# Asset allocation using Harry Markowitz's Portfolio Theory. A Python programming approach.

Omar Andujar

December 11, 2022

**Abstract**

In the mid-1950s, a revolutionary financial concept was introduced, that is *Modern Portfolio Theory*, which focused on diversifying a portfolio to maximize expected return while minimizing risk (volatility). Nowadays, with computers better than ever, I recreate the method using Python programming and analyze thousands of closing price registers for six arbitrary companies based on Markowitz's principles to deliver an optimized portfolio that could be profitable in 2022.

# Contents

# 1 Research Question

This project was motivated by the initial question, How to approach investment mathematically? One way to do it is by applying Dr. Markowitz's Modern Portfolio Theory. So the ultimate objective of this exercise is to answer the question: *Based on the data derived from a particular set of securities throughout the years, what percentage of my capital should I invest in each one, first to reduce the risk of having losses to a minimum, and second maximize my profits?.*

Answering that question raises more challenges, some of them illustrated in questions like: How can one create an investment portfolio where the resources are distributed with minimal risk exposure, but profits can still be earned? How can I develop an algorithm that does this optimization process for me? How can I make this algorithm work for all sets of securities given? How can I adapt Markowitz's theory to a program in python?

# 2 Background

Harry Markowitz won the Nobel Prize in economics in 1990 for developing Modern Portfolio Theory (MPT), a groundbreaking investment strategy based on his realization that the performance of an individual stock is not as important as the performance and composition of an investor's entire portfolio [1]. That said, the objective of this project is to recreate and automatize his theory of the allocation of financial assets under uncertainty using python programming.

Investing is often an intelligent way to counter the negative effects of inflation. With this exercise, what to invest in and where? gets answered using mathematics. Therefore, the following information may be useful to anybody that is looking to put some capital into the stock market but does not know how.

The main orientation point for this project is Markowitz's "Portfolio Selection" article, followed by numerous recreations, the most modern ones using

[1] (Behan, 2022)

programming languages, such as Python. Some examples are the works by Ahmad Bazzi [2] or Ivan Lysenko [3].

## 3 Data source(s)

The data used in this project comes from the *Yahoo! Finance* application programming interface (API) and it was imported using the python library yahoo finance, as:

```
import yfinance as yf
```

This library lets you import a data frame with financial data for a set of tickers. In this case, it was programmed to import the daily closing price of the following securities in the last 72 months (until November 2022):

- Tesla, Inc. (TSLA), Apple, Inc. (AAPL), The Coca-Cola Company (KO), The Walt Disney Company (DIS), Bank of America Corp (BAC), and Microsoft Corporation (MSFT) [4].

## 4 Treatment

First, a list of securities has to be chosen. In this case, we will base the decision on their popularity[5]. The second part is extracting the closing price information for that set of securities from the Yahoo! Finance API. Third, check the table of data to confirm that it only has the closing prices and dates of the securities picked. Then, calculate the necessary indicators for the algorithm (Expected Return, Volatility, etc.). For this, descriptive statistics (mean, standard deviation, median, etc.) and calculus will be necessary (optimization). After that, examine the relationship between stocks (using covariance and correlation). All of this is to be able to use the Markowitz algorithm that gets the coefficients needed to decide how much money to put into each security. Finally, the results will be plotted, using scatter plots and line plots. For this last part, we will use the Python Plotly library.

[2] (Bazzi, 2021)
[3] (Lysenko, 2021)
[4] Securities selected based on their high trading volume
[5] (MarketBeat, 2022)

# 5 Exploratory Data Analysis

To start, we need to get the data that we are going to work with. For this, we use a function that connects to the Yahoo! Finance API:

```
def get_stocks(stock_list, months):
    # returns a named tuple of Ticker objects
    tickers = yf.Tickers(stock_list)
    stock_df = tickers.history(period=months)['Close']
    return stock_df
```

Giving as a result the data in 1.

