NATIONAL INSTITUTE OF TECHNOLOGY - KARNATAKA, SURATHKAL

THE INSTITUTION OF ENGINEERS - NITK CHAPTER
Institutional Member ID: IM0001491
https://ie.nitk.ac.in

Optimum Portfolio Allocation

Team:

Zuhayr Shariff
Dhruv Jha
Hariharan
Rishik TS

Problem statement:

You might have seen people investing in multiple stock companies seeing their past trends and still worry whether they'll get the desired returns. It's disheartening to see people give so much of their time and energy to allocate their asset weights for different companies and still not get the desired results.
We want to tackle this issue using automated optimized asset weights allocation for a portfolio by maximizing expected return based on a given level of market risk in Python. Further we used Monte Carlo simulation for calculating the Value at Risk (VAR) for the optimized portfolio. Hence, users need not to worry about weights allocation & returns for their portfolio.

Strategy:

The rule of thumb when it comes to purchase stocks is simply to **buy low and sell high**.
We have decided to leverage this idea and extrapolate it to the COVID-19 pandemic.

The central dogma is to target companies in industries that do poorly during the pandemic (i.e. low price) but can bounce back in the long run (i.e., get correct valuation).

We have primarily targeted the travelling (**specifically aviation**) industry and banking industry which would hypothetically perform badly during the pandemic due to **reduced tourist activity** and **increased default rates of loans** (caused by shutting down businesses) respectively. However in the **long run** both these industries should be restored to normalcy and the portfolio can be liquidated.

Methodology:

We spent the first few months of the project familiarizing ourselves with **basic terminologies** used in the stock market like **Volatility** (basically how risky the stock is compared to market), **Alpha** (excess return on an investment), **Beta of a stock** (stock's volatility compared to the market), **Sharpe ratio** - basically risk-adjusted return measurement; Higher the Sharpe Ratio more optimal the portfolio is, generally Sharpe Ratio greater than **1.00** is considered as good, We grasp all these majorly through **Investopedia** and **YouTube**.
Later we started with the code part of it.
We loaded stock data of **10 NSE Companies** :-

1. Indigo
2. Jet Airways
3. ICICI Bank
4. SBI Bank
5. TATA Chemicals
6. TATA Communication
7. HDFC Bank
8. BAJAJ Finance
9. KOTAK Bank
10. RELIANCE Industries

And then first calculated **percentage daily return** of each company & **covariance matrix**. Covariance matrix is used here to calculate the standard deviation of companies which in turn is used to measure the risk associated with each company. Which came out to be –

	INDIGO	ICICI	Jet_Airways	SBI	TATA Chem	TATA Comm	HDFC	Bajaj Finance	KOTAK	Reliance
INDIGO	0.000617	0.000057	-0.000147	0.000048	0.000011	0.000059	0.000056	0.000089	0.000050	0.000045
ICICI	0.000057	0.000322	0.000160	0.000241	0.000102	0.000100	0.000115	0.000190	0.000111	0.000067
Jet_Airways	-0.000147	0.000160	0.006872	0.000208	0.000177	-0.000058	0.000102	0.000165	0.000005	0.000074
SBI	0.000048	0.000241	0.000208	0.000480	0.000122	0.000158	0.000152	0.000208	0.000115	0.000087
TATA Chem	0.000011	0.000102	0.000177	0.000122	0.000252	0.000097	0.000052	0.000096	0.000051	0.000015
TATA Comm	0.000059	0.000100	-0.000058	0.000158	0.000097	0.000993	0.000079	0.000102	0.000070	0.000040
HDFC	0.000056	0.000115	0.000102	0.000152	0.000052	0.000079	0.000220	0.000132	0.000097	0.000034
Bajaj Finance	0.000089	0.000190	0.000165	0.000208	0.000096	0.000102	0.000132	0.000416	0.000142	0.000071
KOTAK	0.000050	0.000111	0.000005	0.000115	0.000051	0.000070	0.000097	0.000142	0.000193	0.000053
Reliance	0.000045	0.000067	0.000074	0.000087	0.000015	0.000040	0.000034	0.000071	0.000053	0.000292

Table 1.1

Next, we assigned them **random weights** manually and then found return & volatility, just to know how an **ideal portfolio** with these companies works. It came out to be a pretty decent one with a return of **0.19240409932213892.** Hmmm not bad, but we want more returns that's why we are here for :p

We proceeded with finding **annual return & volatility** for each company, just to get an idea of how they performed annually.

