Portfolio Trading Analysis Report
Name: Rajat Dua

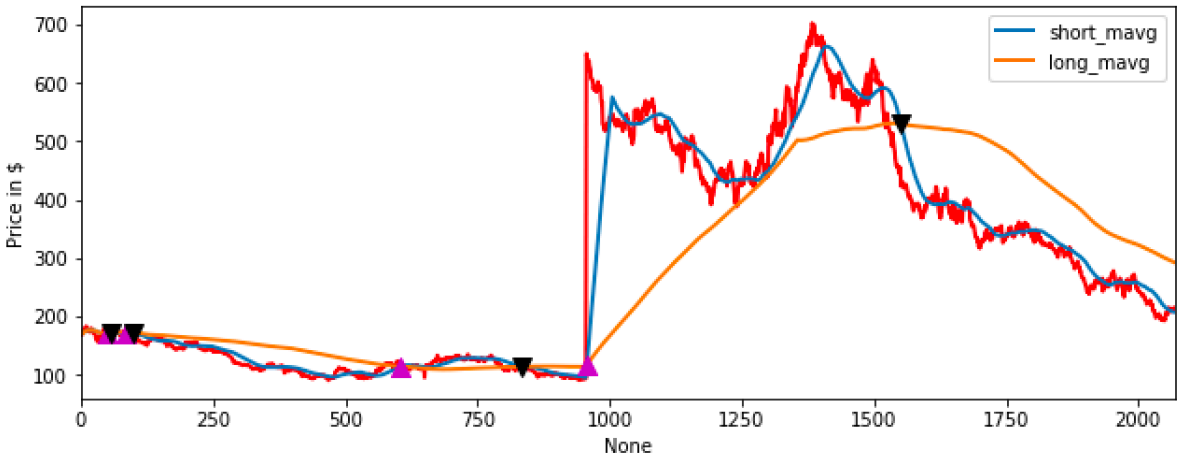
In this folder, I have incorporated some of my projects related to equity trading and portfolio simulation. These projects were highly intuitive and gave me practical experience of developing robust trading algorithms. The projects which I have included in this folder are as follows:

1. MACBackTest.py: In this problem, I have defined a class named Strategy which extracts a list of time series signals and implements the signals in generate signals function. Next, I have defined a class named Portfolio which represents a portfolio of positions (including both cash and instruments) determined using signals from Strategy class. It defines 2 functions, that is, generate_postions to allocate portfolio positions based on forecasting signals and available cash; and backtest_portfolio which generates trading orders as a sum of holdings and cash.

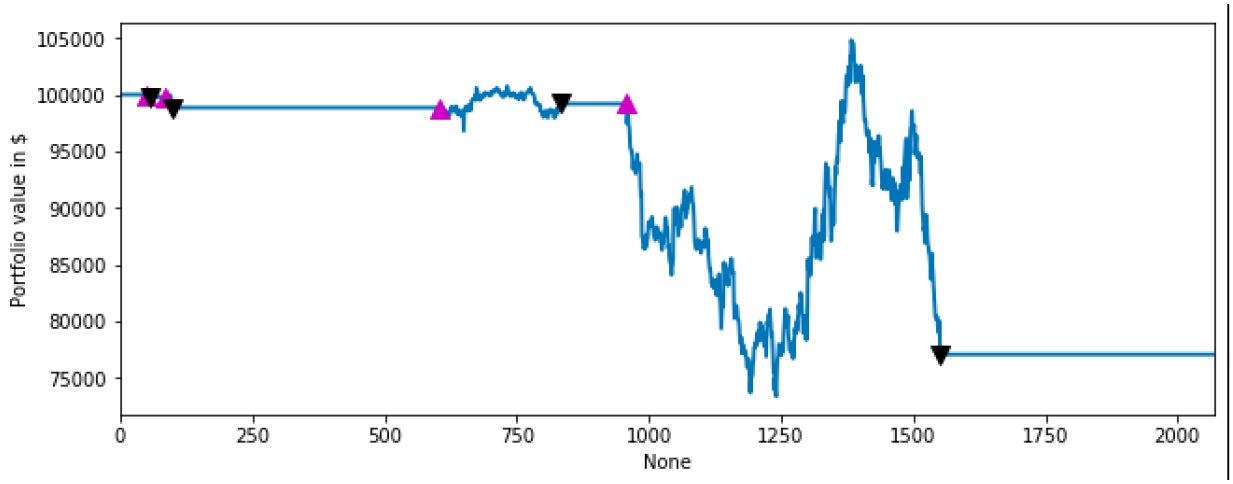
Next, I have defined a class MovingAverageCrossStrategy which creates filters based on moving averages by varying the lookback period of a particular time series. Then, I created a class called MarketOnClosePortfolio which encapsulates portfolio positions based on signals provided by Strategy class. Finally, I have implemented the functions and classes on a portfolio of stocks which include symbols/tickers such as: {'AAPL', 'GOOG', 'MSFT', 'ZNGA', 'TWTR'}. The logic behind this moving average crossover strategy is that when the shorter moving average curve exceeds the longer moving average curve, then the signal should indicate 'Buy' or purchase the asset. On the other side, when the longer moving average curve exceeds the shorter moving average curve, then the signal should indicate 'Sell' order for the asset.

