J.P. Morgan Quant Mentorship Program, 2022

Case Study Challenge

Team 7
Khushi Singhvi
Divyangna Sharma

Monte Carlo Simulation Key Points

- Monte Carlo Simulation is a technique for estimating a variable that depends on one or more random factors.
- The **Law of Large Numbers** can ensure that the estimated value obtained by the Monte Carlo method can gradually converge to the actual value according to the increase in the number of simulations.
- The **Central Limit Theorem** can give the error estimation range of the Monte Carlo model.
- Monte Carlo Method doesn't work for options which have a possibility of getting exercised before their maturity date (for example, in case of American Options) as in such options there is a need to perform a Monte Carlo simulation on a Monte Carlo simulation which would be computationally unfeasible.

Law of Large Numbers

The law of large numbers states that as a sample size grows larger, its mean approaches the population's average.

Central Limit Theorem

The central limit theorem (CLT) claims that as the sample size grows larger, the distribution of a sample variable approaches a normal distribution (i.e., a "bell curve"), assuming that all samples are identical in size and independent of the population's actual distribution shape.

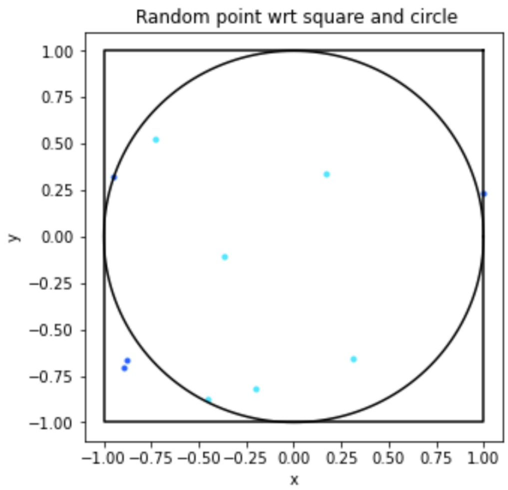
Question-1.

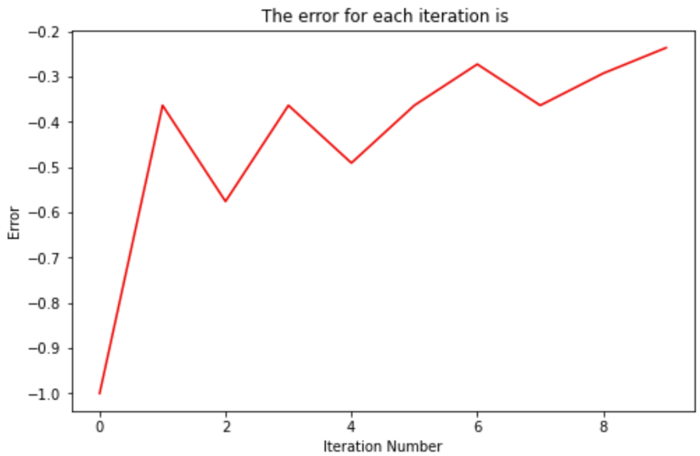
Goal: To estimate the value of pi using Monte Carlo Simulation

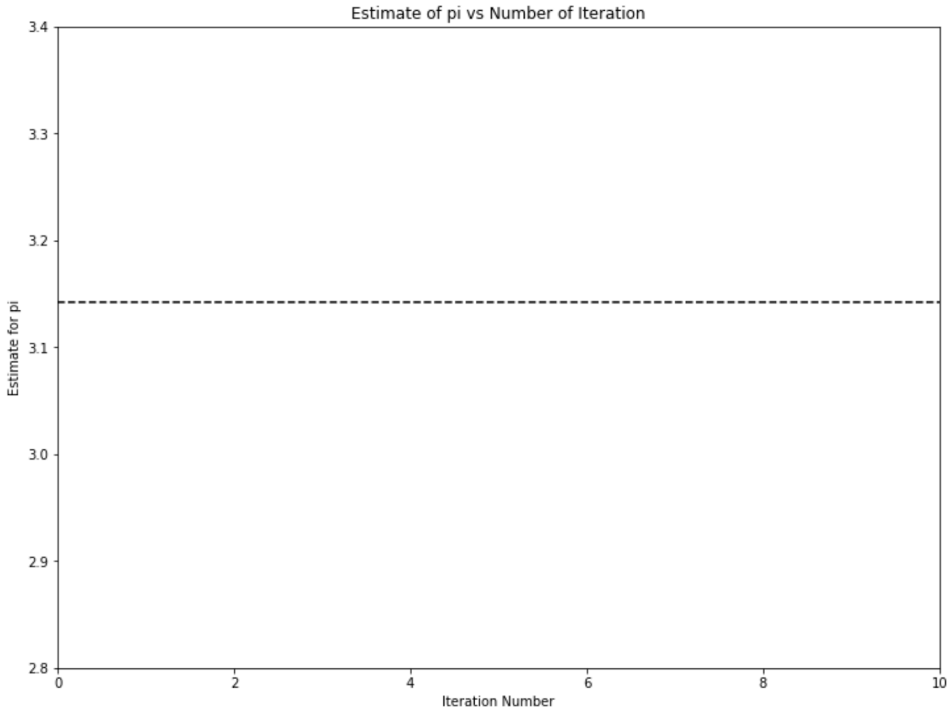
//Please refer to the code for Question 1 in the attached Jupyter Notebook.

