Programming Assignment 2: Research Report

Eli Freedman

October 19, 2025

Problem Description

This assignment focuses on performing a Monte Carlo simulation to simulate the mean and standard deviation of portfolio returns using either only long positions or a combination of long and short positions for two sets of assets:

Asset Set 1 (Dummy Variables):

1. Asset A
2. Asset B
3. Asset C
4. Asset D

Asset Set 2:

1. Procter & Gamble ($PG)
2. Coca-Cola ($KO)
3. PepsiCo ($PEP)
4. Costco ($COST)

Data Preparation and Pipeline

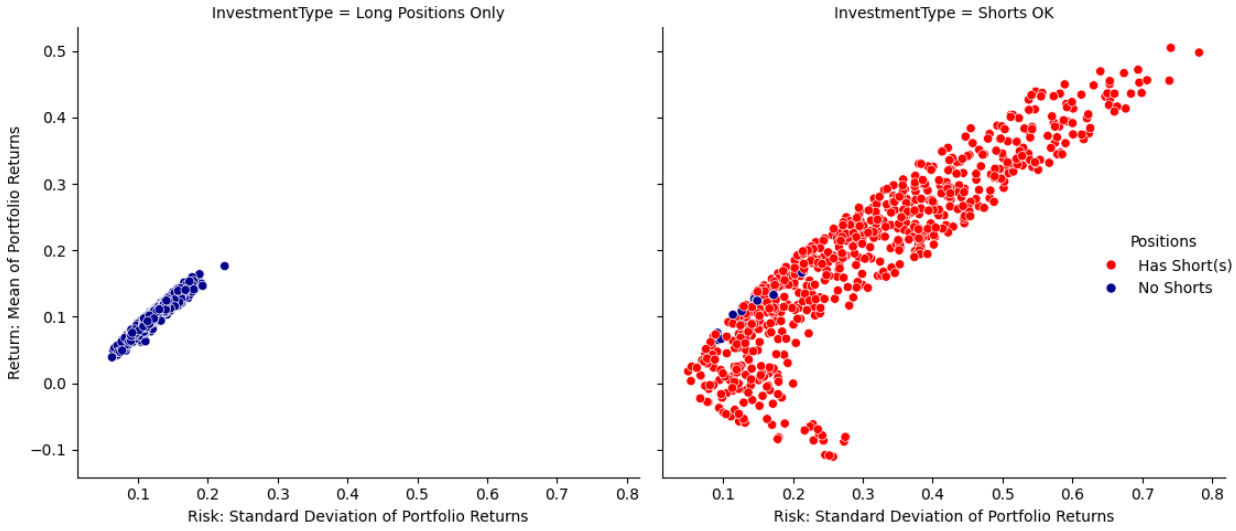
To perform a Monte Carlo simulation of returns on a portfolio of assets, we need to determine the mean annual returns, the volatility of returns, and calculate the correlation between assets within the portfolio and their covariance.

To retrieve the mean annual returns for the assets in my portfolio, I utilized FinanceCharts [1] as they provide return rates over several time ranges including 3Y, 5Y, and 20Y. Unfortunately, I was not as lucky with finding 20-year averages for volatility in returns. For this measurement, I used the yfinance [2] Python module to download 20 years of pricing data (adjusted for stock-splits and dividends), computed the daily standard deviation of log returns, and annualized it. Then, to calculate correlation and covariance, I used the cov() and corr() NumPy [3] functions. With these four variables set, I am able to perform a simulation of returns.