| Date | AAPL | BAC | DIS | KO | MSFT | TSLA |
|---|---|---|---|---|---|---|
| **2016-12-06 00:00:00-05:00** | 25.724188 | 19.557274 | 95.656853 | 33.478867 | 55.222763 | 12.390000 |
| **2022-02-03 00:00:00-05:00** | 171.902313 | 45.373516 | 140.029999 | 59.839626 | 298.453430 | 297.046661 |
| **2022-01-19 00:00:00-05:00** | 165.270813 | 45.383289 | 150.110001 | 59.247154 | 300.514130 | 331.883331 |
| **2019-09-27 00:00:00-04:00** | 53.432846 | 27.256634 | 129.186020 | 49.078590 | 133.444626 | 16.142000 |
| **2019-05-20 00:00:00-04:00** | 44.538841 | 26.060879 | 132.293213 | 43.484646 | 121.886871 | 13.690667 |
| **2022-01-10 00:00:00-05:00** | 171.196411 | 47.816631 | 156.600006 | 58.693531 | 311.352570 | 352.706665 |
| **2018-06-28 00:00:00-04:00** | 44.448505 | 25.904308 | 101.889343 | 37.738701 | 93.722115 | 23.328667 |
| **2022-04-29 00:00:00-04:00** | 156.940002 | 35.038609 | 111.629997 | 63.233784 | 275.512238 | 290.253326 |
| **2021-03-09 00:00:00-05:00** | 119.832428 | 34.996132 | 194.509995 | 47.888775 | 230.213501 | 224.526672 |
| **2016-12-15 00:00:00-05:00** | 27.097546 | 20.439821 | 99.965973 | 34.287575 | 57.645367 | 13.172000 |

Figure 1: Sample of closing prices for the securities selected

After this, the daily returns are calculated using:

- $r_t = \log(\frac{p_t}{p_{t-1}}) = \log(p_t) - \log(p_{t-1})$

With $p_t$ being the price at time $t$ and $r_t$ the return at time $t$.

| | Date | AAPL | BAC | DIS | KO | MSFT | TSLA |
|---|---|---|---|---|---|---|---|
| 0 | 2016-12-05 00:00:00-05:00 | NaN | NaN | NaN | NaN | NaN | NaN |
| 1 | 2016-12-06 00:00:00-05:00 | 0.007669 | 0.014546 | 0.006978 | -0.001231 | -0.004494 | -0.005099 |
| 2 | 2016-12-07 00:00:00-05:00 | 0.009775 | 0.018333 | 0.013126 | 0.017592 | 0.023410 | 0.038527 |
| 3 | 2016-12-08 00:00:00-05:00 | 0.009769 | 0.016697 | 0.021214 | -0.007536 | -0.005883 | -0.004462 |
| 4 | 2016-12-09 00:00:00-05:00 | 0.016190 | 0.006082 | 0.014215 | 0.024585 | 0.015613 | -0.000572 |
| 5 | 2016-12-12 00:00:00-05:00 | -0.005720 | -0.021008 | -0.007658 | -0.002383 | 0.003222 | 0.001300 |
| 6 | 2016-12-13 00:00:00-05:00 | 0.016544 | 0.000000 | -0.002020 | -0.003347 | 0.012945 | 0.029292 |
| 7 | 2016-12-14 00:00:00-05:00 | 0.000000 | 0.002650 | 0.001924 | -0.013258 | -0.004775 | 0.002722 |
| 8 | 2016-12-15 00:00:00-05:00 | 0.005454 | 0.021384 | 0.003262 | 0.008217 | -0.001597 | -0.005602 |
| 9 | 2016-12-16 00:00:00-05:00 | 0.001294 | -0.021825 | -0.004609 | 0.004562 | -0.004484 | 0.024547 |

Figure 2: First ten rows of the daily returns of closing prices for the securities selected

Then we calculated the correlation matrix, the covariance matrix, and the mean returns, using pandas[6] functions for data frame objects:

```
#Mean only for numeric columns
mean_returns = returns_df.mean(numeric_only=True)
#Annualized covariance matrix
returns_cov = returns_df.cov()*252
returns_corr = returns_df.corr()
```