- A square of side length 2 with an inscribed circle in it of radius 1 was drawn.
- Set of random values for x and y depending on the number of iterations such that they lie in the square were generated.
- If the generated point lay in the circle the count of points in circle was increased or else the count of points in square was increased.
- Area of circle = pi
- Area of square = 4
- Ratio of area of circle to area of square will almost be equal to the ratio of points that lie in circle to the ratio of points that lie in square.
- $\pi = \frac{Points\ Inside\ Circle}{Total\ No\ Of\ Points * 4}$

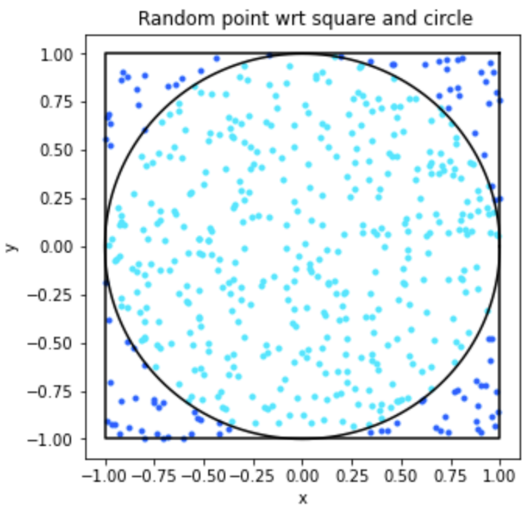
Graphs for number of iterations = 10:

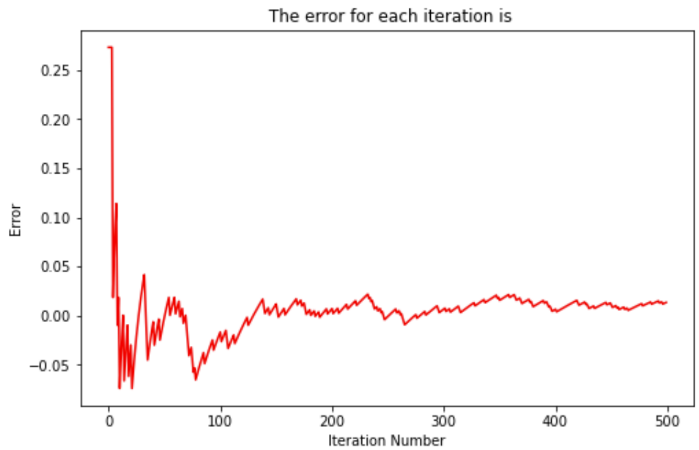


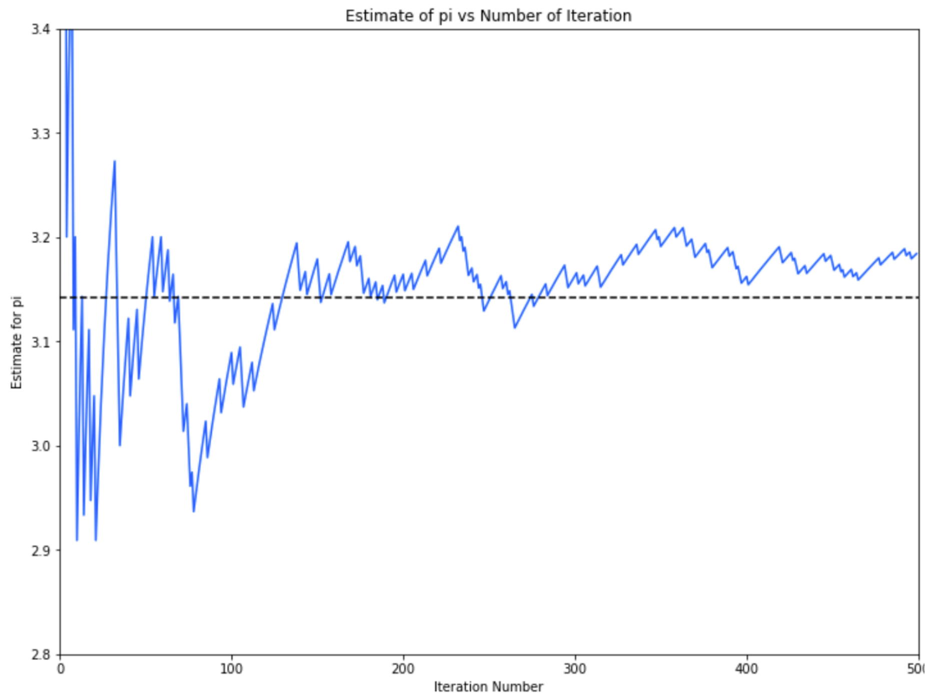




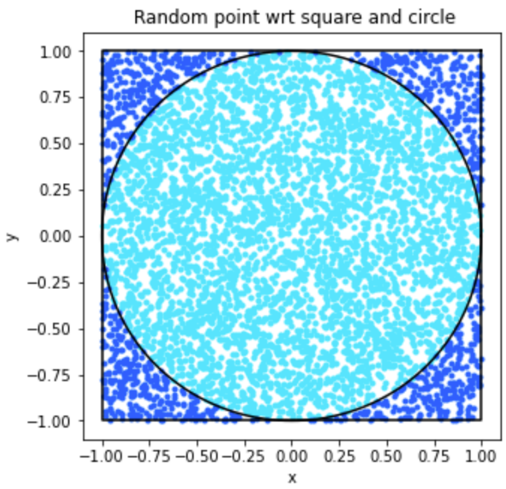
Graphs for number of iterations = 500:

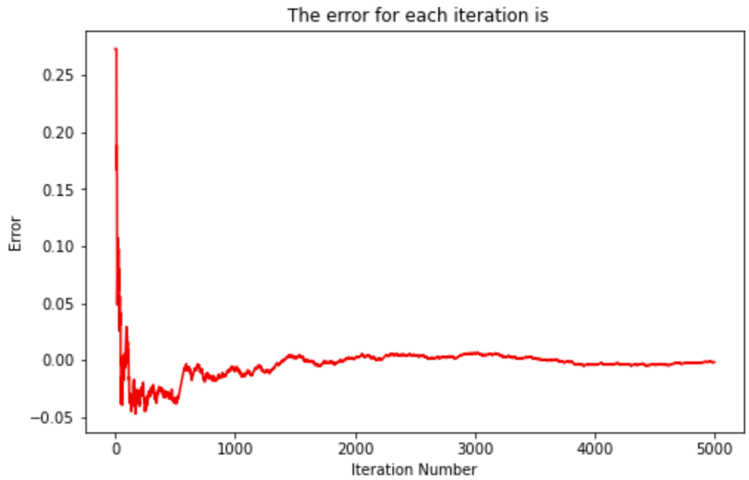


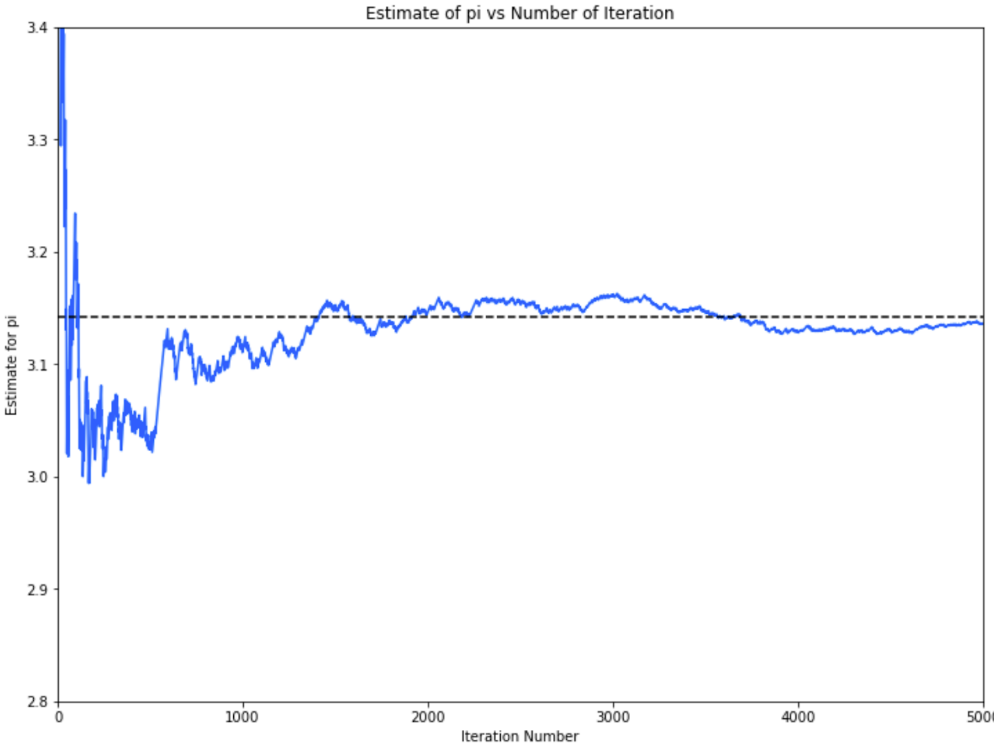




Graphs for number of iterations = 5000:







From the above plots we can see that the the estimated value of pi converges to the actual value as we increase the number of iterations. The error also decreases and becomes almost constant.

The error in the Monte Carlo Method in comparison to the actual value of pi:

Error = $\frac{abs(3.137-3.1416)}{3.1416} * 100 = 0.14\%$

Question-2.

Given a company XYZ capital trading at 100 rupees (at time = t_0) whose stock price moves up and down by 1 rupee with equal probability at each minute.

Let S be the Stock price of the company at the time t.

At $t=t_0$, S = 100 rupees

Transitions: 1.) S+1 with Probability(P) = 0.5

2.) S-1 with Probabilty(Q) = 0.5

Part A

Goal: To find the Expected value of the stock price after a given time.

The expected value refers to the weighted average of all possible values of a random variable. The expected value formula is given as:

$E(f(x)) = \Sigma(P(x) * f(x))\ \forall\ x$

//Please refer to the code for Question 2, Part A in the attached Jupyter Notebook.

(i) The expected value of stock price after 1 minute:

Sol: At $t=t_1$, there are 2^1 possible values for the stock price which are:

-> S_{t1} = S+1 = 101 rupees

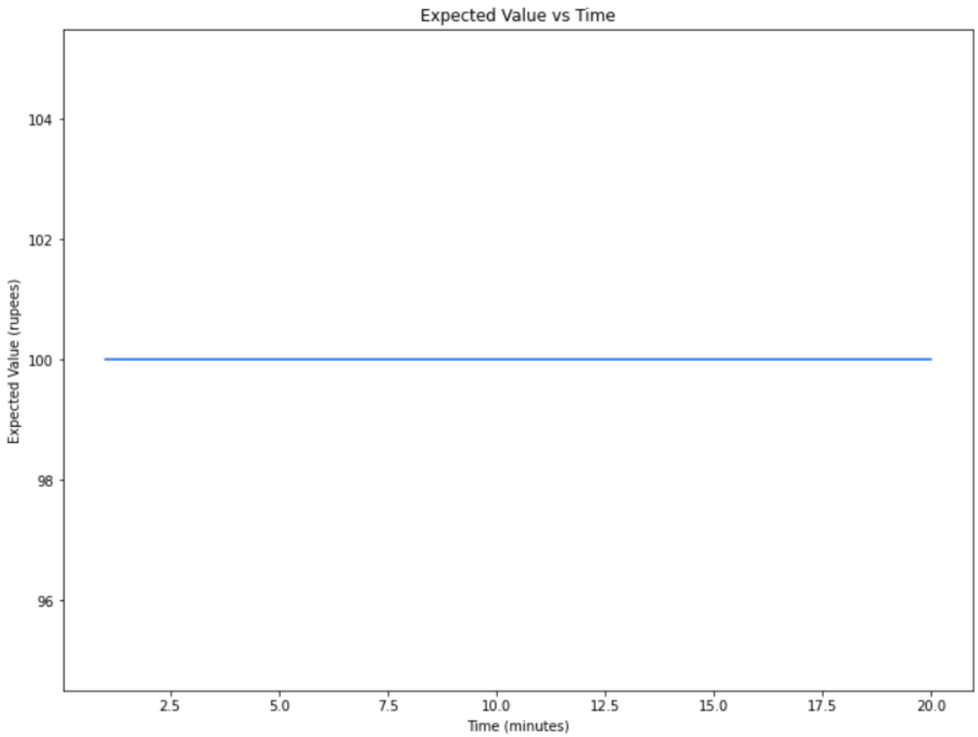
-> S_{t1} = S-1 = 99 rupees

Thus the Expected value of the stock price after 1 minute will be:

$E(t_1) = (\frac{1}{2} * 101 + \frac{1}{2} * 99)$

$E(t_1) = \frac{1}{2}(101 + 99) = \frac{1}{2}(200)$

$E(t_1) = 100$



(ii) The expected value of stock price after 10 minutes:

Sol: At $t=t_{10}$, there are 2^{10} possible values for the stock price and the expected value is the weighted average of all the 2^{10} possible values.

(iii) The expected value of stock price after 1 hour:

Sol: Since there are 60 minutes in an hour thus, at $t=t_{60}$, there are 2^{60} possible values for the stock price and the expected value is the weighted average of all the 2^{60} possible values.

(iv) The expected value of stock price after 1 month:

Sol: On average, there are 43,804 minutes in a month thus, at $t=t_{1\ month}$, there are 2^{43804} possible values for the stock price and the expected value is the weighted average of all 2^{43804} possible values.

We can observe from the graph that the answer for the **Expected value** to all the above-stated 4 parts will come out to be equal which is **100**. This is because, after each minute, the net increment or net decrement in the value of the stock price is 0 because both the transitions have an equal probability as well as an equal value of change(+1 or -1) in the stock price.

According to the Law of Large Numbers, the difference between the theoretical probability of an event and the relative frequency approaches zero as the number of trials in a probability experiment rises.

[Part B]

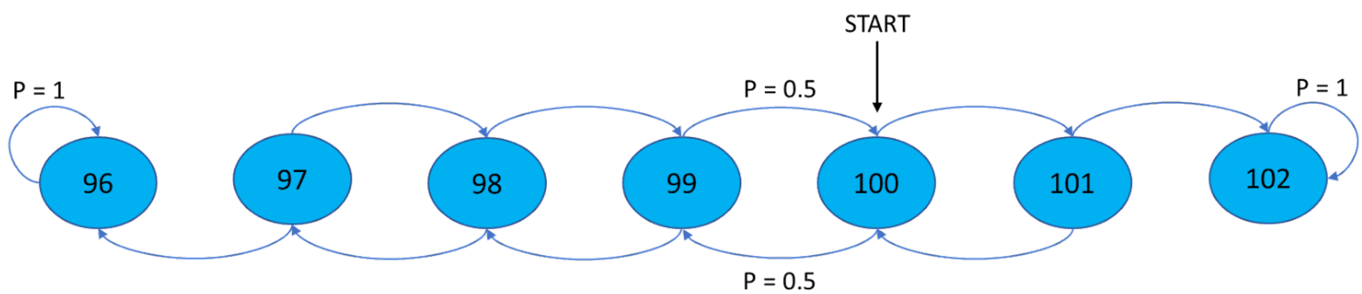
Goal: To find the probability of the stock price starting from 100 rupees and going to 102 rupees before going to 96 rupees.

<u>**(i) Analytical Solution**</u>

The given problem is a **1-dimensional random walk** on the **Markov Chain.**
A Markov chain is a mathematical process that transitions from one state to another within a finite number of possible states. An important property of the Markov Chain is that the next state depends only on the current state and not the states before.

Mathematically, $\mathbf{P(X_{n+1} = x \mid X_1=x_1, X_2=x_2,...X_n=x_n) = P(X_{n+1} = x \mid X_n=x_n)}$
where n is the current state.

Probability Transition Graph



The above probability transition graph shows a random walk on the Markov Chain with 7 states, {96,97,98,99,100,101,102}, where:

- State 96 and 102 are the absorbing states (i.e., the process ends if we reach either of them) while the rest of the states are transient states.
- State 96 is a losing state whereas State 102 is a winning state.
- For any State i (where i ϵ [97,101]), there are two possible paths(forward or backward) with an equal probabilities of 0.5.

Let i and j denote the current and next step respectively. Let P_{ij} be the probability of going to state j from state i.

Transition(i -> j)	Probability from i -> j (P_{ij})
If i=j=96 OR i=j=102	1
If j=i+1, i ϵ [97,101]	0.5
If j=i-1, i ϵ [97,101]	0.5
Otherwise	0

As the stock price reaches either 96 or 102, it will stop at that particular state and hence the probability of that transition is 1.

Since all the states except 96 and 102 are transient thus after a finite amount of time, the stock price would attain one of the absorbing states and the process would end.

Now, let's say we start with the initial value of stock price as i.
Let Pi denote the probability of the stock price reaching N rupees before going to 0.
If i = 0, $P_0 = 0$,
If i = N, $P_N = 1$.
For iϵ[1,...N-1], $P_i = 0.5*P_{i+1} + 0.5*P_{i-1}$
Solving this system of equations for iϵ[1,...N-1], we get
$P_i = \frac{i}{N}$
which means that if we start with i as the stock price, the probability that the stock price will reach N before going to 0 is, $\mathbf{P_i = \frac{i}{N}}$.

In our case starting from 100 and going to 102 before reaching 96 is equivalent to starting from 4 rupees and going to 6 rupees before 0.

Thus, the probability of the stock price reaching 102 before going to 96 would be:
$\mathbf{P_{req} = \frac{4}{6} = 0.6667}$

(ii) Monte Carlo Method

//Please refer to the code for Question 2, Part B in the attached Jupyter Notebook.

