ROBERT H. SMITH
SCHOOL OF BUSINESS

Assignment 1
Portfolio Construction and Optimization

BUFN620: Capital Markets

Project Group Members:

Chi-Jui Wang
Tianyi Hou
Hongquan Lin
Weiyi Li

2024/09/08

1. Identify the largest 10 components of the S&P500 Index (based on weight in index). (Hint: Don't use Alphabet twice)

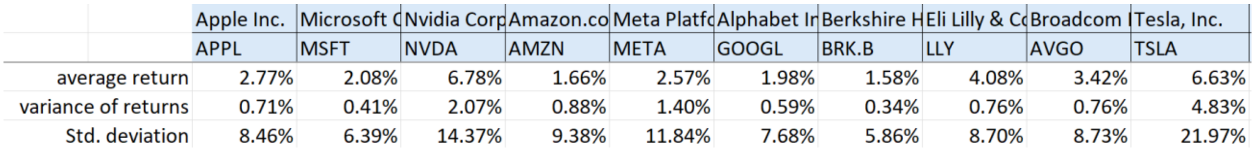
According to the information from https://www.slickcharts.com/sp500 as of 2024/9/6, we identified the largest 10 components of the S&P 500 as [Apple Inc., Microsoft Corp, Nvidia Corp, Amazon.com Inc, Meta Platforms, Inc. Class A, Alphabet Inc. Class A, Berkshire Hathaway Class B, Eli Lilly & Co., Broadcom Inc., Tesla, Inc.]. Therefore, this report will focus on analyzing these ten companies.

2. Download five years of historical monthly prices for each stock.

According to the data from https://workspace.refinitiv.com/web, we selected the stock prices of these ten companies within the range from 09/30/2019 to 08/31/2024. Additionally, we obtained the T-bill rates from https://fred.stlouisfed.org/ to use as the risk-free rate for our analysis.

3. Calculate the monthly returns, standard deviation and variance for each stock over the five years using the monthly data previously downloaded.

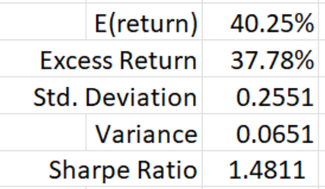
After performing calculations in Excel, we obtained the following results:



	Apple Inc.	Microsoft C	Nvidia Corp	Amazon.co	Meta Platfo	Alphabet In	Berkshire H	Eli Lilly & C	Broadcom I	Tesla, Inc.
	APPL	MSFT	NVDA	AMZN	META	GOOGL	BRK.B	LLY	AVGO	TSLA
average return	2.77%	2.08%	6.78%	1.66%	2.57%	1.98%	1.58%	4.08%	3.42%	6.63%
variance of returns	0.71%	0.41%	2.07%	0.88%	1.40%	0.59%	0.34%	0.76%	0.76%	4.83%
Std. deviation	8.46%	6.39%	14.37%	9.38%	11.84%	7.68%	5.86%	8.70%	8.73%	21.97%

4. Construct a portfolio of the ten stocks with an initial weighting of 10% each, creating an equal-weighted portfolio. What is the variance, standard deviation and the Sharpe ratio of this portfolio?

After annualizing returns and standard deviation, we obtained the following results:



E(return)	40.25%
Excess Return	37.78%
Std. Deviation	0.2551
Variance	0.0651
Sharpe Ratio	1.4811

5. Next construct the optimal combination of these stocks (i.e., maximize your Sharpe ratio) with no shorting and total weight of 100%.

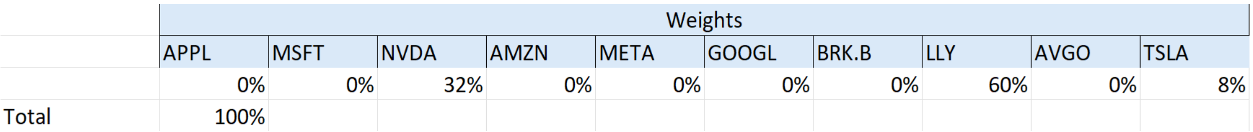
a. What are the new portfolio weights?

b. What is the new variance, standard deviation and Sharpe ratio of the portfolio?