[6]Python library

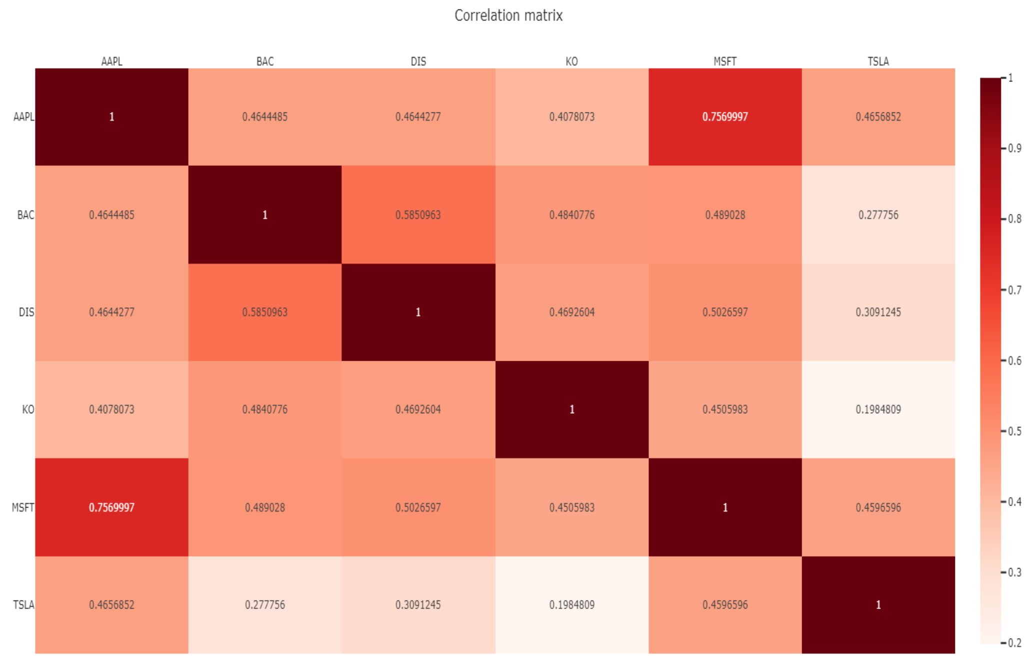


Figure 3: Correlation matrix of the securities selected

In figure 3 we can see the level of correlation between the securities. Following Markowitz's philosophy the more diversified the portfolio, the better. So, having as many uncorrelated assets as possible would be the ideal scenario. In this case, we see that there is a strong positive correlation of 0.75 between Apple and Microsoft stock prices and another high correlation of 0.59 between Bank of America and Disney stock prices. This means, that unfortunately, the securities selected don't constitute a well-diversified portfolio. Among other things this has to do with the industries from where the securities were selected, for example, Apple and Microsoft being similar technological companies. Said this, when we analyze the final results, we have to be conscious that our portfolio selection could be further optimized if we consider the possibility of exploring other securities in different industries. This happened because we put too much weight on popularity when selecting the securities.

After examining how the securities behave in comparison to each other, we examine their behavior over time. Here, we use two important plots.


Figure 4: Closing price over time for the securities selected

In figure 4 we can observe the closing price for the companies selected. As of November 2022, Microsoft has the most expensive stock at US$255, with Bank of America being the cheapest at US$37. Historically, Tesla is the one with the biggest peak and the most drastic changes, which can be confirmed in the next figure.

