To what extent does a Monte Carlo Simulation help financial analysts assess possible portfolio and stock returns?

<u>Group 27</u>

Valeri Andreev - s2697831
Oren Arbiv - s2485915
Yasin Fahmy - s2843870
Oskar Fromm - s2762234
Maxim Fraňo - s2784394
Damon Kaewborisut - s2532409

Table of Contents

Introduction

In today's financial industry, market analysts or even ordinary investors, have large amounts of sophisticated tools to aid them during their research into the markets. These tools can range from simple price prediction tools to complex sets of instruments which factor in macroeconomics and microeconomics, unique aspects of each industry, past prices and performance, etc., to give realistic predictions of whatever the user desires.

This research will be primarily concerned with predicting future price movements and in turn returns of individual stocks, as well as stock portfolios. Future price movement means the percentage change of a stock price that occurs after some period of time. After one would sell the underlying stock, this percentage change would be our return.

The method of prediction and the focus of this research is an algorithm called the **Monte Carlo Simulation**, which our team recreated in Python. You can read more closely about this algorithm in the next sections. We will be using this algorithm to simulate the future returns of stocks of multiple companies and try to elaborate on the usefulness of this algorithm to analysts and investors as the aim of this research.

Rationale behind this is to see if tools such as this can have a tangible impact on investing decisions made by individuals and to see if these decisions, affected by these tools, would actually net a positive return to the investor.

Thus our research question is **to what extent does Monte Carlo Simulation help financial analysts assess possible portfolio and stock returns?**

In the following pages a description is given of the Monte Carlo Simulation is, how it works, some history, uses and its relation to our project. Next, the research question is discussed in greater detail, what should be the limits of the 'extent', how should the algorithm actually help analysts and which parts of the result should they use to assess aforementioned returns. The section after this will include a thorough explanation of the entire algorithm including mathematical formulas for both single stock estimation and portfolio estimation. Our team has also decided to run an experiment to further solidify the usefulness of this algorithm which can be found in the following part. Then we will look and compare Monte Carlo Simulation to some other prediction tools, closely followed by investigating the use of this algorithm in fields other than finance. Lastly a wrap up with a conclusion of our findings.

Monte Carlo Simulation and our project

The Monte Carlo method was invented in the late 1940s by Stanislaw Ulam and John von Neumann while they were looking for a way to improve decision-making based on existing data during World War 2.
The method has been later named after the city of Monte Carlo, which is well-known for its casinos. The reason for that was that the modeling developed by Ulam and Neumann was counting on the element of chance, which is similar to roulette.

Monte Carlo Simulation, also known as the Monte Carlo method, is a mathematical technique used to estimate the outcomes of an uncertain event.
The simulation predicts a set of outcomes based on a range of values(population) against a set of inputs(sample).
In practice, the simulation assigns a random value to the variable that has uncertainty. The model then runs and provides results. This process is occurring over and over again while different values are being assigned to the uncertain value each time.
Once the simulation is over, the final estimation is the average of all the results.
Depending on the number of uncertainties, the simulation can execute thousands or even millions of iterations before it is completed.

Naturally, the Monte Carlo Simulation is closely related to the law of large numbers, which implies that as the sample size grows, Its mean gets closer to the mean of the entire population. In the Monte Carlo Simulation context, the more calculations with random variables that will be executed, the better the final estimation will be.

With that said, the simulation also has a few limitations concerning the required amount of data and the correlation between the variances.
To achieve relatively accurate estimation, the implementer should provide a wide range of data, since a lack of data will force the analyst to use subjective judgment that will bias the simulation which in turn will lead to inaccurate results.
Correlation between events/variables plays a major role in the Monte-Carlo simulation, hence the lack of data concerning the correlation of the variables or working under the assumption of no correlation will lead to wrong estimations.

However, these limitations can mostly be addressed using extensive data mining or data gathering. Hence, the Monte Carlo Simulation is still considered a powerful tool for analysts and is being widely used in a variety of fields such as project management, mining, manufacturing, banking, and more.

For instance, In the 1990s the US Environmental Protection Agency(EPA) started using Monte Carlo Simulation to perform risk assessments.
The EPA tried to analyze the overall health risk of smog in a city, however, there were two uncertain variables - the levels of smog in each neighborhood and the amount of time the

people spend outdoors. Given a range of values for each variable, the simulation assigned a random value to each variable and saw how they combined. After the process was repeated tens of thousands of times, it built a realistic picture of the population's smog exposure which was later used by the EPA[1].