The random variable, in this case, is a transition of +1 or -1 after every minute. For this, a random number x is generated between 0 and 1. Now since both the transitions have an equal probability, thus starting with the initial stock price S,

If x>0.5, S = S+1
else S = S-1

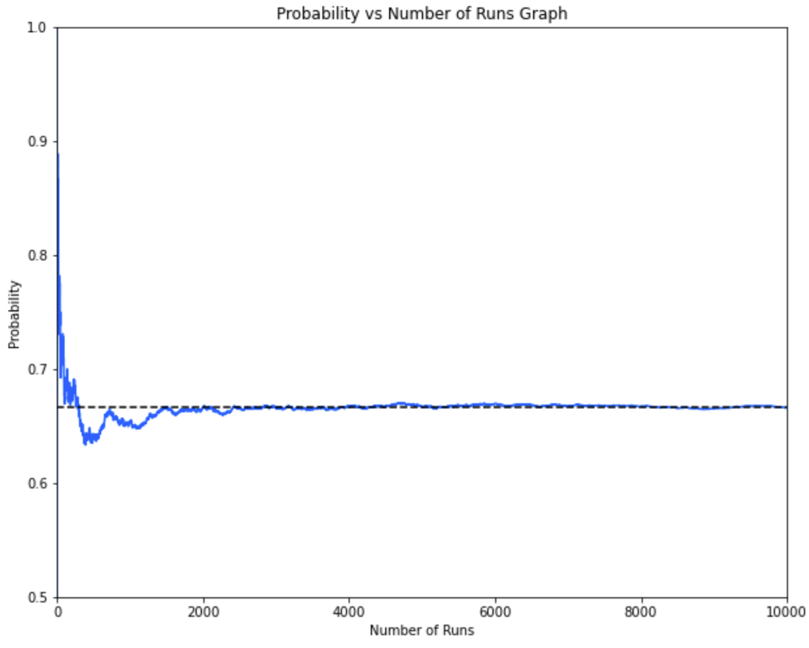
We keep a check on whether the Stock price is hitting 96 before it hits 102. If that happens we increase the value of the variable named failure by 1 else we increase the value of the variable named success by 1.

The probability is calculated as $P_{req} = \frac{No.\,of\ successes}{No.\,of\ successes + No.\,of\ failures}$

We simulated the process for a different number of runs until it converged to a value of 0.6733 which is below the tolerance limit stated in the problem statement.

```
Number of runs required for a converged vale: 3001

The estimated probability is: 0.6659152609879686
```



The error in the Monte Carlo Method in comparison to the Analytical solution:

Error = $\frac{abs(0.6667-0.6659)}{0.6667} * 100 = 0.11\%$

Note: we assume that minimum number of runs should be greater than 3000 as there's a possibility we might get error less than 10^-2 before that due to the high disturbance. Since we want a value after which it converges and by looking at the graph for various runs we assume it to be greater than 3000.

Question-3.

Goal: To calculate the value of the Call Option Premium using the Black-Scholes Formula.

For the company XYZ capital, the given parameters of the Option contract:

Spot price at time t_0 (S_0) = 200 rupees
Strike Price(K): 180
Time to expiration(T): 30 days (1/12th of a year)
Implied Vol(σ): 0.15
Interest Rate(r): 0.02
Type of option: Call

Black Scholes Merton (BSM) is a mathematical model which gives the fair value of the financial option. It comes with a number of assumptions which are as follows:

- There are no dividends paid during the life of the option.
- There are no transaction costs in buying the option.
- The markets operate under a Markov process in continuous time.
- The risk-free rate and volatility of the underlying asset are known and constant.
- The returns on the underlying asset are log-normally distributed.
- Applies to European style options only.

The Black Scholes Formula for a Call option is:

$$C(S,t) = N(d_1)S - N(d_2)Ke^{-r(T-t)}$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right]$$

$$d_2 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(T-t)\right]$$

$$= d_1 - \sigma\sqrt{T-t}$$

where,
$C(S,t)$ - BSM price of a call option at time t and stock price S(in \$)
S - The stock price at the time t (in \$)
σ - Stock volatility (absolute, not percent)
K - Option strike price (in \$)
(T-t) - Time to maturity (in years)
N - CDF of the standard normal distribution
r - the annual risk-free rate of interest (absolute, not percent)

Let the price of the Call Option Premium be C, Then, using the above formula we calculate

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2}) * (T - t)\right]$$

$\Rightarrow$ $$d_1 = \frac{1}{0.15\sqrt{\frac{1}{12}}}\left[ln(\frac{200}{180}) + (0.02 + \frac{0.15^2}{2}) * (\frac{1}{12})\right]$$

$\therefore$ $d_1 = 2.4933$

And from the CDF table,

$N(d_1) = 0.9936$

Now, $d_2 = d_1 - \sigma\sqrt{T - t}$

$\Rightarrow$ $d_2 = 2.4933 - 0.15\sqrt{\frac{1}{12}}$

$\therefore$ $d_2 = 2.4500$

And from the CDF table,

$N(d_2) = 0.9928$

Thus, the premium of a Call Option at $t = t_0$:

$C(S,t) = N(d_1)\, S - N(d_2)\, Ke^{-r(T-t)}$

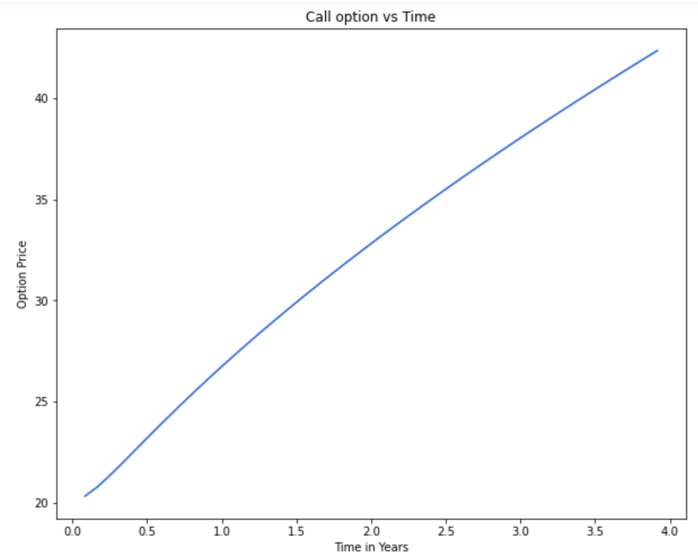
$\Rightarrow$ $C(S,t) = 0.9936*200 - 0.9928*180*e^{-0.02*(1/12)}$

$\therefore$ **$C(S,t) = 20.3136$ rupees.**

//Please refer to the code for Question 3 in the attached Jupyter Notebook.

The value that we are getting from the code using Black Scholes Formula is 20.317 rupees which is almost close to what we're getting from manual calculation.

Graph: Call Option Premium vs Time



From the curve, we can see that the premium for the given option of the underlying asset is increasing as time increases.

Question-4.

Goal: Given the equation of how the stock price of the underlying asset moves with time, find the Call Option Premium with the help of Monte Carlo Simulation.

Given equation of how the underlying asset moves with time:

S(t) = S(t-1)*exp((u-0.5*sigma2)*dt + sigma*N(0,1)*sqrt(dt))**

where dt = t – (t-1),
sigma = Implied Vol,
N(0,1) = Standard Normal variable.

We assume that the given option is a **European Call Option** which can be exercised only at maturity.

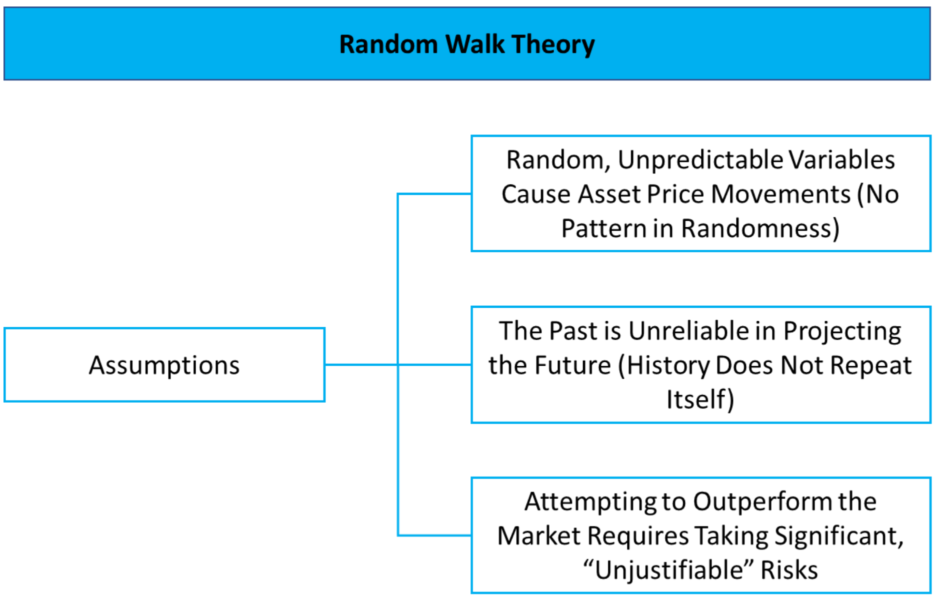
In the terms of pricing an option, the very famous Black-Scholes Model comes with a number of assumptions(as mentioned in Question 3) that don't always hold true in the real-world scenario.

The Monte Carlo methods are used to price options for which no closed-form solution exists and help to tackle multiple sources of uncertainties and random features such as changing interest rates, stock prices, etc.

In order to incorporate **"randomness"** in the context of simulating stock prices, Monte Carlo Simulation applies a model but in general, it is a model-independent method.

Now, the equation of the stock price of the underlying asset(as a function of time) given to us in the problem statement is exactly similar to that of the **Geometric Brownian Motion** which is an extension of the very famous **Random Walk Theory.**

One of the very popular theories for predicting stock prices is the Random Walk Theory.

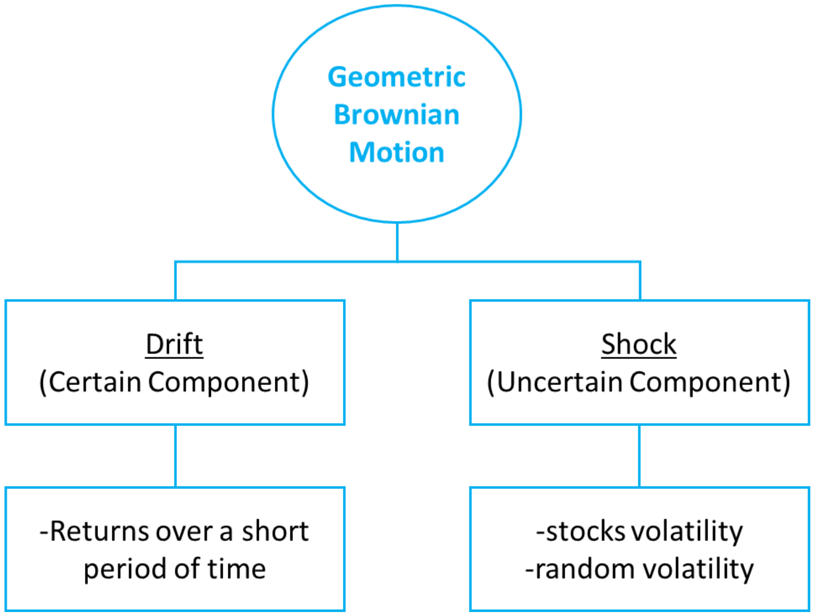


The **functional central limit theorem** extends the central limit theorem from random variables to random functions and goes on to state that one-dimensional Brownian motion is the limiting distribution of random walks defined from any distribution with zero mean and finite variance. Thus, a random walk is a discrete-space (integers) and discrete-time model, and Brownian Motion is a continuous-space and continuous-time model motivated by the same.