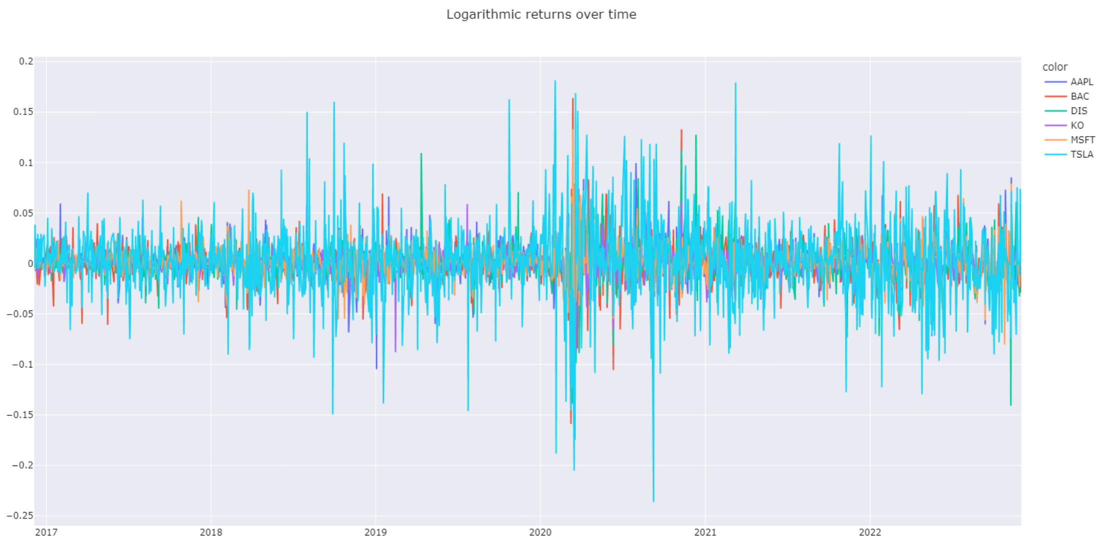


Figure 5: Daily returns variation for the securities selected

In figure 5 we can appreciate how the closing price changed over time. With this, we can see that Tesla had the most drastic changes in the closing price, for example, their almost 0.2 increase in March 2021, or their almost 0.3 decrease in September 2020. More recently in November 2022, we can observe a -0.15 decrease in the Disney stock price.

From this exploratory analysis, we can give some tentative answers to what percentage of one's capital to invest in each security. First, if we want to diversify our portfolio, and reduce risk, we should not invest the majority of our money in one industry, in this case, the technological industry, that is why it seems reasonable to have 50% of the portfolio in stocks like Tesla, Microsoft or Apple, and the other 50% in stocks like Disney, Coca-Cola or Bank of America. The other factor that we have to take into account is their return performance. From the previous figures, we see that Tesla and Disney had reasonable recent increases in their closing price, letting us believe that the algorithm will allocate a significant amount of our Portfolio to those securities. For the final answer to this question, we program and run Markowitz's model in the next section.

# 6 Testable hypothesis and model description

To face the problem of where to allocate the resources in a portfolio given the closing price historic data for a set of securities we take the approach of simulating thousands of possible combinations of allocations and testing them to see which ones are optimal based on a set of restrictions. To do this, we first have to gather some information.

We have to compute the expected return. According to James Chen: *The expected return is the profit or loss that an investor anticipates on an investment that has known historical rates of return (RoR). It is calculated by multiplying potential outcomes by the chances of them occurring and then totaling these results*[7]. The formula is the following:

$E(R) = \sum_{i=1} w_i E(R_i)$

In this case, we take the mean returns for the securities selected and we calculate the final return of that portfolio given an arbitrary weight for each security. With $w_i$ being each possible weight, and $E(R_i)$ the expected or mean return for that weight.

Then we have to take into account the risk of adopting any of the portfolio allocations generated in the simulations. For this, we define the volatility of a portfolio as:

$V = w^t \sum w$

With $w$ being the weights vector, assigning a weight to each security, $w^t$ the transpose of $w$, and $\sum$ being the covariance matrix[8].

Also, we calculate an indicator known as the Sharpe ratio. It was created by William F. Sharpe[9] and serves as a measure of the risk-adjusted return of a financial portfolio. The higher the Sharpe ratio, the more profitable the portfolio is compared to its peers. The formula is:

[7] (Chen,2021)

[8] $\sum_{ij} cov = E[(R_i - E(R_i))(R_j - E(R_j)]$ with $R_i$ being the return for the $ith$ security.