	Returns	Volatility
INDIGO	0.216268	0.384953
ICICI	0.440757	0.277820
Jet_Airways	-1.585271	1.284217
SBI	0.166892	0.339468
TATA Chem	-0.028061	0.246118
TATA Comm	-0.166241	0.488268
HDFC	0.219995	0.229976
Bajaj Finance	0.517981	0.316035
KOTAK	0.324503	0.215026
Reliance	0.354910	0.264654

Table1.2

After getting a rough idea about companies' movements in the stock market. We plotted **returns vs volatility** for **1000 different portfolios** with **random weights**. Remember more the no. of simulations (e.g., 1000 in our case), more possibility of finding a better portfolio i.e., portfolio with better Sharpe ratio.

```python
for portfolio in range(num_portfolios):
    weights = np.random.random(num_assets)
    weights = weights/np.sum(weights)
    p_weights.append(weights)
    returns = np.dot(weights, ind_er) # Returns are the product of individual expected returns of asset and its weights.
    p_ret.append(returns)
    var = cov_matrix.mul(weights, axis=0).mul(weights, axis=1).sum().sum()  # Portfolio Variance
    sd = np.sqrt(var)                                                        # Daily standard deviation
    ann_sd = sd*np.sqrt(240)                                                 # Annual standard deviation = volatility
    p_vol.append(ann_sd)
    ratio = np.subtract(p_ret,rf)
    sharpe_ratio = np.divide(ratio, p_vol)
```

Fig. 1.1

Extracted a **data frame** from this which included Sharpe ratio, volatility, returns and companies' weights for all 1000 different portfolios.

	Returns	Volatility	Sharpe Ratio	INDIGO weight	ICICI weight	Jet_Airways weight	SBI weight	TATA Chem weight	TATA Comm weight	HDFC weight	Bajaj Finance weight	KOTAK weight	Reliance weight
0	-0.084708	0.248819	-0.501200	0.068220	0.145953	0.145539	0.042832	0.155335	0.129469	0.187842	0.000645	0.002151	0.122015
1	-0.109831	0.252117	-0.594291	0.097394	0.052366	0.149658	0.072227	0.218422	0.134606	0.013071	0.027937	0.025702	0.208617
2	0.239098	0.200437	0.993323	0.033148	0.088289	0.002770	0.077384	0.079760	0.170807	0.112881	0.252767	0.174276	0.007918
3	-0.036328	0.275319	-0.277237	0.194605	0.075348	0.172355	0.022890	0.028521	0.067854	0.102654	0.175656	0.017505	0.142611
4	-0.095094	0.259157	-0.521281	0.047044	0.057579	0.135872	0.144090	0.113211	0.208517	0.141935	0.085800	0.017550	0.048402
...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	-0.068510	0.246009	-0.441083	0.021210	0.074055	0.131091	0.149307	0.137676	0.159067	0.100612	0.036777	0.040269	0.149936
996	0.239825	0.172045	1.161471	0.014086	0.144580	0.005252	0.074602	0.107222	0.095276	0.155394	0.092246	0.154231	0.157111
997	0.085007	0.206253	0.218212	0.055903	0.154535	0.095154	0.055155	0.067309	0.078497	0.160507	0.079054	0.163868	0.090017
998	0.148622	0.192511	0.564241	0.170534	0.047643	0.053676	0.037200	0.070468	0.115060	0.174054	0.219311	0.030533	0.081521
999	0.225952	0.178092	1.044136	0.024894	0.121122	0.040846	0.058360	0.067748	0.024254	0.200830	0.139127	0.088484	0.234336

1000 rows × 13 columns

Table 1.3

Plotted a **Return vs Volatility Scatter Plot** for our Data Frame. As you can see the **top-most point** must be one with the best **Sharpe Ratio** and hence optimal portfolio out of all. Similarly, **left-most** point must be the least **volatile** one.