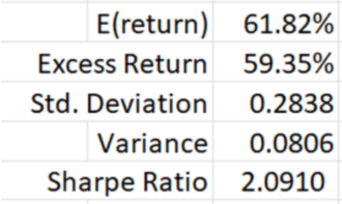
c. What stock is the most underweighted in the optimal portfolio and why?

d. Which stock is the most overweight and why?

After using the Solver function in Excel, we obtained the following results:



	Weights									
	APPL	MSFT	NVDA	AMZN	META	GOOGL	BRK.B	LLY	AVGO	TSLA
	0%	0%	32%	0%	0%	0%	0%	60%	0%	8%
Total	100%									



E(return)	61.82%
Excess Return	59.35%
Std. Deviation	0.2838
Variance	0.0806
Sharpe Ratio	2.0910

We can observe that APPL, MSFT, AMZN, META, GOOGL, BRK.B, and AVGO are underweighted in the optimal portfolio. Since these seven stocks have lower returns, reducing their weights to 0% in the optimal portfolio would result in a better Sharpe ratio. On the other hand, LLY has the second-highest return and relatively low variance. More importantly, its correlation is also quite low, indicating that it had relatively low correlation with other stocks in the portfolio, providing diversification benefits and reducing the overall risk of the portfolio. Therefore, compared to NVDA, which has the highest return, LLY carries a higher weight in the optimal portfolio.

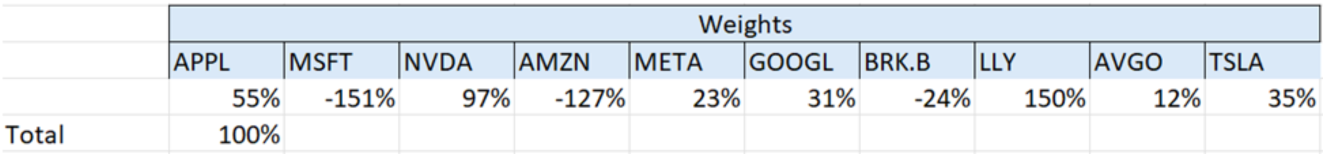
6. Optimize the portfolio one more time, but this time allows for shorting. The total weight must still add up to 100%.

a. What are the new portfolio weights?

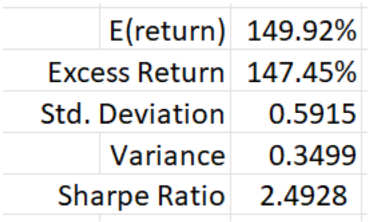
b. What is the new variance, standard deviation and Sharpe ratio?

c. What stock is the most underweighted in the optimal portfolio and why? Which stock is the most overweight and why?

After using the Solver function in Excel, we obtained the following results



	Weights									
	APPL	MSFT	NVDA	AMZN	META	GOOGL	BRK.B	LLY	AVGO	TSLA
	55%	-151%	97%	-127%	23%	31%	-24%	150%	12%	35%
Total	100%									