[9] Nobel Memorial Prize in Economic Sciences, 1990

$SR = \frac{R_p - R_f}{\sigma_p}$

With $R_p$ being the return of the portfolio, $R_f$ being the risk-free rate, in this case, 0.0125, and $\sigma_p$ the volatility of the portfolio.

Having the main indicators defined, we program a *for loop* that generates 20,000 random weight vectors, using:

```
weights = np.random.random(number_of_stocks)
weights = weights/np.sum(weights)    #Sum to 1
```

Computing the volatility and return for each iteration, applying the formulas defined previously. After this, we end up with a list of simulated portfolios $P$ with $(V, E(R))$ each with a specific weight distribution for the securities.

Finally, we have all the necessary data to answer the main question. Which of these portfolios we should select? To answer this we use Markowitz portfolio theory. As proposed in his article *Portfolio Selection*, first, we have to start with the assumption that the returns from securities are inter-correlated in most cases and that diversification cannot eliminate all variance. So, the portfolio with the maximum expected return is not necessarily the one with minimum variance. There is a rate at which the investor can gain expected return by taking on variance or reduce variance by giving up expected return[10]. In this case we are going to focus on two main scenarios; minimum variance and maximum return.

To derive the main result we will omit mathematical rigor since that is not the objective of this project[11]. In this case for a given portfolio of six securities we have to take into account the following conditions:

- $E(R) = \sum_{i=1}^{6} w_i * E(R_i)$
- $V = W^t * \sum * W \ (W \in R^6)$
- $\sum_{i=1}^{6} w_i = 1$
- $w_i >= 0$ for every $i$

[10](Markowitz, 1952)

[11]The reader can refer to the original article: *Portfolio Selection* (Markowitz, 1952)

In other words, the sum of the percentages invested in each security must be 100%, there should not be a negative percentage, and the volatility and expected return should be computed as defined previously.

Given these constraints, we perform convex optimization using *scipy.optimize*[12], with the objective of producing an efficient frontier on the graph composed by the 20,000 points with $(V_n, E(R_n))$. The efficient frontier can be defined as the set of investment portfolios that produce the highest expected returns given a certain volatility (or level of risk).

# 7 Results/Model Output

Finally, after configuring the model, we run the simulations and plot the results, ending with figures 6 and 7.

In figure 6 we stand out the minimum volatility and the maximum Sharpe ratio points (with yellow and red) as the portfolios with less risk and the biggest return respectively. The dark blue line surrounding the points is composed of the portfolios that have the least amount of risk given a certain return (efficient frontier). The points $(V, E(R))$ are color-coded based on their Sharpe ratio. The higher the Sharpe ratio, the bluer the color.