In this project, we will explore whether the Monte Carlo Simulation can be used in the stock market to assist financial analysts to assess possible portfolios and stock returns. Considering the fact that Monte Carlo is considered a prominent tool for estimating outcomes of uncertain events and the number of uncertain variables in the stock market, we find it interesting to look into the possibility of forecasting the potential of stocks and portfolios.

[1] https://news.mit.edu/2010/exp-monte-carlo-0517

Research question discussion

Our goal of this research is to find out whether implementing Monte Carlo Simulation to estimate possible portfolio and stock returns is helpful and accurate enough to use on a regular basis to maximize the potential returns in the stock market. In theory, using the simulation would give good results because as mentioned above, as the sample size grows, Monte Carlo Simulation's mean gets closer to the mean of the entire population. This means that the result would become more accurate. In the context we are running the simulation, the sample size is immense, as we have daily returns on stocks from the earliest date which can be found on Yahoo finance.

Since the Monte Carlo Simulation results form a bell curve (normal distribution), it would give a considerable estimate where most of the simulation results fall. For example, if the tool would be used on stocks like Tesla and Ford, and one sees that based on the distribution of the results, the standard deviation of Tesla stock is two times of Ford's (NB! reality may, and probably will, differ from this example), it would give the analyst food for thought if it is perhaps better to invest in Ford rather than Tesla as it is less risky.

There is a very important aspect to consider, though. Monte Carlo Simulation, when run in the context of a stock market, assumes a perfectly efficient market. That means that it ignores important factors such as market hype, company's leadership, macro trends, etc. Because of that, the estimates portrayed by the results of Monte Carlo Simulations may not always be accurate (for example the *meme stock,* AMC*,* skyrocketed in 2021 in response to a large group of *redditors* purchasing the stock to make a statement, and there was no way this could've been predicted by running the simulation). More information concerning the limitations and the assumptions of the tool will be discussed in the following sections.

We are going to develop the simulation in Python, feed it data about certain stocks we want to get an estimate of, then monitor the movement of those stocks and conclude whether the estimation was accurate enough to be considered useful and helpful. Since the time period for testing the estimation is short, we are making our conclusion based off short-term estimations.

How does the tool work?

Mathematical Background

It is assumed that stock prices follow a random walk and that the weak form of the efficient market hypothesis is true, meaning that future returns are independent of past ones. The stochastic process of stock price development is memoryless, which is referred to as the Markov property (*Markov Property*, n.d.).

The price of today is the price of yesterday times its daily return. Since log returns are used for the simulation, to get the real return, it is needed to calculate e to the power of the logarithmic return.

$$Price_t = Price_{t-1} * e^r$$

Yesterday's stock price is known, but r is a random variable. The movement of a random variable that follows a Markov Stochastic Process can be described as the Wiener's Process or Brownian Motion (*Wiener Process*, n.d.). It allows us to model that randomness. It consists of the addition of two components, the drift (approximation of future expected daily return of stock) and stock's volatility (shock).

$$Drift = \mu - \frac{1}{2}\sigma^2$$

Z corresponds to the distance between the data point and its mean within a standard normal distribution (Number of standard deviations), also referred to as the standard Brownian motion.

$$Shock = \sigma * Z[Rand(0; 1)]$$

Based on the geometric Brownian motion (Siegrist, 2022), today's price will be:

$$Price_t = Price_{t-1} * e^{(\mu - \frac{1}{2}\sigma^2) + \sigma * Z[Rand(0;1)]}$$

For a stocks portfolio with one or multiple stocks, it will be assumed daily returns are distributed by a Multivariate Normal Distribution that uses the number of timesteps ‘time’ and number of stocks ‘stocks’ (*Multivariate Normal Distribution*, n.d.) . For sampling an daily return from the multivariate normal distribution, the correlation between every stock will be used (∑), such that there will be correlated randomness.

$Rt \sim MVN(\mu, \sum)$ where ∑ is the correlation table and μ contains the means of every stock

For sampling random daily returns for multivariate normal distribution, it can make use of Cholesky decomposition(L), normal distribution (Zt) and the mean(μ) to calculate the daily return of 1 stock:

$Rt = \mu + LZt$

$Zt \sim N(0, I)$ Where Zt are the samples from a normal distribution (I represents the Identity matrix).