The output of the code is the maximum annualized Sharpe ratio for each asset, by varying the short and long lookback period (window sizes). The initial capital amount is $100,000 and purchase sizes are fixed to 100. Also, I have incorporated a logic which prevents us from buying assets when the cash falls below $25,000. Following are the graphs generated by my code for each of the asset.

1. For Apple ('AAPL'):



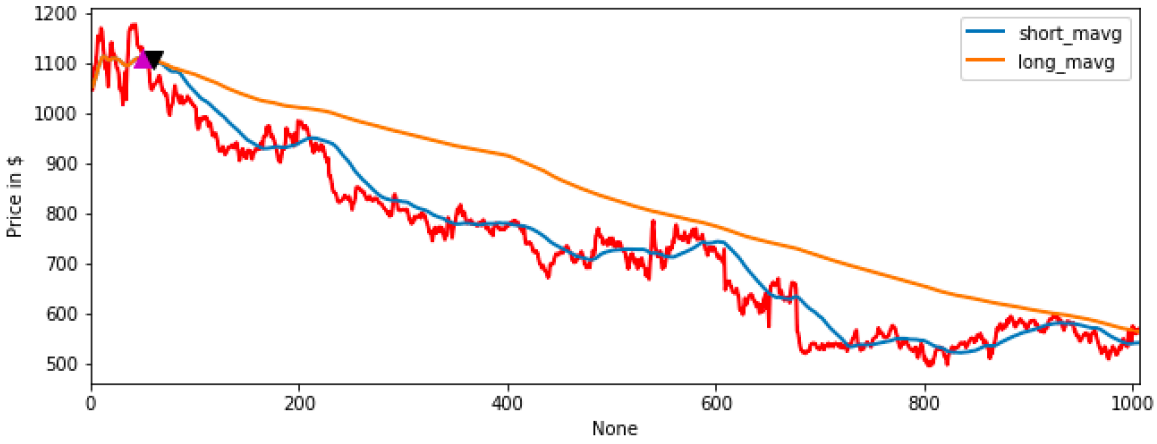
This chart plots the historical stock price movement of Apple, Short Moving Average and Long Moving Average. It generates signals to buy, hold or sell based on long and short moving averages curve.



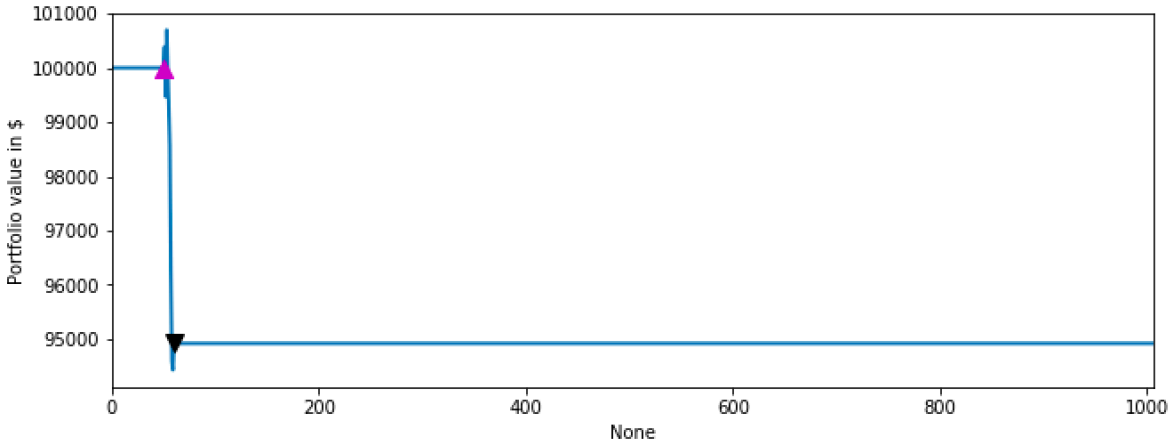
This curve represents the total value of Apple stock portfolio based on the cash and portfolio stock holdings. It is the sum of the capital left in the portfolio and stock holdings.