In a risk-neutral world the stock price follows a Generalised Wiener Process or the Geometric Brownian Motion i.e., it has a constant drift (r) and a variability in the path followed by stock (S), i.e., prices have a random walk, dz along this drift,

$$\frac{dS}{S} = rdt + \sigma dz$$

where dz over a small time period is equivalent to $\varepsilon\sqrt{dt}$.



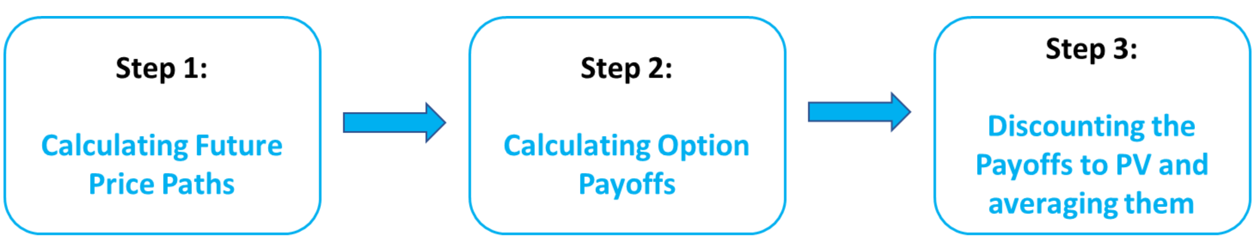
$$S_{t+\Delta t} = S_t \exp\left[\underbrace{\left(\mu - \frac{\sigma^2}{2}\right)}_{\text{Drift}}\Delta t + \underbrace{\sigma\varepsilon}_{\text{Shock}}\sqrt{\Delta t}\right]$$

ε here follows a standard normal distribution (with μ=0 and σ=1)

We can observe that though the volatility parameter is incorporated in Black-Scholes, the drift parameter is missing as it's derived based on the idea of arbitrage-free pricing.

Monte Carlo simulation uses the risk-neutral valuation result, i.e., price generation and discounting are carried out assuming **risk-neutrality**, which means that the asset price growth is governed by the risk-free rate as is the current value of the payoff. Thus, $\mu = r$.

The process of generating Option Premium using Monte Carlo is a 3 step process:



//Please refer to the code for Question 4 in the attached Jupyter Notebook.

The future price paths are calculated by simulating possible stock price paths. The time between t (0) to T($\frac{1}{12}$(year)) is discretized into M(given as 240) equally-spaced time steps. So, $\Delta t = \frac{T-t}{M} = \frac{\frac{1}{12}-0}{240}$ and $t_i = t + \Delta t * i$ for i = 0,1,2....240. We then simulate the stock price under risk-neutral probability (μ = r) using the given equation of the underlying asset. Let's consider that we simulate N paths and each path yields a stock price $S_{T,j}$, where j = 1,2,3...N, at T(Maturity).

Thus, the estimated premium of the European Call Option (C) from the Monte Carlo Method is the discounted or the Present Value of the expected payoff, which is given as

$$C = e^{-r(T-t)} \frac{\sum_{j=1}^{N} max(S_{T,j} - K, 0)}{N}$$

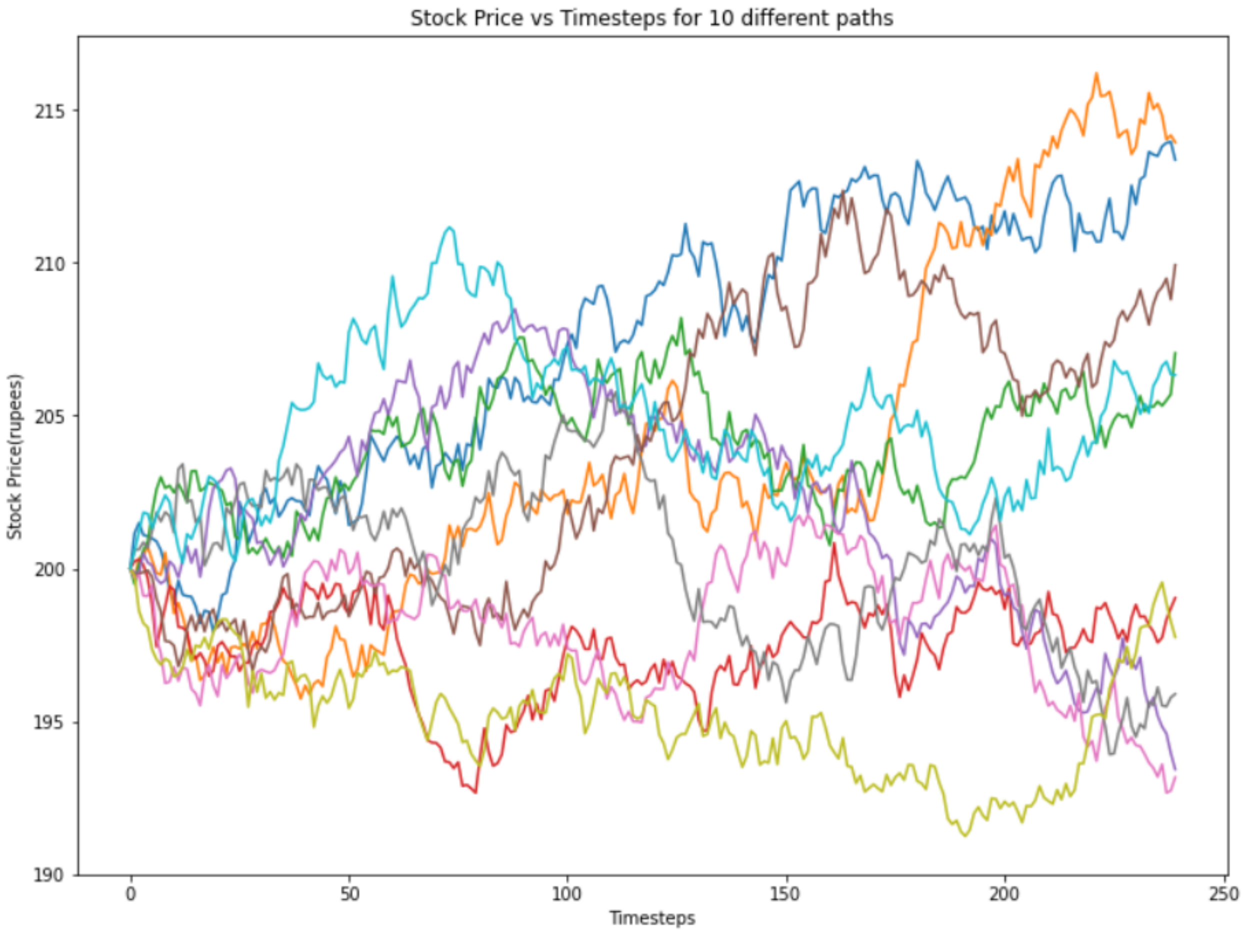
When S>K , the given Call Option is said to be in the money.