μ has the shape of a matrix with ‘time’ rows and ‘stocks’ columns, (‘time’, ‘stocks’), L has (‘stocks’, ‘stocks’) and Zt has (‘time’, ‘stocks’). The dot product of L and Zt is not possible, because the number of columns of L is not equal to the number of rows of Zt. Zt must be transposed, such that the dot product of L and Zt becomes (‘stocks’, time), then we need to transpose it back to (‘time’, ‘stocks’) such that LZt can be added to μ.

all daily returns are calculated for 1 timestep for multiple stocks. the stock allocation determines daily returns of the whole portfolio, so the daily log return on that time step is:

$R_{dailyX} = \sum_{i}^{N} e^{(w_i * R_i)}$ where N is the number of stocks you have in the portfolio, i is the stock with the given weight and daily return and X is the timestep.

finally, with the use of multiple daily returns over a period, the returns over period T can be calculated by multiplying each other and for price at day T:

$R_{dayT} = \prod_{i}^{T} R_{dailyI}$ where $RdayT$ is the return between day 0 till day T

$price_T = price_0 * R_{dayT}$ where T is the day which you want to calculate the price

Single Stock Simulation Algorithm

After importing all the necessary libraries, the historical adjusted stock price data is fetched from yahoo.finance. After that, the data is transformed into logarithmic daily returns. Based on that, the mean and variance are calculated in order to determine the drift component.

To calculate the shock component, the standard deviation of the log returns is computed and multiplied with a data frame consisting of Z-values based on a random probability. The number of rows and columns in the data frame correspond to the number of days to simulate and iterations of the simulation, respectively. In order to get the actual daily returns, the log returns are converted back into arithmetic values, by taking all the values to the exponent of e (Euler's number).

A new data frame, with the same size is created, to simulate the prices. First, the first row is filled with the most recent stock price available. With the help of a loop the next price is calculated by multiplying the last price with the simulated daily return.

To showcase the lower and upper bounds of the confidence interval with 95% certainty, the returns were calculated in a similar way as for the simulations, but the shock component was not random this time. Instead the standard deviation was multiplied by +/- 1.96 to achieve the upper and lower bounds. For the expected return, each price is calculated by multiplying the precedent price point with the mean return.

Portfolio Simulation Algorithm

We will be using the same libraries as the single stock simulation, but we will be using multivariate normal distribution to simulate a stock portfolio. We first need to fetch the stock prices of the preferred stock and calculate the means and covariances to determine the daily returns per timestep in a simulation.

With the given stock allocation(number of stocks 'stocks'), number of Monte Carlo Simulations('sims') and number timetsteps ('time'), we can determine for every simulation will go per timestep with the use of daily returns when assuming we are using Multivariate normal distribution ($R_t = \mu + LZ_t$), We first prepare an 2D array with 'time' rows and 'stocks' columns where in every row will be the corresponding means in every stock in a column, then we will determine the cholesky decomposition and finally the random sample of the normal distribution for every timestep and stock(matrix ('time', 'stock'),

As explained in the background, to make use of the daily returns function, we need to transpose Z_t and the dot product of L and Z_t, after transposing LZ_t, we can add the μ to the transposed LZ_t to get the daily returns of all the time step of every stock for 1 simulation. Then we can use the function 'np.cumprod' to generate a list to simulate the portfolio value with the given weights and daily returns per timestep.

We will do this process 'sims' times and we can then plot all the Monte Carlo simulations onto 1 plot and see all the simulated portfolio changes over time.

Optimizing stock portfolio

It is also possible to calculate the optimized stock allocation of given stocks and time range. We used the library PyPorfolioOpt' which contains implemented portfolio optimization methods for choosing the optimal portfolio and we made use of the Efficient frontier algorithm that returns the stock allocation (*Pyportfolioopt · PyPI*, n.d.).

