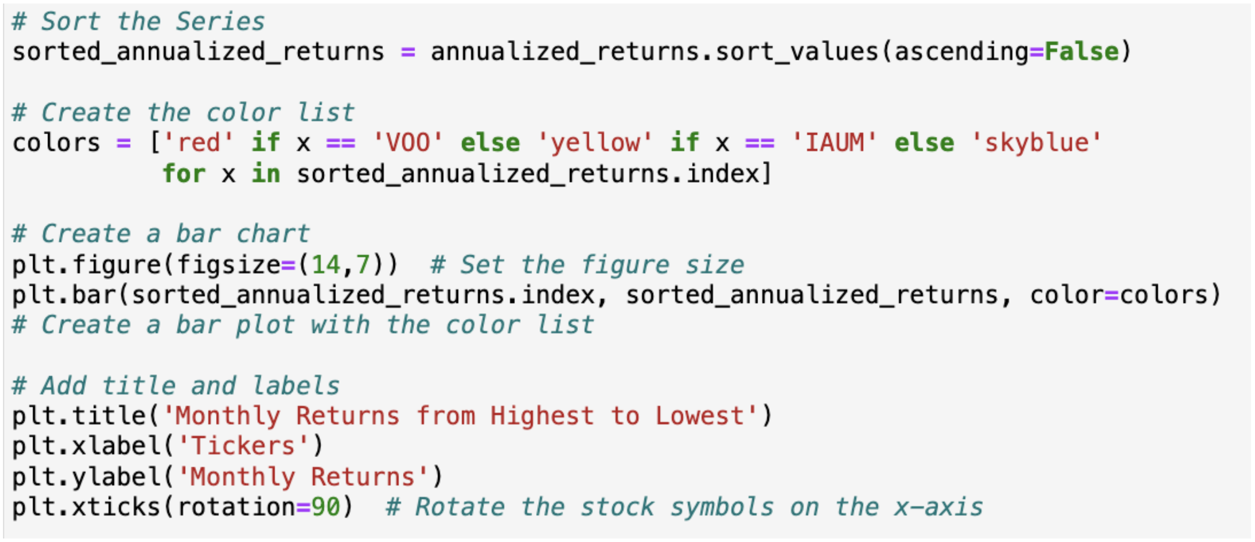
In conclusion, when investors face no constraints, they have a wider array of investment opportunities, including both long and short positions, which can potentially result in higher returns relative to the level of risk involved.

However, when constraints are in place, the range of investment options becomes limited. Investors may have to choose between assuming higher risk to achieve the same expected return or settling for a lower return at a given risk level compared to unconstrained portfolios.

Essentially, short-sale constraints limit an investor's ability to fully diversify and hedge their portfolio, leading to a less favorable balance between risk and return, as evident by the shift in the efficient frontier. This, in turn, has implications for both the Minimum Variance Portfolio (MVP) and the tangency portfolio, potentially resulting in a less-than-optimal allocation of assets compared to an unconstrained investment environment.