Maximum Annualized Portfolio Sharpe Ratio = 0.02599978279004889

2. <u>For Google ('GOOG'):</u>



This chart plots the historical stock price movement of Google, Short Moving Average and Long Moving Average. It generates signals to buy, hold or sell based on long and short moving averages curve.

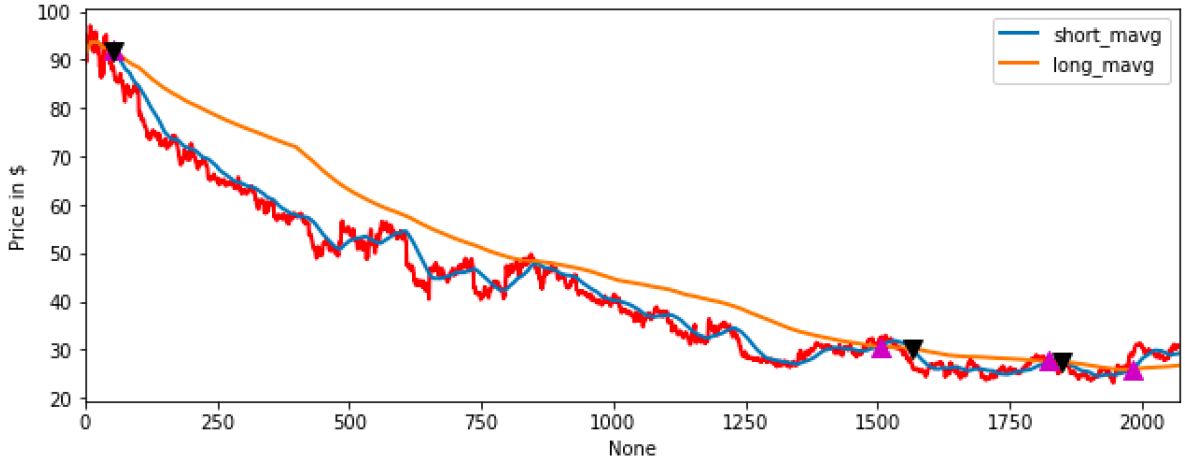


This curve represents the total value of Google stock portfolio based on the cash and portfolio stock holdings. It is the sum of the capital left in the portfolio and stock holdings.

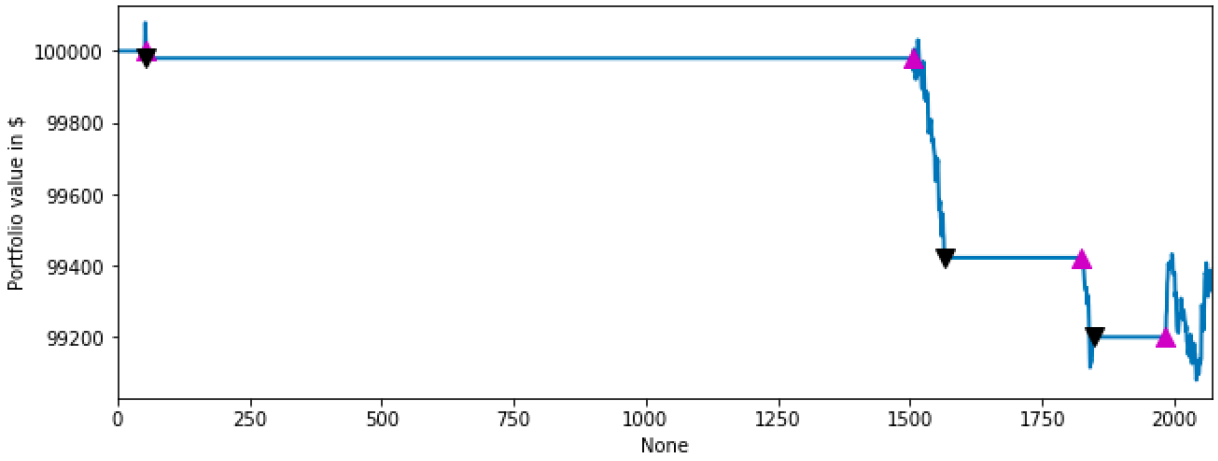
Maximum Annualized Portfolio Sharpe Ratio = 0.1981216875174513