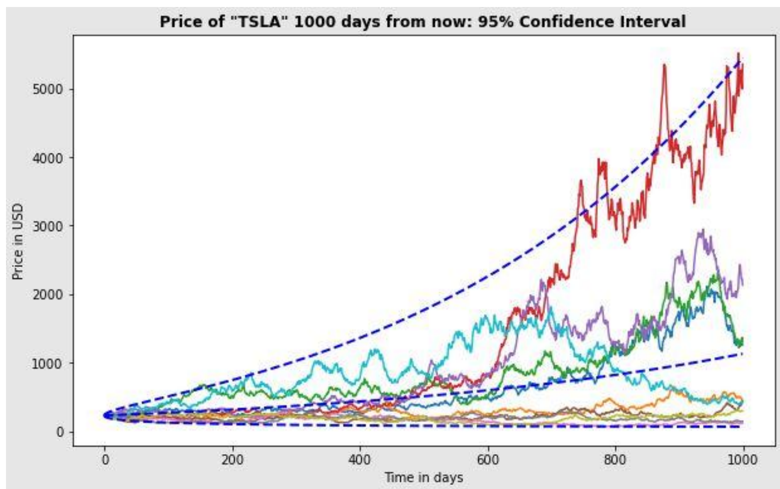
To use the Efficient frontier algorithm, the mean and variance must be calculated first from the fetched daily returns of time range of choice and preferred stocks. The efficient frontier algorithm can return the max sharpe ratio with the given yearly volatility, expected annual return and most importantly, the weight distribution of the stock allocation.

The optimized stock portfolio can also be seen in terms of stock holdings from given funds. The efficient algorithm will calculate the stock distribution, then we check the latest prices of the preferred stocks and finally we will make use of the function 'DiscreteAllocation' to return the stock portfolio with the amount of stock we own and the funds remaining.

Limitations and Assumptions of the tool

Explaining the past is much easier than predicting the future. As briefly mentioned in the research question discussion, in spite of the confidence interval that was applied in the tool, the Monte Carlo simulation has limitations that can affect the accuracy of the results. These drawbacks concern the fact that the simulation only considers past price data and does not take into account macro economical scenarios or black swan events. When considering only price data, plenty of additional important variables which directly affect the stock price or portfolio are naturally ignored, which may result in inaccurate estimations.

The simulation does not consider macro economical scenarios such as inflation, unemployment rate and industrial output which may correlate with the stock price. As for the confidence interval itself, It also has drawbacks, as the confidence interval will only be valid if the market will stay the same in the upcoming period, as it was during the time which the data was taken. For example, in case the data has been taken last year but in the upcoming future a recession will take place, the volatility will be higher for every stock (VIX index) than it was at the time that the data was taken, hence affecting the accuracy of prediction.