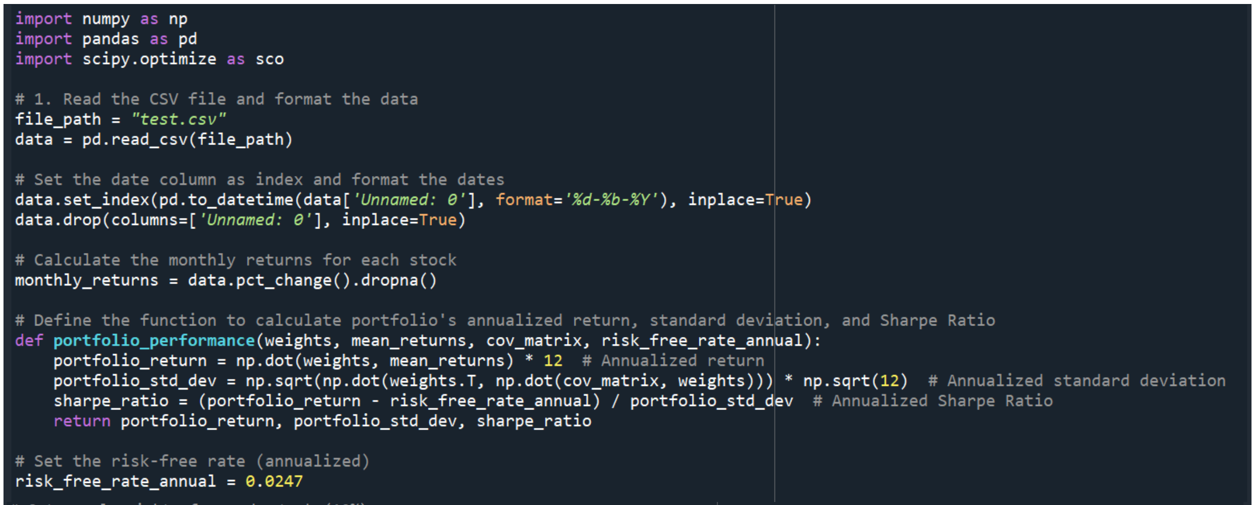
E(return)	149.92%
Excess Return	147.45%
Std. Deviation	0.5915
Variance	0.3499
Sharpe Ratio	2.4928

With shorting allowed, LLY's weight remains the highest, for the same reasons mentioned in question five. It has the second-highest return and relatively low variance. Its low correlation provide diversification benefits and reducing the overall risk of the portfolio. By the same principle, on the other hand, due to their low returns, high variance and high correlation with other stocks, MSFT is the most underweighted stock in the optimal portfolio.

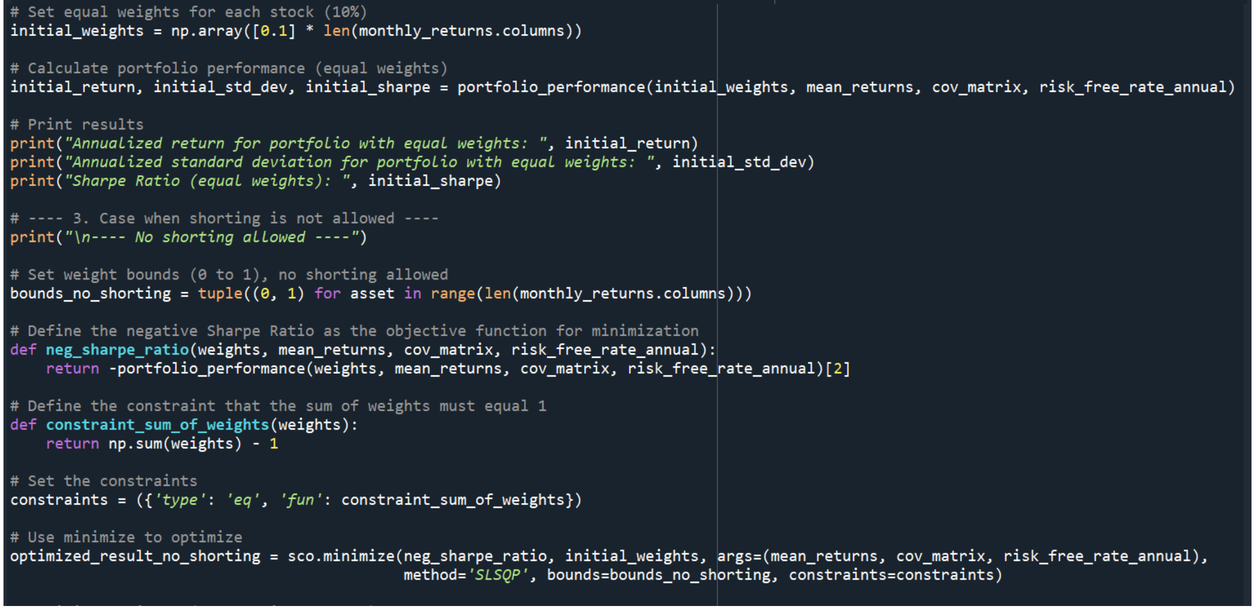
Appendix:

In addition to using Excel Solver for calculations, we also attempted to analyze using Python and obtained results similar to those from Excel. The slight discrepancies may arise from the variance calculations. In Excel, we used the population variance (VAR.P), whereas in Python, we used the sample variance. This difference in degrees of freedom could have caused the slight variance in results.

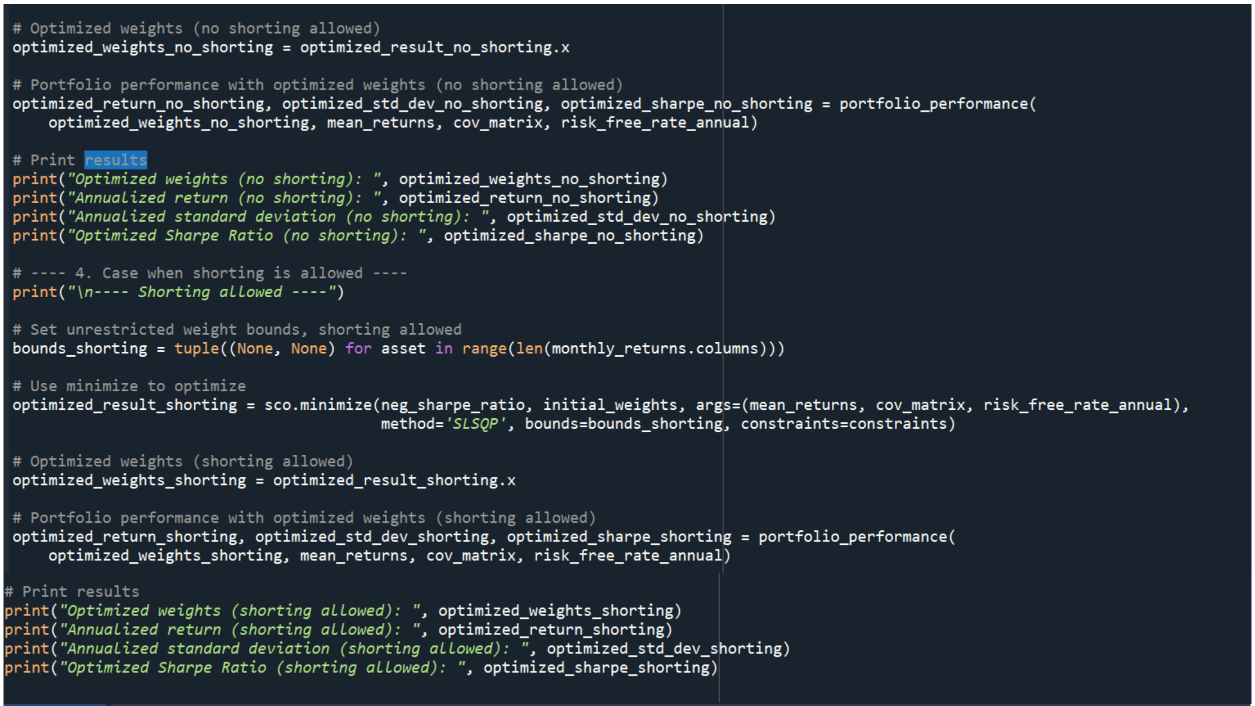
Here is the code along with the key results, and the code file will also be submitted along with the assignment:



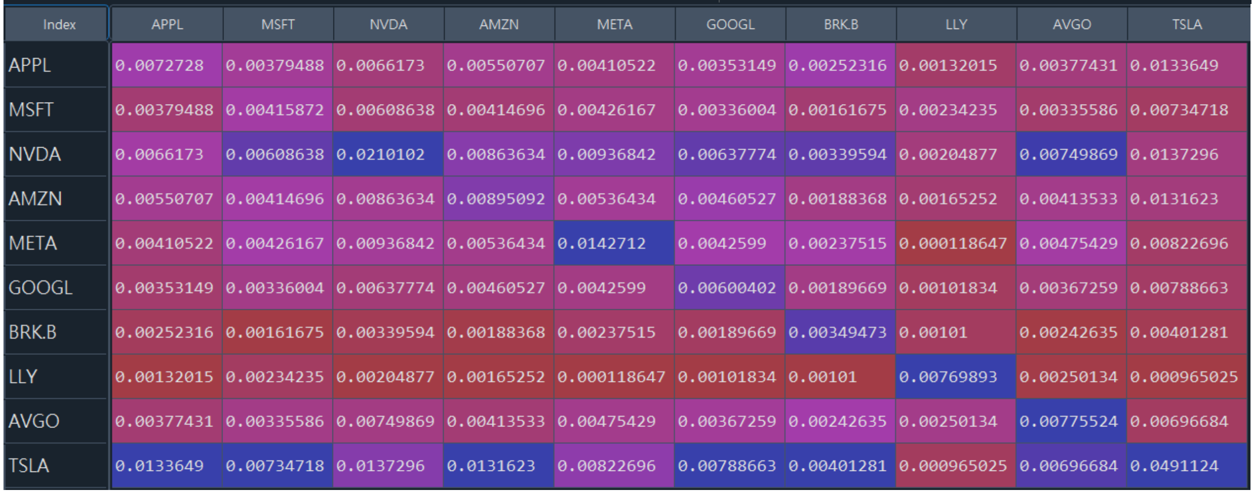
```
import numpy as np
import pandas as pd
import scipy.optimize as sco

# 1. Read the CSV file and format the data
file_path = "test.csv"
data = pd.read_csv(file_path)

# Set the date column as index and format the dates
data.set_index(pd.to_datetime(data['Unnamed: 0'], format='%d-%b-%Y'), inplace=True)
data.drop(columns=['Unnamed: 0'], inplace=True)

# Calculate the monthly returns for each stock
monthly_returns = data.pct_change().dropna()

# Define the function to calculate portfolio's annualized return, standard deviation, and Sharpe Ratio
def portfolio_performance(weights, mean_returns, cov_matrix, risk_free_rate_annual):
    portfolio_return = np.dot(weights, mean_returns) * 12  # Annualized return
    portfolio_std_dev = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights))) * np.sqrt(12)  # Annualized standard deviation
    sharpe_ratio = (portfolio_return - risk_free_rate_annual) / portfolio_std_dev  # Annualized Sharpe Ratio
    return portfolio_return, portfolio_std_dev, sharpe_ratio

# Set the risk-free rate (annualized)
risk_free_rate_annual = 0.0247
```