---

3. For Microsoft ('MSFT'):



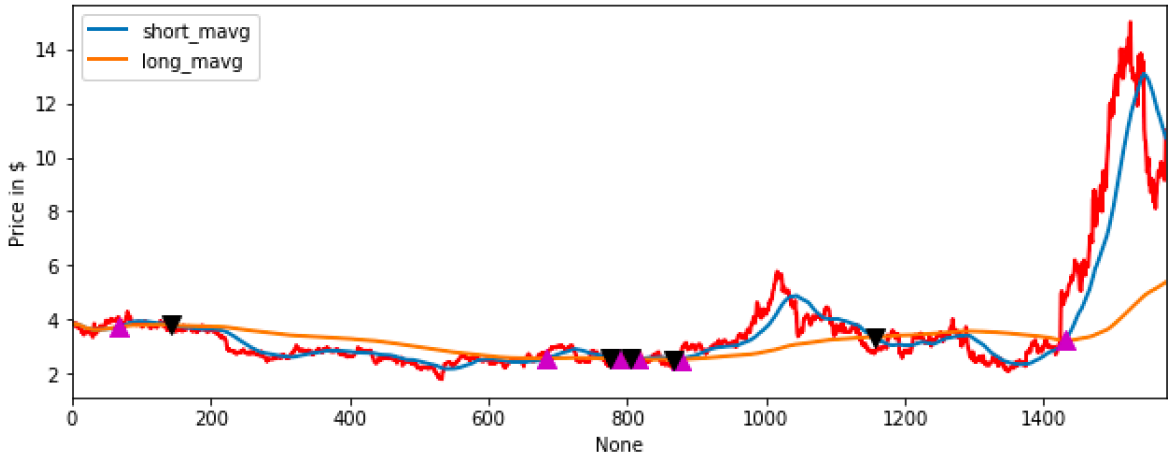
This chart plots the historical stock price movement of Microsoft, Short Moving Average and Long Moving Average. It generates signals to buy, hold or sell based on long and short moving averages curve.



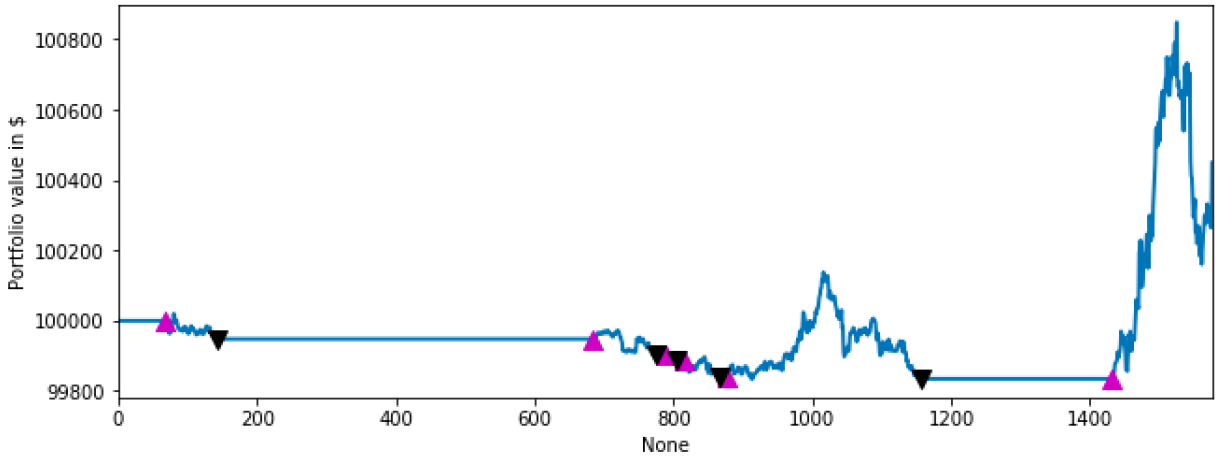
This curve represents the total value of Microsoft stock portfolio based on the cash and portfolio stock holdings. It is the sum of the capital left in the portfolio and stock holdings.