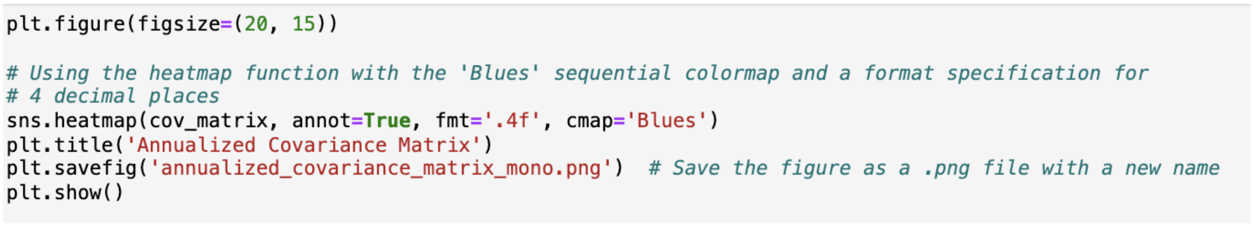
APPENDIX: CODE FOR VISUALIZATION

1. Bar graph



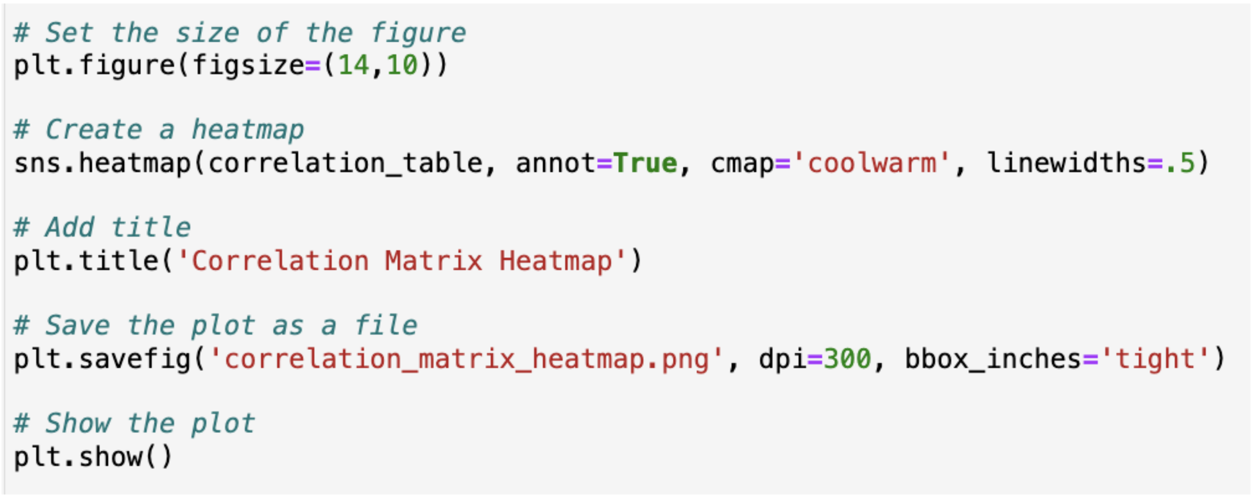
```
# Sort the Series
sorted_annualized_returns = annualized_returns.sort_values(ascending=False)

# Create the color list
colors = ['red' if x == 'VOO' else 'yellow' if x == 'IAUM' else 'skyblue'
          for x in sorted_annualized_returns.index]

# Create a bar chart
plt.figure(figsize=(14,7))  # Set the figure size
plt.bar(sorted_annualized_returns.index, sorted_annualized_returns, color=colors)
# Create a bar plot with the color list

# Add title and labels
plt.title('Monthly Returns from Highest to Lowest')
plt.xlabel('Tickers')
plt.ylabel('Monthly Returns')
plt.xticks(rotation=90)  # Rotate the stock symbols on the x-axis
```

2. Covariance heat map



```
plt.figure(figsize=(20, 15))

# Using the heatmap function with the 'Blues' sequential colormap and a format specification for
# 4 decimal places
sns.heatmap(cov_matrix, annot=True, fmt='.4f', cmap='Blues')
plt.title('Annualized Covariance Matrix')
plt.savefig('annualized_covariance_matrix_mono.png')  # Save the figure as a .png file with a new name
plt.show()
```

3. Correlation heat map



```
# Set the size of the figure
plt.figure(figsize=(14,10))

# Create a heatmap
sns.heatmap(correlation_table, annot=True, cmap='coolwarm', linewidths=.5)

# Add title
plt.title('Correlation Matrix Heatmap')

# Save the plot as a file
plt.savefig('correlation_matrix_heatmap.png', dpi=300, bbox_inches='tight')

# Show the plot
plt.show()
```

4. Plot standard deviation and return

```python
fig, ax = plt.subplots(figsize=(10,6))
# Plot the efficient frontier without constraint
ax.plot(portfolio_std_devs, portfolio_returns, 'o-', label='Efficient Portfolios w/o Constraint')

# Plot the efficient frontier with constraint on the same axes
ax.plot(portfolio_std_devs2, portfolio_returns2, 'o-', label='Efficient Portfolios w/ Constraint')

# Set labels and title
ax.set_xlabel('Standard Deviation')
ax.set_ylabel('Expected Return')
ax.set_title('Combined Efficient Frontier')

# Enable the grid
ax.grid(True)

# Set the x and y axis limits
ax.set_xlim(0, 0.4)  # Adjust the upper limit if necessary
ax.set_ylim(0,1.1)  # Adjust the upper limit if necessary

# Plot additional marker
ax.plot(mve_std, mve_return, 'bo', markersize=10, label='Mve w/o Constraint')
ax.plot(mve_std2, mve_return2, 'ro', markersize=10, label='Mve w/ Constraint')
ax.plot(tangency_volatility, tangency_return, 'y*', markersize=15, label='Tangency Portfolio w/o Constraint')
ax.plot(tangency_volatility2, tangency_return2, 'g*', markersize=15, label='Tangency Portfolio w Constraint')

# Display the legend to identify the lines
ax.legend()

# Show the combined plot
plt.show()
```