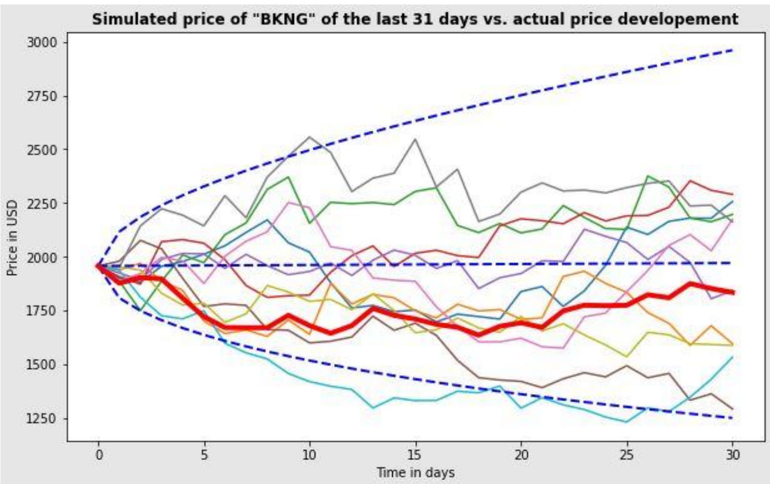


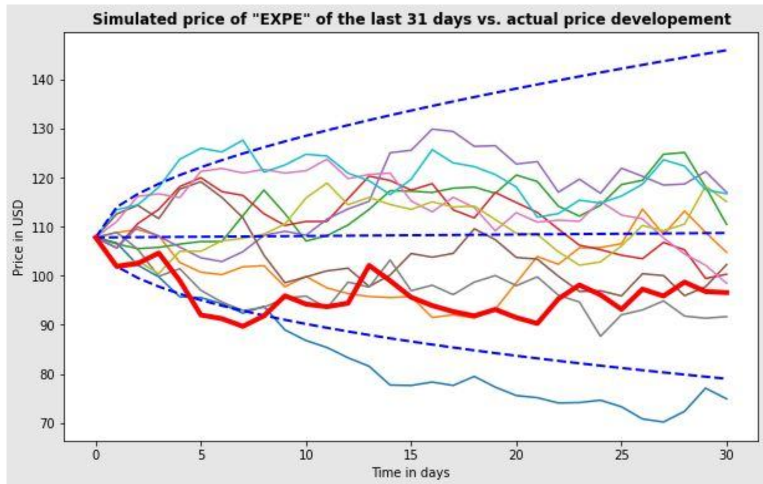
Moreover, if the stock has experienced massive growth over the period, the data was fetched from, the simulation will assume steady growth for that stock and reflect unrealistic future scenarios. Therefore it is important to feed the algorithm a lot of data, from a long period, to account for this. For the simulation of Tesla for example, sometimes the simulated stock developments were so extreme that Tesla's stock would reach a stock price of thousands of dollars after a 1000 days period (current stock price: 225$, 27.10.2022). That is because Tesla has experienced extreme stock price growth since 2020.

In theory, a Monte Carlo simulation is able to incorporate autocorrelations like inflation and non-normal stock returns, but the constraints are most often set by the creator of a Monte Carlo simulation tool, often due to missing or inaccurate data sources. If one decides to still take those factors into account, it is recommended to do that with proper care and choose high quality inputs, because the principle of “garbage in, garbage out” also applies in this case.

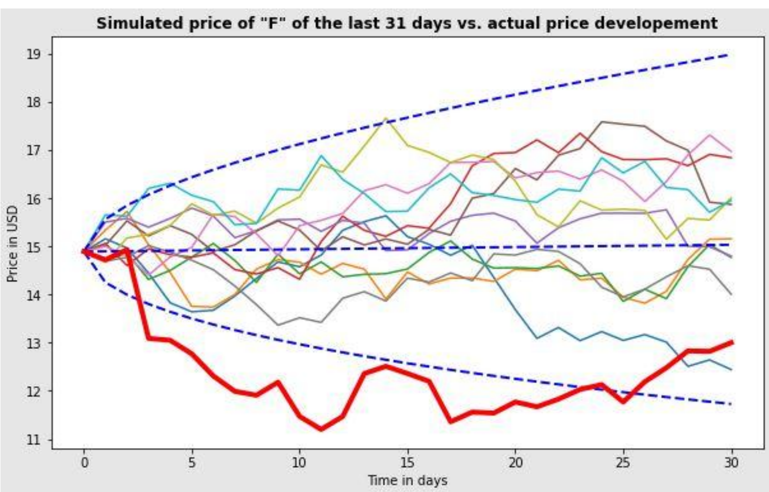
Experiment & Results

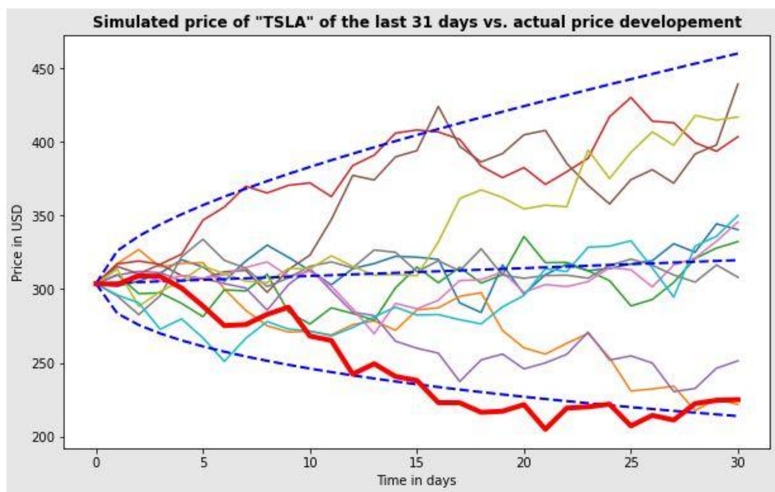
These pictures show the simulated stock price paths vs the actual development (red line) of the stock. The simulated period was from 15.09.2022 until 27.10.2022, 31 days that the stock exchange was open. The dashed blue line represents the upper and lower bound of the confidence interval for the price, with a certainty of 95%, as well as the expected price, shown in the middle.