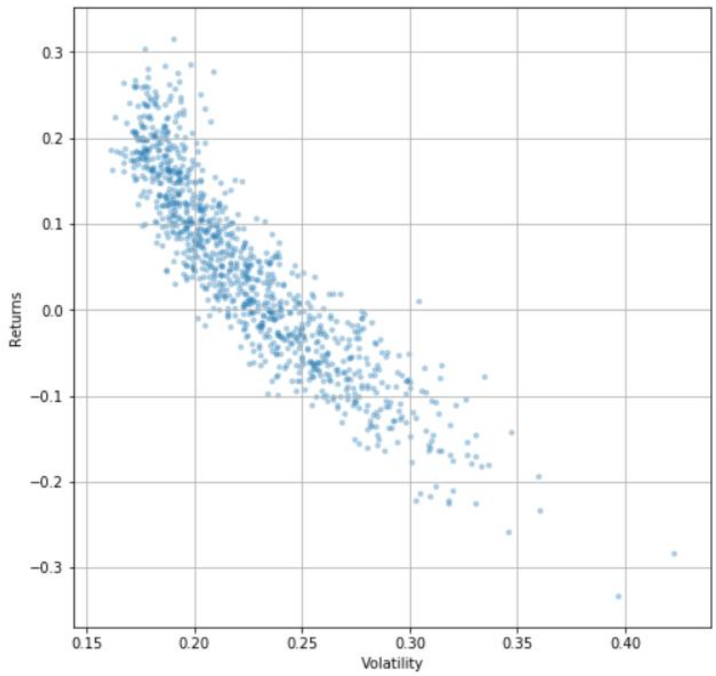

Fig. 1.2

And hence **STAR marked** the optimal portfolio (max Sharpe Ratio) with GREEN and least volatile one with RED.

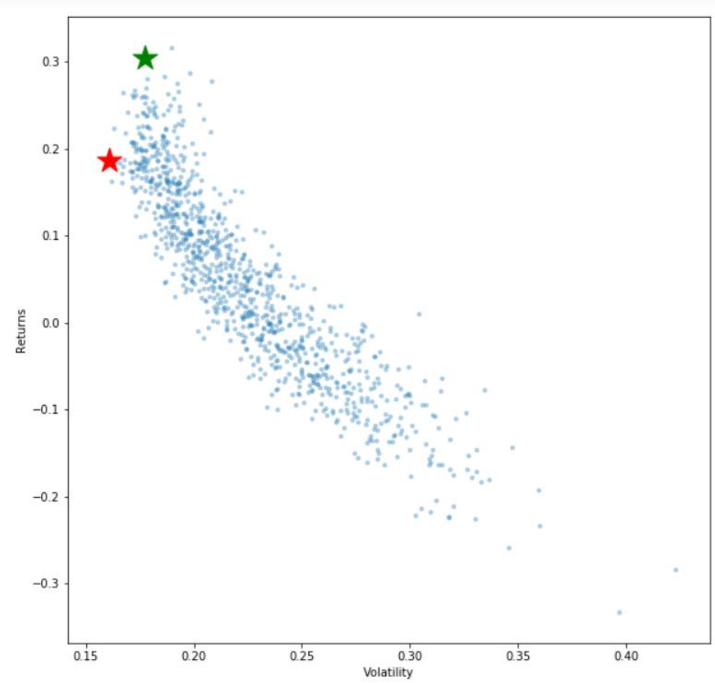

Fig. 1.3

We also tried enabling **negative weights,** which gives the **Full Returns vs Volatility plot** :-

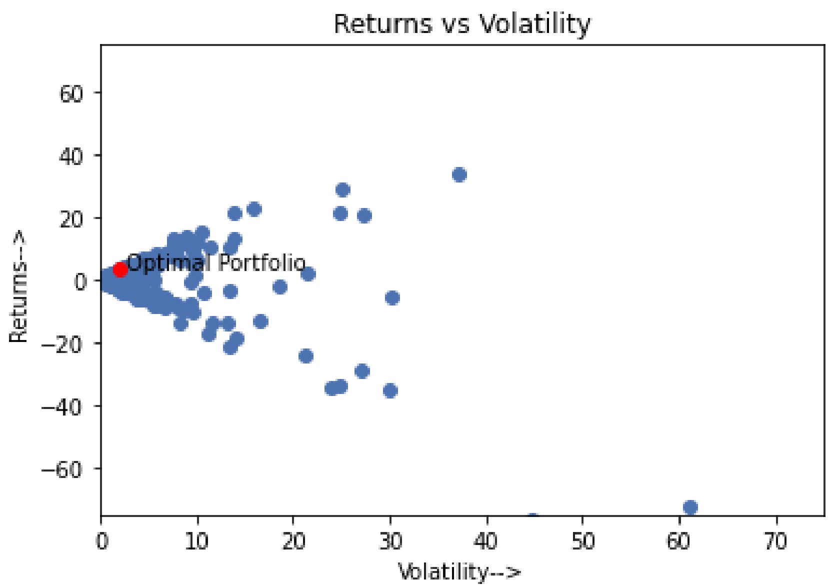

Fig. 1.4

As you can see, our **Optimal Portfolio** came out to be one with –

Returns	0.303722
Volatility	0.177114
Sharpe Ratio	1.488998
INDIGO weight	0.104453
ICICI weight	0.162098
Jet_Airways weight	0.002686
SBI weight	0.088745
TATA Chem weight	0.114897
TATA Comm weight	0.015540
HDFC weight	0.010980
Bajaj Finance weight	0.180896
KOTAK weight	0.152645
Reliance weight	0.167060

Sharpe Ratio of **1.488998** (way more than 1.00), Volatility of **0.177114** (way too stable) and Returns of **0.303722** (**0.1113179 more** than the manual one).