Maximum Annualized Portfolio Sharpe Ratio = 0.039172140973033474

4. For Zynga ('ZNGA'):



This chart plots the historical stock price movement of Zynga, Short Moving Average and Long Moving Average. It generates signals to buy, hold or sell based on long and short moving averages curve.

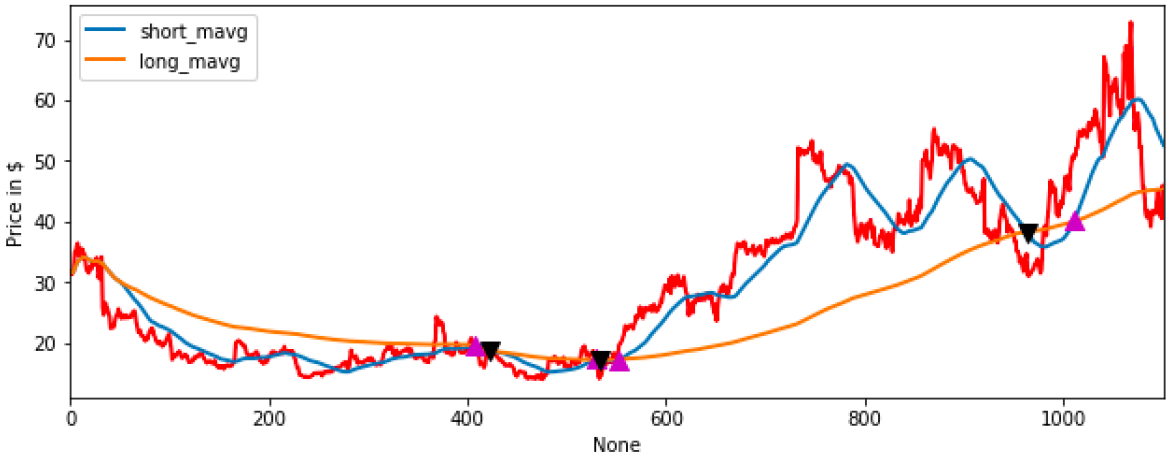


This curve represents the total value of Zynga stock portfolio based on the cash and portfolio stock holdings. It is the sum of the capital left in the portfolio and stock holdings.

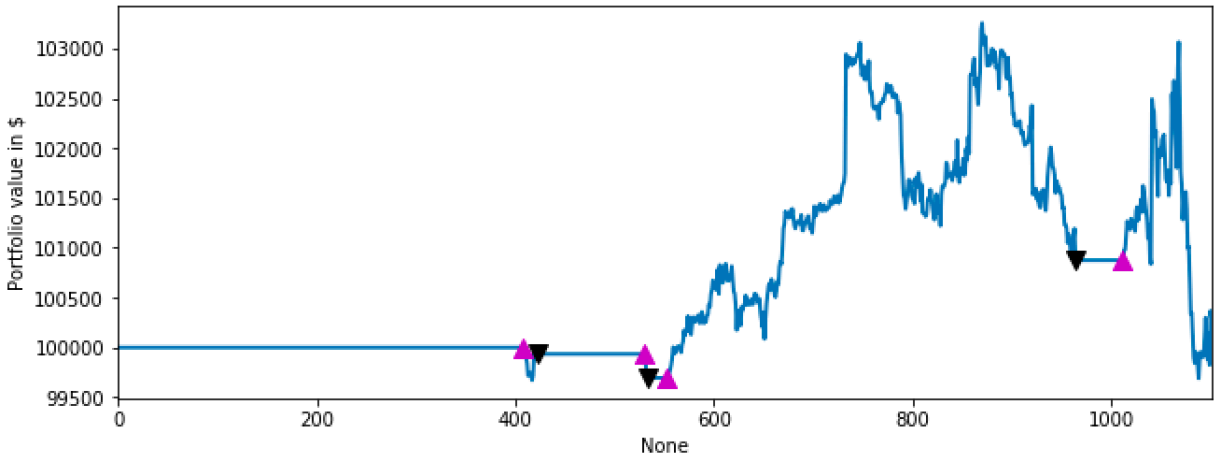
Maximum Annualized Portfolio Sharpe Ratio = 0.47499407301258934