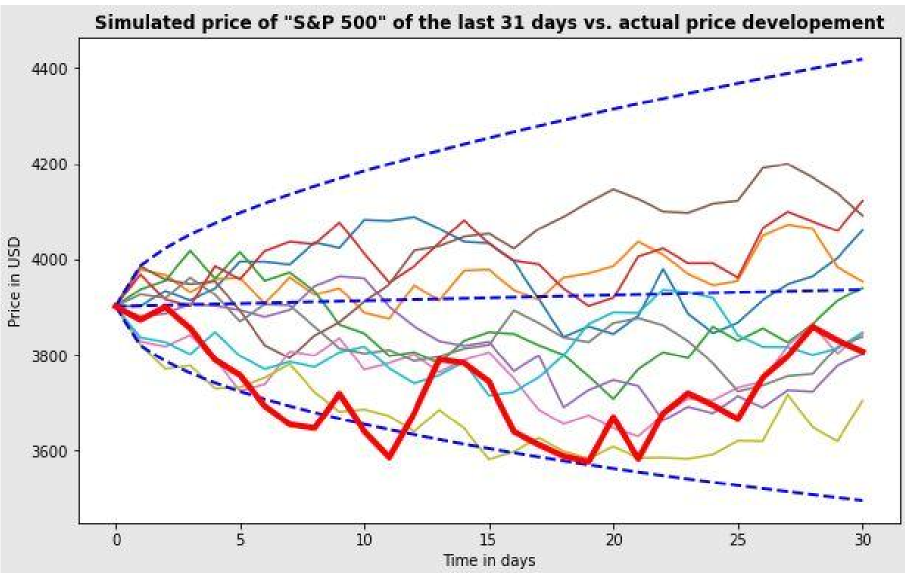


Since Booking and Expedia are both travel agencies and belong to the same industry, it makes sense that they show a similar stock price development. Almost all price points were captured in between the confidence intervals.





Ford and Tesla, as well, belong to the same industry. The stock prices of both experienced heavier losses than the ones from the travel agency industry. Moreover, it seems like both single stocks do not share a similar price development, as it was with the two other companies above. Ford experienced a large drop and was able to recover in the lower region of the confidence interval. For Tesla, it was more a linear and steady decrease in price, ending up almost exactly on the lower bound of 95% certainty.



The benchmark S&P 500 in that period showed a decreasing development, which shows the overall trend in American stocks. Comparing this result to the four single stocks, there is definitely a correlation to be observed, since all stocks, as well as the index moved in the lower half of the confidence interval.

Monte Carlo Vs other prediction tools/models

In order to search for the answer to our research question of to what extent the Monte Carlo Simulation helps financial analysts assess possible portfolio and stock returns, we are going to compare the Monte Carlo Simulation with other prediction tools or models. Of course, there are hundreds of thousands of prediction tools and data models which have the same purpose as the Monte Carlo Simulation. Most of them, as well as Monte Carlo, have their own way of doing it using past data, different ways of modeling, and optimizing by different methods. In order to show some advantages and disadvantages of different prediction tools and models in comparison with the Monte Carlo Simulation, we are going to select some of the most popular tools and compare them with the Monte Carlo Simulation.

In order to get a better perspective on how important a Monte Carlo Simulation is, we can take into consideration another way of modeling - “Second Order Reliability Modeling” (SORM). As it is stated in the Probabilistic Engineering Design textbook, chapter 7 “First Order and Second Reliability Methods”, the Monte Carlo Simulation is used to compare the accuracy of the results from SORM. This demonstrates the reliability and accuracy of the model and by this statement, we can confidently argue that Monte Carlo Simulation can be considered more accurate relative to other prediction tools used in the financial market.

When considering predictions of the market, we wish to compare the simulation with a popular indicator. Even though it is an indicator more than a specified tool, we want to compare the Moving Averages (MA) with the Monte Carlo simulation. Their purpose is to help level the price data over a specified period by finding an average price.
The following table describes the differences between the Simple moving average(SMA), the Empirical moving average (EMA), and the Monte Carlo simulation.