Finally, the last part was **Value at Risk (VAR)** for this Optimal Portfolio. Value at Risk (VAR) measures the level of financial risk within a portfolio over a specific time frame (like a year in our case). The **confidence level** in VAR determines how sure a risk manager can be when they are calculating the VAR. The confidence level is expressed as a percentage, and it indicates how often the VAR falls within the confidence interval.
As in our case, we have **95% confidence level**, it indicates we can be 95% certain that the VAR will fall within the stated interval and won't go beyond that. This metric is most commonly

used by investment and commercial banks to determine the potential losses in their institutional portfolios.

We used **Monte Carlo simulation** for this since it performs **risk analysis** by building models of possible results by substituting a variety range of values. Based on the given percentile it will show you the amount required to **cover minimum losses** for the time frame (one year in our case).

Thus, we ran **Monte Carlo** simulation for **1000 simulations** and found the amount required for covering the loss for one year, in the worst case possible. And, it came out to be $ **18.08318784231223** provided you have invested **$ 1000** in the portfolio.

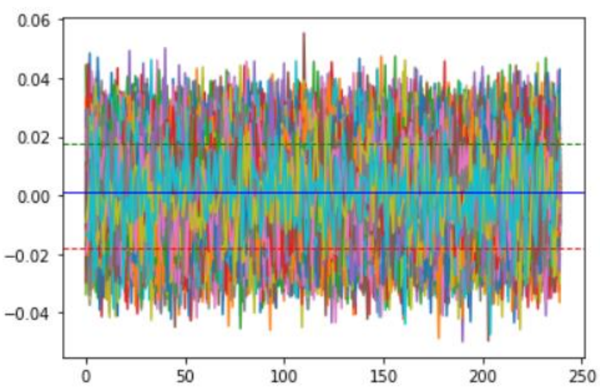

Fig. 1.5

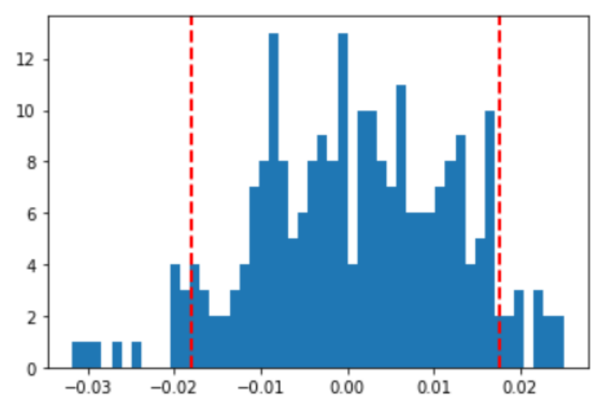

Fig. 1.6

Results:

Successful in finding the Optimum Portfolio with shown percentage of weights allocated in different companies below. With **Sharpe Ratio** of **1.488998**, **Volatility** of **0.177114** and **Returns** of **0.303722**.

Returns	0.303722
Volatility	0.177114
Sharpe Ratio	1.488998
INDIGO weight	0.104453
ICICI weight	0.162098
Jet_Airways weight	0.002686
SBI weight	0.088745
TATA Chem weight	0.114897
TATA Comm weight	0.015540
HDFC weight	0.010980
Bajaj Finance weight	0.180896
KOTAK weight	0.152645
Reliance weight	0.167060

Also, found a **very low Value at Risk** (VAR) using Monte Carlo Simulation. Results showed the amount required to cover minimum losses for one year is **$ 18.08318784231223** provided the portfolio value is **$ 1000**.

The following contains the **Python code** we developed over the course of the project:
1. https://github.com/dhruvjha206/Optimal-Stock-Allocation/blob/main/Optimal%20Allocation%20of%20Stocks.ipynb
2. Google Collab Notebook

Future work:

1. We plan to **expand the project** to cover **30+ Indian Companies Shares** and use it to model the returns of **SENSEX / S&P 500**, and understand why they are considered as a **universal benchmark quantitatively**.

2. We would also like to **upgrade** our present model and add features to account for the **efficient market hypothesis** and produce returns in an environment with a large number of "**intelligent investors**".

3. Extend the model to work for returns of **bonds**, **derivatives** and **currency markets**.

Key Learnings:

1. Knowing more about the functioning of the stock/share market.

2. Limitations in Modern Portfolio Theory.

3. Applications of Monte Carlo Simulations

4. Teamwork and Coordination

References:

1. https://finance.yahoo.com/ for **Stock Data** of various Companies.
2. Used **Investopedia** for basic terminologies.
3. YouTube for **Python** Tutorials.