Research Design

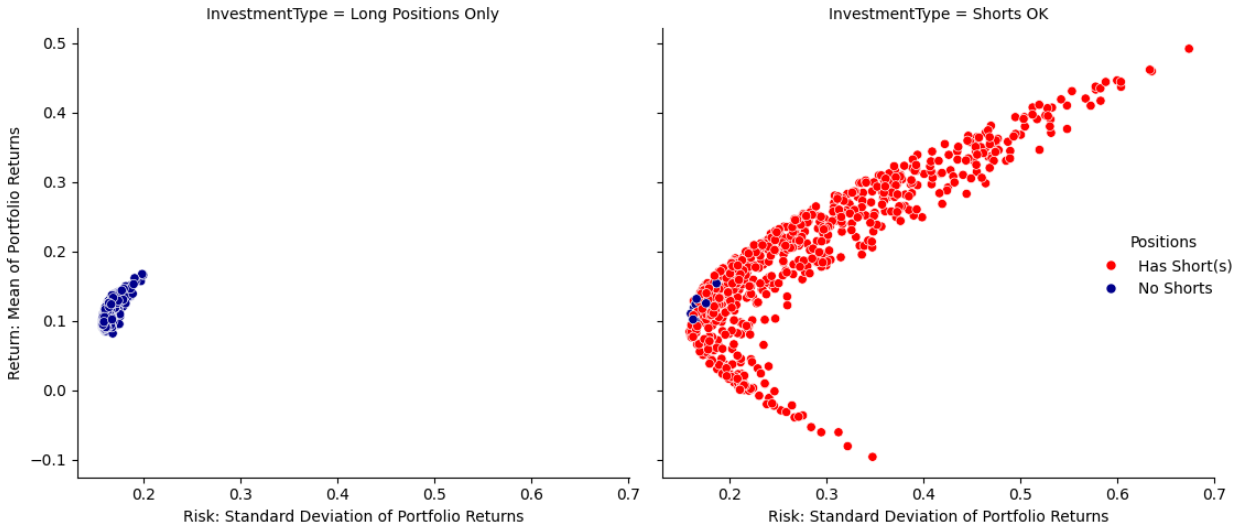
I then took the mean returns from each set of assets, the covariance from each set, set the sample size to 700, and passed this through a multivariate, normally distributed random sample to generate returns based on the previously calculated statistics.

These are the returns for asset set 1:



We can see that for these dummy assets, it is much safer and more reliable to take only long positions in this portfolio. In 700 randomly generated samples using the average returns and volatility, the mean return falls between around 3% and 18%. In the samples that also took short positions, the mean return is drastically wider, falling between around -10% and 50%. Additionally, the standard deviation for the simulation that took only long positions is between 10% and 20%, while the standard deviation for the simulation that took both long and short positions is between 5% and 80%

These are the returns for asset set 2:



We can see a similar trend here for the real equities in the second portfolio. Taking only long positions is, again, much safer, and returns around 9% to 17%. On the other hand, including short positions widens this range from around -10% to 50%. Following a similar trend, the standard deviation gap for the sim that took only long positions has a range of around 10%, whereas the range for the sim that also took short positions has a range of over 50%.

Programming

I utilized an assortment of Python packages for this assignment:

NumPy: handles math and arrays [3]
Pandas: data frames [4]
Matplotlib: basic charts and plots [5]
Seaborn: nicer plots [6]
yfinance: downloads stock market data from Yahoo! [2]

Exposition

In this assignment, I built a Monte Carlo simulation to simulate the mean and standard deviation of portfolio returns on a portfolio that either took only long positions or took a combination of long and short positions. The results showed that for steadier, more consistent returns, taking only long positions is the best option. If the owner of the portfolio is not risk-averse, however, incorporating short positions can significantly increase the rate of return, while also opening the possibility of lower returns – higher risk, higher reward.

References

[1] Financecharts.com - powerful tools for serious investors. Accessed October 17, 2025. https://www.financecharts.com.

[2] “Yahoo Finance - Stock Market Live, Quotes, Business & Finance News.” Yahoo! Finance. https://finance.yahoo.com/.

[3] NumPy. https://numpy.org/.

[4] “Pandas.” pandas. Accessed October 17, 2025. https://pandas.pydata.org/.

[5] “Visualization with Python.” Matplotlib. https://matplotlib.org/.

[6] “Statistical Data Visualization#.” seaborn. https://seaborn.pydata.org/.