	Simple moving average (SMA)	Empirical moving average (EMA)
Description	A way of calculating the arithmetic mean of given set of prices over specific number of days in the past with all the data equally weighted	Same as SMA, but gives a bigger weight to the latest trends in the market since they are perceived as the most valuable ones which determine the behavior of the market.
Advantages of Monte Carlo Simulation	● The biggest advantage of Monte Carlo Simulation over SMA and EMA is that it can also be used for a portfolio consisting of multiple stocks, not just for a single stock. ● SMA and EMA solely provide indicators that suggest whether it's recommended to buy or sell a stock and an extra orientation based on some average prices. However, the Monte Carlo simulation predicts a set of outcomes based on a range of values(population) against a set of inputs(sample) which is considered more powerful information for experienced financial analysts. ● EMA’s weighting system allows it to respond more quickly to price changes, but this makes it more vulnerable to false signals (What Are the Main Advantages and Disadvantages of Using a Simple	

	Moving Average (SMA)?, 2022). The Monte Carlo simulation, on the other hand, cannot be manipulated that easily since the simulation assigns a random value to the variable that has uncertainty, and depending on the number of uncertainties, the simulation can execute thousands or even millions of iterations before it is completed.
Disadvantages	● As it is stated above, to achieve relatively accurate estimation using Monte Carlo simulation, the implementer should provide a wide range of data, since a lack of data will force the analyst to use subjective judgment that will bias the simulation which in turn will lead to inaccurate results. In the case of SMA and EMA, the indicators can work properly even with a limited amount of data and no restrictions concerning the amount of time or data.

When considering prediction tools, we must conclude that many online tools belonging to fast-growing companies are using Monte Carlo simulation behind their services exactly due to the benefits which the simulation brings. One of the best examples is the company Palisade, a leading provider of risk and decision analysis software for three decades. In their article about Monte Carlo simulation, they said: “Probability distributions are a much more realistic way of describing uncertainty in variables of a risk analysis, making Monte Carlo simulation far superior to common “best guess” or “best/worst/most likely” analyses”.

To conclude, one can argue that Monte Carlo simulation is one of the most powerful tools In the sphere of market prediction and decision-making tools.
The simulation is a relatively reliable tool that can assist financial analysts in assessing possible portfolios and stock returns to a higher extent than other traditional prediction tools such as EMA and SMA.
Considering the aforementioned advantages, Monte Carlo is definitely one of the best and most useful tools for stock market predictions.

Additional industry applications of Monte Carlo

Monte Carlo Simulation has found its use in many industries other than finance. In this section we will look at some of the applications of Monte Carlo Simulation and see why it can be useful to experts or systems in these fields.

Monte Carlo Simulation has found its niche in **computer science**, mainly in the gaming industry. Here it is most prominently used in two different categories. The first is aiding artificial intelligence algorithms in video games. The video games in question are mainly board or strategy games, where the computer player needs to make the best move possible. These types of games are perfect for Monte Carlo because they have a so-called search tree of possible moves. Then, every path of this search tree is simulated by Monte Carlo, or in this case its derivative “Monte Carlo tree search”, to estimate the long-term potential of every move. These estimates are then used by the computer player to make a move in the game. The second use of Monte Carlo Simulation in this industry concerns computer graphics. Here it is used in a method called Path tracing which is a method used to make light reflecting off 3D surfaces as realistic as possible. It is currently one of the most accurate graphics rendering methods.

In terms of general **engineering** Monte Carlo Simulation can be used in fields such as telecommunication, where Monte Carlo simulates different scenarios whose main variables are the number of concurrent users using the network, the services they are requesting or even their location. Robotics also uses Monte Carlo to simulate the positioning of a robot. The energy industry can use Monte Carlo when trying to estimate the lifetime of wind farms where there are many different scenarios and variables affecting the life expectancy of these turbines.

Rescue services have found their own use for Monte Carlo Simulation, mainly in the US, where it is used to calculate the probable locations of vessels that have gone off radar during search and rescue operations. It is effective in most notably saving lives, but also in saving resources by sending the rescue teams to the location where to ship might actually be so no guesswork a fuel wastage has to occur during rescue.

The last example, but certainly not the last application, would be **biology**. Here Monte Carlo Simulation is used in the field of Computational Biology, which is a field concerned with using data analysis and mathematical modeling to understand relationships in biological systems. Monte Carlo is mainly used in studies of genomes, proteins and membranes where computer simulations can allow us to monitor the local environment of a particular molecule to for instance see if some reaction is occurring within it. These simulations prove to be a good alternative to when actual physical experiments are not possible.