---

5. For Twitter ('TWTR'):



This chart plots the historical stock price movement of Twitter, Short Moving Average and Long Moving Average. It generates signals to buy, hold or sell based on long and short moving averages curve.



This curve represents the total value of Twitter stock portfolio based on the cash and portfolio stock holdings. It is the sum of the capital left in the portfolio and stock holdings.

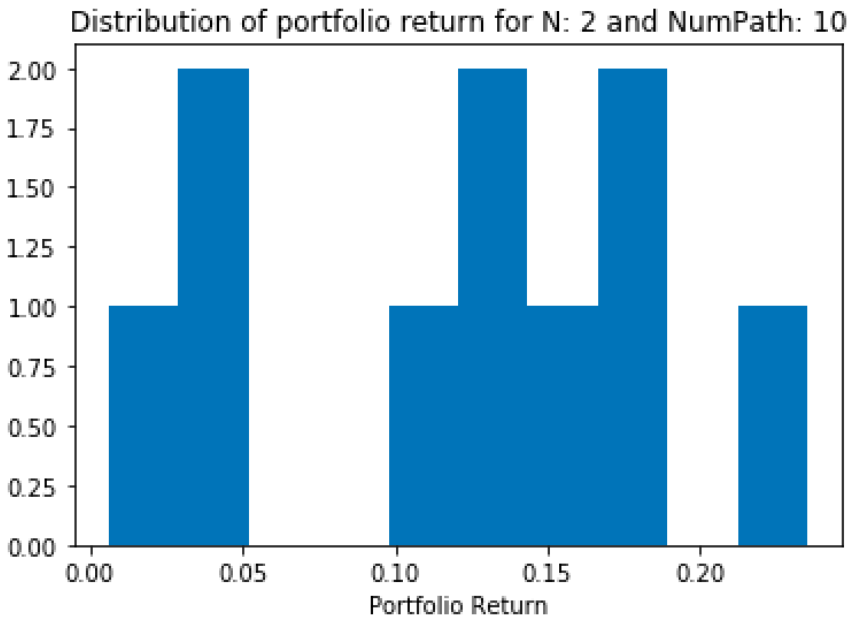
Maximum Annualized Portfolio Sharpe Ratio = 0.20636263101061997

---

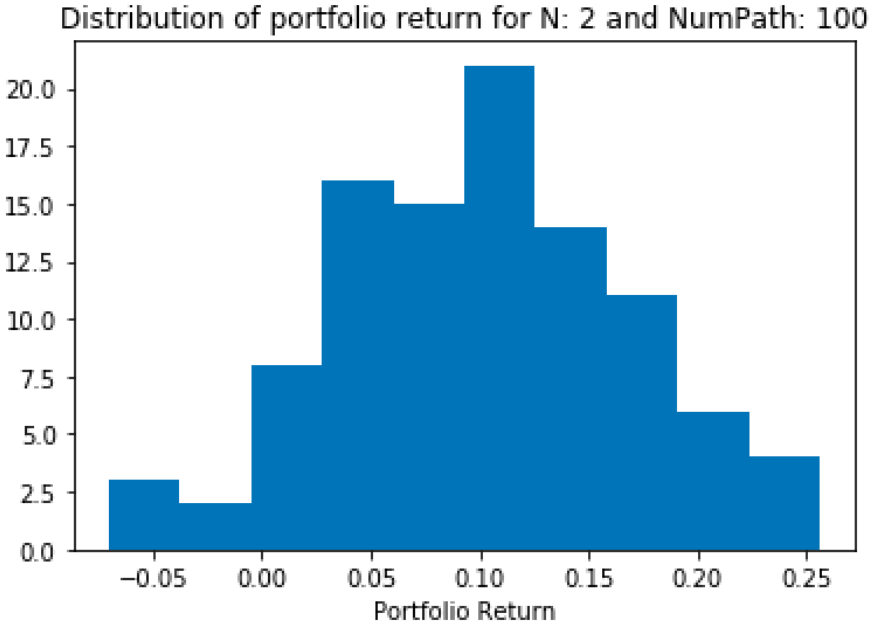
2. PortfolioSimulation.py: In this problem, I have estimated 99th percentile of portfolio loss distribution using Monte-Carlo simulation and the Euler scheme under the assumption of correlated driving Geometric Brownian Motion. I have assumed parameters as given in the question. There are 2 equally weighted stocks in the portfolio. After computationally running the algorithm for 252-time steps and 10,000 paths, I have found out that the actual 99th percentile is 0.01862 using the empirical mean-variance formula [2.326 standard deviations from the mean for 99th percentile]. Using Monte-Carlo simulation, the 99th percentile estimate came out to be 0.01849.