[12]Python Library

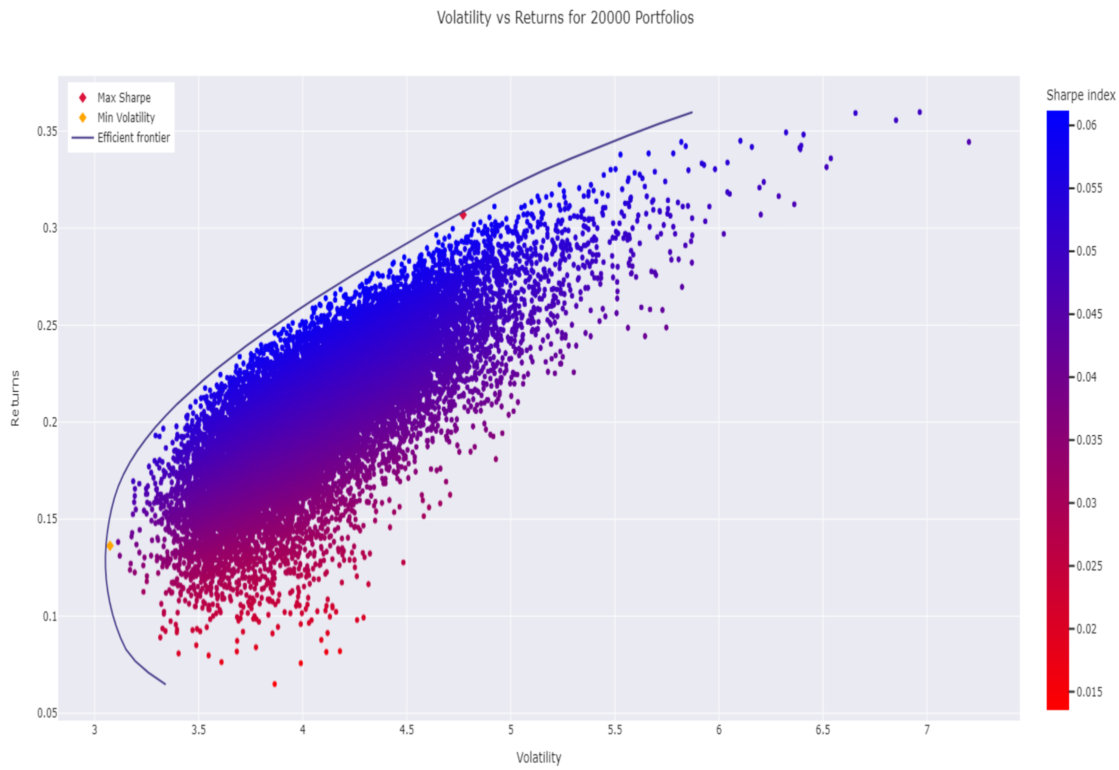


Figure 6: Markowitz efficient frontier

Each point in that scatter plot has a set of weights assigned, one for each security in the portfolio. In this case, we are only interested in the weights on the efficient frontier or at least close to it. We highlight two important cases: the Maximum Sharpe Portfolio and the Minimum Volatility Portfolio. The results of these two scenarios are presented in figure 7.

```
Maximum Sharpe portfolio with weights
  Stocks Weights
0   TSLA  46.03%
1   AAPL   0.58%
2     KO   0.58%
3    DIS   5.05%
4    BAC  26.21%
5   MSFT  21.54%

Minimum volatility portfolio with weights
  Stocks Weights
0   TSLA  10.76%
1   AAPL   1.94%
2     KO  11.54%
3    DIS  63.35%
4    BAC  11.82%
5   MSFT   0.59%
```

Figure 7: Weights for maximum Sharpe ratio and minimum volatility

According to figure 7, if a person wants to be exposed to the minimum risk possible, given the returns data, he should invest 10.76% of his money in Tesla, 1.94% in Apple, 11.54% in Coca-Cola, 63.35% in Disney, 11.82% in Bank of America, and 0.59% in Microsoft.

On the other hand, if the investor is comfortable with a higher risk and wants to pursue higher expected earnings, he should adhere to the Maximum Sharpe Portfolio, and invest 46.03% in Tesla, 0.58% in Apple, 0.58% in Coca-Cola, 5.05% in Disney, 26.21% in Bank of America, and 21.54% in Microsoft.

# 8 Interpretation of Results

The model output confirmed the initial hypothesis that diversifying the portfolio is sensible to reduce volatility. In an ideal case, if one security is extremely profitable, it will be logical to put 100% of your capital there, but since uncertainty exists it is safer to diversify. This was confirmed with the results in figure 7 for the minimum volatility portfolio. There we saw that the weights weren't distributed in just one industry and very volatile stocks such as Tesla didn't get a big percentage of the portfolio.

On the opposite side, if one wants the highest expected returns possible, it is better to go for the securities with the best performances, being less strict on diversification. This was shown by the model when it assigned a very low percentage to companies such as Coca-Cola and Apple, and a very high percentage to companies such as Microsoft and Tesla.