The value of the European Call option premium that we're getting from Monte Carlo simulation is 20.311 rupees.

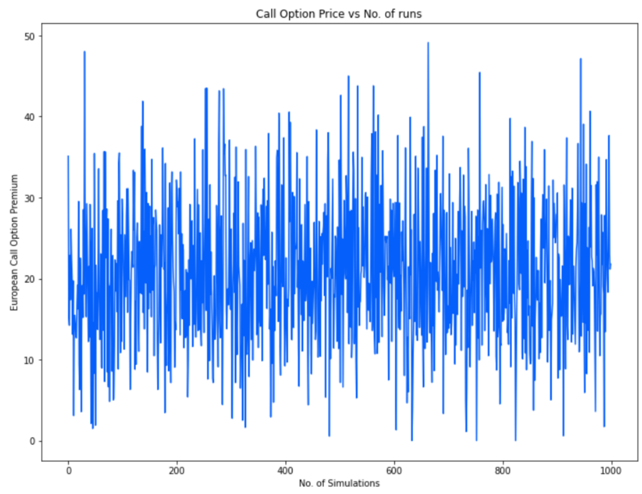
The error in the Monte Carlo Method in comparison to the Black Scholes solution:

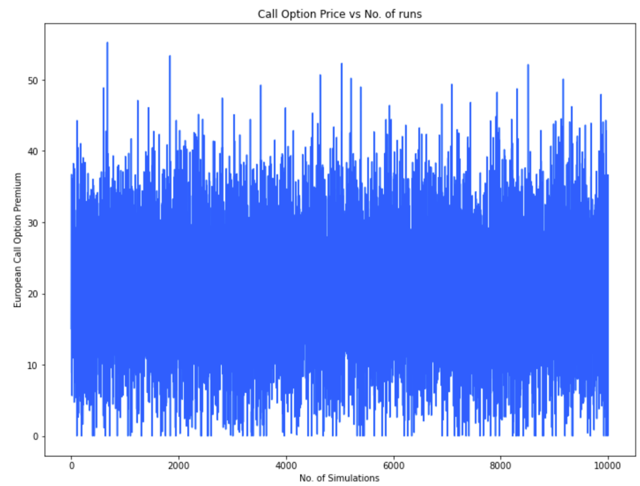
Error = $\frac{abs(20.311-20.3136)}{20.3136} * 100 = 0.01\%$

As, N -> ∞, the price approaches the Black-Scholes price, due to Central Limit Theorem.



Graph: Variation in the stock price of the underlying asset with timesteps





(a) (b)

(a): Call Option Price vs No. of runs for 1000 simulations
(b): Call Option Price vs No. of runs for 10000 simulations

Trade-off in Monte Carlo

The theory of law of large numbers requires a simulation to be repeated many times in order to have an accurate estimate.

While performing Monte Carlo Simulations, the more paths that are generated, the longer the simulation takes to run, and hence the longer the time it takes to price the option. Thus there's often a trade-off between the computational time and the precision of the price regarding number of number of paths in simulation.

Optimal number of runs for Monte Carlo Simulation

The number of paths used for simulation play an impactful role in the accuracy of the resultant prices.

- Too many no. of paths would make the calculation computationally expensive
- Too few paths would make the results heavily influenced by the unique random numbers.

Thus, variance reduction techniques have been developed in an attempt to minimize the number of simulations required to generate an accurate option price.

Way Forward..

Central Limit Theorem could be applied to estimate the optimal number of trials required for a desired confidence interval in the context of a Monte Carlo simulation.

To calculate number of runs required such that estimated value in Monte Carlo converges we have a formula:

$$m = \left(\frac{z_{\alpha/2} \times \sigma}{Er(\mu)} \right)^2$$

where $\frac{Z\alpha}{2}$ is the standard normal statistic for a two-sided C confidence, where $\alpha = (1 - C)$, with C being the statistical sampling confidence, commonly set to 95%, and $Er(\mu)$ is the standard error of the mean and is related to the amount of variation of the estimated mean.

Acknowledgement

The contribution of both the team members was extremely vital and this case study would not have been possible without the individual efforts. We would like to express our sincere gratitude to all the mentors as well as the organising team of JP Morgan Quant Mentorship Program for their valuable efforts and guidance throughout the course of this mentorship which helped us grasp the concepts efficiently and for giving us an opportunity to implement our learning on practical applications.