I have further evaluated the convergence properties of Monte-Carlo simulation technique by plotting multiple histograms for number of paths corresponding to 10, 100, 1000, 10000, and 100000. I have found that as we increase the number of paths the difference between the actual 99th percentile and estimated 99th percentile using Monte-Carlo simulation decreases. This result can be shown using the convergence of histogram curves towards the Actual 99th percentile of portfolio-loss distribution, as we increase the number of paths from 10 to 100000. The histograms are shown below:

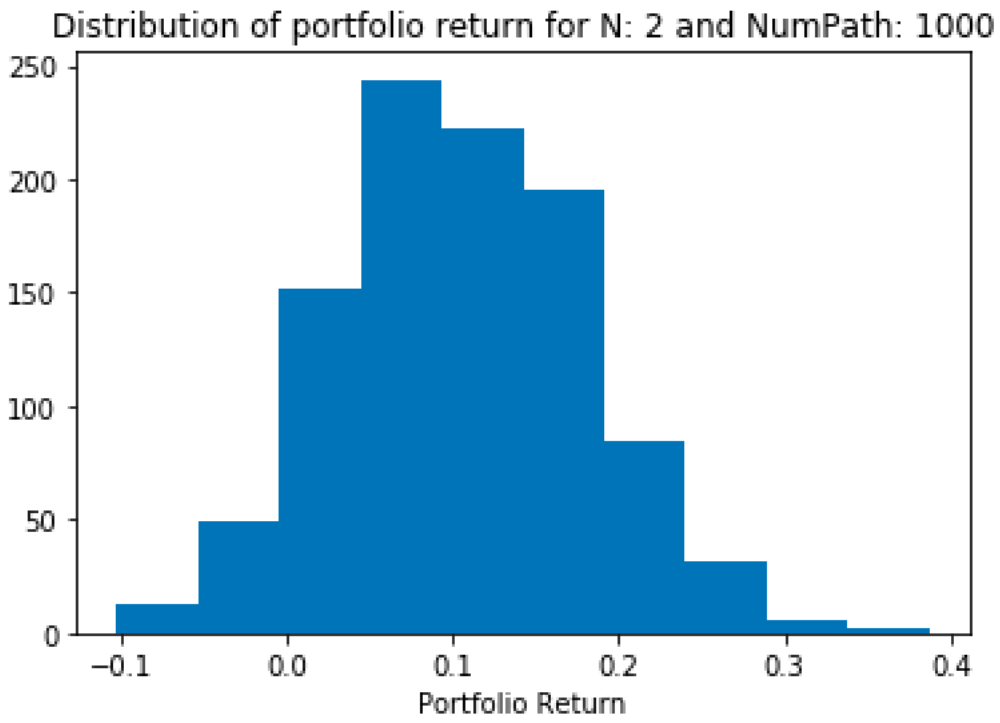
For 2 Stocks and 10 paths:



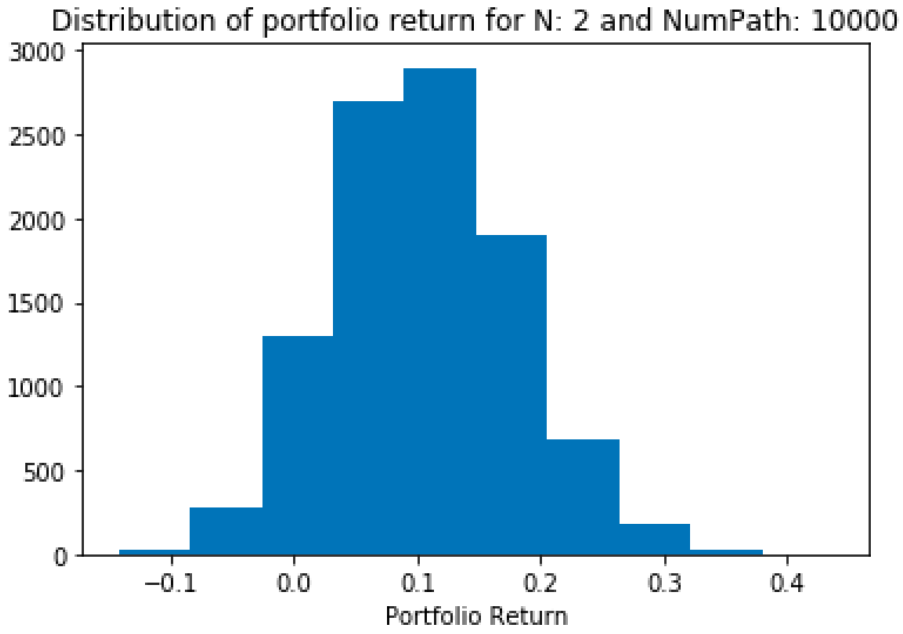
For 2 Stocks and 100 paths:



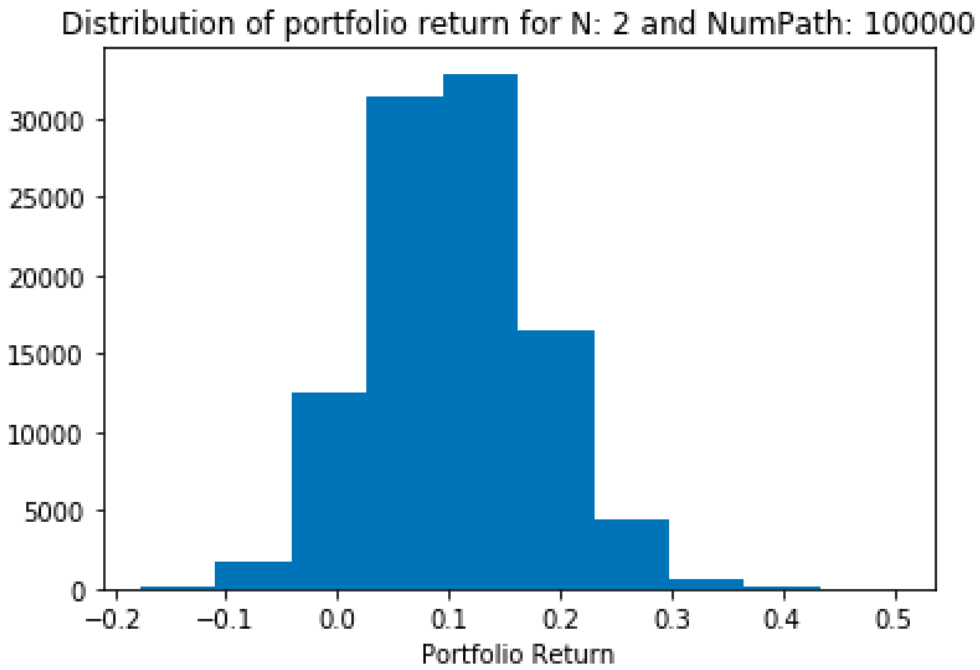
For 2 Stocks and 1000 paths:



For 2 Stocks and 10000 paths:



For 2 Stocks and 100000 paths:


Distribution of portfolio return for N: 2 and NumPath: 100000
30000
25000
20000
15000
10000
5000
0
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
Portfolio Return