The model also showed that investment can be approached mathematically and that interpreting data could help make more educated decisions. The opportunity of seeing the behavior of the stocks via simulations and historic plots gives an overall vision of the companies' development in the market. Referring to optimization of portfolios in his Nobel speech Markowitz said: *Equipped with databases, computer algorithms and methods of estimation, the modern portfolio theorist is able to trace out mean-variance frontiers for large universes of securities... We seek a set of rules which investors can follow in fact - at least investors with sufficient computational resources. Thus, we prefer an approximate method which is computationally feasible to a precise one which cannot be computed.*[13]

# 9 Conclusion and Discussion

Harry Markowitz and William Sharpe are very important references in the quantitative finance world. This project helped me get acquainted with some of their work and motivated me to keep doing research in this area. They were trailblazers who innovated in an area that did not have recognition. As Markowitz shared in his Nobel speech: *...when I defended my dissertation as a student in the Economics Department of the University of Chicago, Professor Milton Friedman argued that portfolio theory was not Economics, and that they could not award me a Ph.D. degree in Economics for a dissertation which was not in Economics. I assume that he was only half serious, since they did award me the degree without long debate. As to the merits of his arguments, at this point I am quite willing to concede: at the time I defended my dissertation, portfolio theory was not part of Economics. But now it is.*

Nowadays quantitative finance is at the core of financial markets, with references like James (Jim) Simmons and Renaissance Technologies, the possibility of taming the market using mathematical models is more real than ever.

[13](Markowitz, 1990)

That possibility motivates me to keep working in the area, and it raises questions such as: Why have some hedge funds been successful using this method and some others haven't? What is more effective to analyze the market, technical analysis or fundamental analysis? Why is it so hard to have a formula that predicts market behavior? Are Harry Markowitz and William Sharpe's postulates useful in today's economy? Among other questions.

# A Source code

Github repository link:
https://github.com/OmarAndujarL/MarkowitzSimulation.git

# B References


- Markowitz, Harry (1952). Portfolio Selection. The Journal of Finance, 7(1), 77–91. https://doi.org/10.2307/2975974

- Markowitz, Harry (Dec.7, 1990). Foundations of Porfolio Theory. Nobel Lecture. Baruch College, The City University of New York, New York, USA.

- Banas, Derek (2021). Modern Portfolio Theory : Python for Finance 8. Retrieved November 30, 2022, from https://www.youtube.com/watch?v=f2BCmQBCwDs&t=505s.

- Bazzi, Ahmad (Feb.1, 2021) Stock Market Analysis and Markowitz Efficient Frontier on Python - Python 11. Retrieved December 3, 2022, from https://www.youtube.com/watch?v=f2BCmQBCwDs&t=579s

- Starke, T., Edwards, D., & Wiecki , T. (n.d.). Markowitz Portfolio Optimization in Python/v3. Retrieved November 30, 2022, from https://plotly.com/python/v3/ipython-notebooks/markowitz-portfolio-optimization/.

- Lysenko, Ivan (Jan.3, 2021). Modern portfolio theory model implementation in Python. Retrieved December 3, 2022, from https://medium.com/analytics-vidhya/modern-portfolio-theory-model-implementation-in-python-e416facabf46

- Efstathopoulos, Georgios (2022). Python for Finance, Part I: Yahoo & Google Finance API, pandas, and matplotlib. Retrieved December 3, 2022, from https://www.learndatasci.com/tutorials/python-finance-part-yahoo-finance-api-pandas-matplotlib/

- Aroussi, Ran (Nov.16, 2022) yfinance 0.1.87
  https://pypi.org/project/yfinance/

- Behan, Ann (May 5, 2022) Who Is Harry Markowitz? Retrieved December 3, 2022, from
  https://www.investopedia.com/terms/h/harrymarkowitz.asp

- Chen, James (Aug. 29, 2021) Expected Return: Formula, How It Works, Limitations, Example. Retrieved December 6, 2022, from
  https://www.investopedia.com/terms/e/expectedreturn.asp

- The Economic Times (n.d.) What is 'Sharpe Ratio'. Retrieved December 6, 2022, from
  https://economictimes.indiatimes.com/definition/sharpe-ratio

- CFI Team (Oct. 23, 2022) Efficient Frontier. Retrieved December 6, 2022, from
  https://corporatefinanceinstitute.com/resources/wealth-management/efficient-frontier/

- MarketBeat (n.d.) Most Active Stocks by Dollar Volume. Retrieved December 6, 2022, from
  https://www.marketbeat.com/market-data/most-active-stocks/