In conclusion to this section, we can find the use of Monte Carlo Simulation in almost any field where there is a necessity to simulate some phenomena with significant uncertainty.

Conclusion

The Monte Carlo simulation is named after the gambling hotspot in Monaco because chance and random outcomes are central to this modeling technique (the more calculations with random variables and parameters are executed, the more accurate the outcome will be), as they are to games like roulette, blackjack, and slot machines, to name a few. It was first introduced all the way back in the 1940s, and the algorithm has stayed relevant to this day. The Monte Carlo simulation is mostly used in the industries where one can run into problems that have high uncertainty, in fields such as engineering and technology.

In our project, our initial assumptions were that using the simulation, one could get a rather good indication of the potential stock price movement if normal market times and steady growth is assumed, and granted that there are no significant unpredictable happenings, macro economical scenarios and uncertain events in the stock market.

The Monte Carlo simulation developed throughout the project has significant advantages over other modern prediction tools, mainly due to its ability to evaluate both portfolios and single stocks. In theory, a Monte Carlo simulation is able to incorporate autocorrelations like inflation and non-normal(Black-Swan events) stock returns, but the constraints are most often set by the creator of a Monte Carlo simulation tool, often due to missing or inaccurate data sources.

Now, to answer our initial research question, if implementing Monte Carlo Simulation to estimate possible portfolio and stock returns is helpful and accurate enough to use on a regular basis to maximize the potential returns in the stock market, the answer is YES. Based on the results under section “Experiment & Results”, one can notice that by the end of the 31 days of observation, every stock remained within the confidence interval and the simulation was fairly accurate for most of the stock prices simulated. Of course there were bigger and lesser deviations from the actual stock price, but in the overall picture one could observe that even taking into account the unpredictable events, the stock prices’ movements stayed relatively true to the simulation.

Due to market efficiency and influence by external events, which sometimes cannot be foreseen(such as Covid-19), there will never be an algorithm that will assist one to consistently beat the market- this is simply impossible(for now). However, the tool provided important insights that can improve decision making processes in the stock market, it showed the approximate price deviation and movement direction of the stocks, which is something that is incredibly useful in decision making being a financial analyst or just a regular investor.

And now a bit about our project experience. We were delighted by how synergetic our teamwork was throughout assignments and in the project itself. Tasks were executed on time and there were no irreconcilable opinions, everyone was on the same wavelength. Each member of the group played an important role in completing the assignments and the project, and tasks were divided equally, so that every member could have the chance to include their thoughts and skillset. This concludes the report of Module 5 Finance for Engineers project 2022.

References

About Palisade – The Leader in Risk and Decision Analysis Software | Palisade. (2013). Palisade. https://www.palisade.com/company/

Du, X. (2005). *First Order and Second Reliability Methods*. https://pdesign.sitehost.iu.edu/me360/ch7.pdf

Explained: Monte Carlo Simulations. (2010, May 17). *MIT News | Massachusetts Institute of Technology*. Retrieved October 19, 2022, from https://news.mit.edu/2010/exp-monte-carlo-0517

Moving Average (MA): Purpose, Uses, Formula, and Examples. (2022). Investopedia. https://www.investopedia.com/terms/m/movingaverage.asp

What are the main advantages and disadvantages of using a Simple Moving Average (SMA)? (2022). Investopedia. https://www.investopedia.com/ask/answers/013015/what-are-main-advantages-and-disadvantages-using-simple-moving-average-sma.asp

What is Monte Carlo Simulation and How Does it Work | Palisade. (2021). Palisade. https://www.palisade.com/monte-carlo-simulation/

Monte Carlo method. (2022, 10 27). Wikipedia. Retrieved October 28, 2022, from https://en.wikipedia.org/wiki/Monte_Carlo_method

Harper, D. (n.d.). *How to use Monte Carlo simulation with GBM*. Investopedia. Retrieved October 19, 2022, from https://www.investopedia.com/articles/07/montecarlo.asp

Markov property. (n.d.). Wikipedia. Retrieved October 27, 2022, from https://en.wikipedia.org/wiki/Markov_property