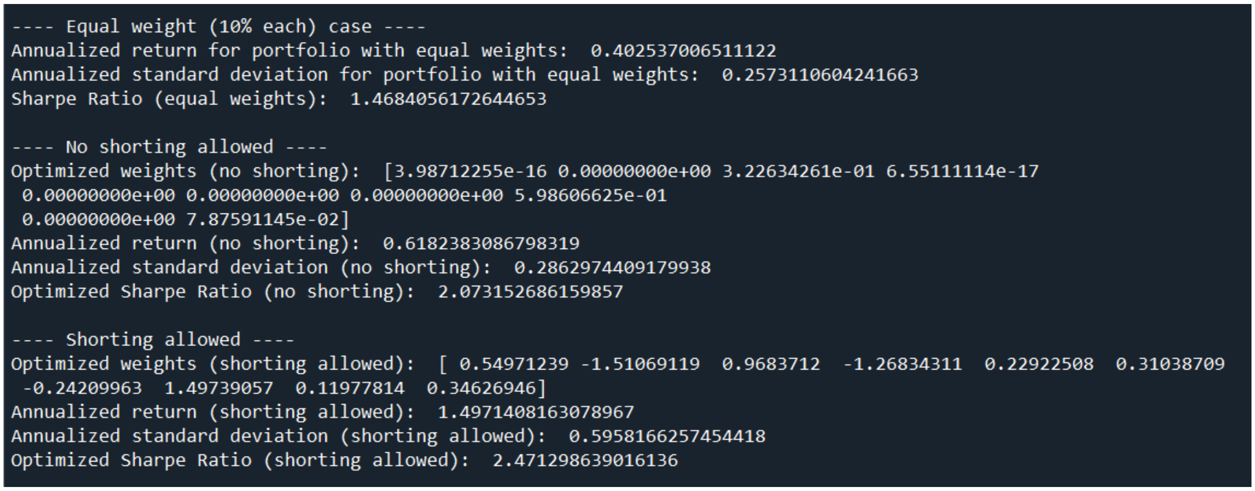
```
# Set equal weights for each stock (10%)
initial_weights = np.array([0.1] * len(monthly_returns.columns))

# Calculate portfolio performance (equal weights)
initial_return, initial_std_dev, initial_sharpe = portfolio_performance(initial_weights, mean_returns, cov_matrix, risk_free_rate_annual)

# Print results
print("Annualized return for portfolio with equal weights: ", initial_return)
print("Annualized standard deviation for portfolio with equal weights: ", initial_std_dev)
print("Sharpe Ratio (equal weights): ", initial_sharpe)

# ---- 3. Case when shorting is not allowed ----
print("\n---- No shorting allowed ----")

# Set weight bounds (0 to 1), no shorting allowed
bounds_no_shorting = tuple((0, 1) for asset in range(len(monthly_returns.columns)))

# Define the negative Sharpe Ratio as the objective function for minimization
def neg_sharpe_ratio(weights, mean_returns, cov_matrix, risk_free_rate_annual):
    return -portfolio_performance(weights, mean_returns, cov_matrix, risk_free_rate_annual)[2]

# Define the constraint that the sum of weights must equal 1
def constraint_sum_of_weights(weights):
    return np.sum(weights) - 1

# Set the constraints
constraints = ({'type': 'eq', 'fun': constraint_sum_of_weights})

# Use minimize to optimize
optimized_result_no_shorting = sco.minimize(neg_sharpe_ratio, initial_weights, args=(mean_returns, cov_matrix, risk_free_rate_annual),
                                            method='SLSQP', bounds=bounds_no_shorting, constraints=constraints)
```



```
# Optimized weights (no shorting allowed)
optimized_weights_no_shorting = optimized_result_no_shorting.x

# Portfolio performance with optimized weights (no shorting allowed)
optimized_return_no_shorting, optimized_std_dev_no_shorting, optimized_sharpe_no_shorting = portfolio_performance(
    optimized_weights_no_shorting, mean_returns, cov_matrix, risk_free_rate_annual)

# Print results
print("Optimized weights (no shorting): ", optimized_weights_no_shorting)
print("Annualized return (no shorting): ", optimized_return_no_shorting)
print("Annualized standard deviation (no shorting): ", optimized_std_dev_no_shorting)
print("Optimized Sharpe Ratio (no shorting): ", optimized_sharpe_no_shorting)

# ---- 4. Case when shorting is allowed ----
print("\n---- Shorting allowed ----")

# Set unrestricted weight bounds, shorting allowed
bounds_shorting = tuple((None, None) for asset in range(len(monthly_returns.columns)))

# Use minimize to optimize
optimized_result_shorting = sco.minimize(neg_sharpe_ratio, initial_weights, args=(mean_returns, cov_matrix, risk_free_rate_annual),
                                         method='SLSQP', bounds=bounds_shorting, constraints=constraints)

# Optimized weights (shorting allowed)
optimized_weights_shorting = optimized_result_shorting.x

# Portfolio performance with optimized weights (shorting allowed)
optimized_return_shorting, optimized_std_dev_shorting, optimized_sharpe_shorting = portfolio_performance(
    optimized_weights_shorting, mean_returns, cov_matrix, risk_free_rate_annual)

# Print results
print("Optimized weights (shorting allowed): ", optimized_weights_shorting)
print("Annualized return (shorting allowed): ", optimized_return_shorting)
print("Annualized standard deviation (shorting allowed): ", optimized_std_dev_shorting)
print("Optimized Sharpe Ratio (shorting allowed): ", optimized_sharpe_shorting)
```



Index	APPL	MSFT	NVDA	AMZN	META	GOOGL	BRK.B	LLY	AVGO	TSLA
APPL	0.0072728	0.00379488	0.0066173	0.00550707	0.00410522	0.00353149	0.00252316	0.00132015	0.00377431	0.0133649
MSFT	0.00379488	0.00415872	0.00608638	0.00414696	0.00426167	0.00336004	0.00161675	0.00234235	0.00335586	0.00734718
NVDA	0.0066173	0.00608638	0.0210102	0.00863634	0.00936842	0.00637774	0.00339594	0.00204877	0.00749869	0.0137296
AMZN	0.00550707	0.00414696	0.00863634	0.00895092	0.00536434	0.00460527	0.00188368	0.00165252	0.00413533	0.0131623
META	0.00410522	0.00426167	0.00936842	0.00536434	0.0142712	0.0042599	0.00237515	0.000118647	0.00475429	0.00822696
GOOGL	0.00353149	0.00336004	0.00637774	0.00460527	0.0042599	0.00600402	0.00189669	0.00101834	0.00367259	0.00788663
BRK.B	0.00252316	0.00161675	0.00339594	0.00188368	0.00237515	0.00189669	0.00349473	0.00101	0.00242635	0.00401281
LLY	0.00132015	0.00234235	0.00204877	0.00165252	0.000118647	0.00101834	0.00101	0.00769893	0.00250134	0.000965025
AVGO	0.00377431	0.00335586	0.00749869	0.00413533	0.00475429	0.00367259	0.00242635	0.00250134	0.00775524	0.00696684
TSLA	0.0133649	0.00734718	0.0137296	0.0131623	0.00822696	0.00788663	0.00401281	0.000965025	0.00696684	0.0491124



```
---- Equal weight (10% each) case ----
Annualized return for portfolio with equal weights:  0.402537006511122
Annualized standard deviation for portfolio with equal weights:  0.2573110604241663
Sharpe Ratio (equal weights):  1.4684056172644653

---- No shorting allowed ----
Optimized weights (no shorting):  [3.98712255e-16 0.00000000e+00 3.22634261e-01 6.55111114e-17
 0.00000000e+00 0.00000000e+00 0.00000000e+00 5.98606625e-01
 0.00000000e+00 7.87591145e-02]
Annualized return (no shorting):  0.6182383086798319
Annualized standard deviation (no shorting):  0.2862974409179938
Optimized Sharpe Ratio (no shorting):  2.073152686159857

---- Shorting allowed ----
Optimized weights (shorting allowed):  [ 0.54971239 -1.51069119  0.9683712  -1.26834311  0.22922508  0.31038709
 -0.24209963  1.49739057  0.11977814  0.34626946]
Annualized return (shorting allowed):  1.4971408163078967
Annualized standard deviation (shorting allowed):  0.5958166257454418
Optimized Sharpe Ratio (shorting allowed):  2.471298639016136
```