Monte Carlo simulation:

* A mathematical model that generates random variables for modeling uncertainty like risk assessment
* The inputs are modeled after probability distribution
* Monte Carlo Simulation provides a probabilistic estimate of the uncertainty in a model
* Stock prices follows a geometric Brownian motion which is a Markov process
* which means a certain state follows a random walk and its future value is dependent on the current value.

A basic example:

With that, let's consider a basic example. Here, we will consider a gambling scenario, where a user can "roll" the metaphorical dice for an outcome of 1 to 100. If the user rolls anything from 1-50, the "house" wins. If the user rolls anything from 51 to 99, the "user" wins. If the user rolls a 100, the house wins.

**Gambler`s fallacy:**

It is the mistake that if something that happens more frequently than normal in a given period of time then there`s a chance that it`ll happen more or less frequently in future.

Consider a coin flipping [example](/home/swayam/Desktop/desk/gambler's_fallacy.pdf)

[Code for example](/home/swayam/python/monte_carlo_sim/dice_ex.py)

The [output graph](bettor_result.svg) as you can see

It does n`t matter how many time you bet the odds are same.

**Martingale strategy:**

It relies on theory of mean-reversion i.e the profit or return no matter how much increased or fallen always moves back to average or mean of all its historical value over a period of time.

The system's mechanics involve an initial bet; however, each time the bet becomes a loser, the wager is doubled such that, given enough time, one winning trade will make up all of the previous losses.

To understand the basics behind the martingale strategy, let's look at an example. Suppose we had a coin and engaged in a betting game of either heads or tails with a starting wager of $1. There is an equal probability that the coin will land on heads or tails, and each flip is independent, meaning that the previous flip does not impact the outcome of the next flip. As long as you stick with the same directional view each time, you would eventually, given an infinite amount of money, see the coin land on heads and regain all of your losses, plus $1. The strategy is based on the premise that only one [trade](https://www.investopedia.com/terms/t/trade.asp) is needed to turn your account around.

Assume that you have $10 to wager, starting with a first wager of $1. You bet on heads, the coin flips that way and you win $1, bringing your [equity](https://www.investopedia.com/terms/e/equity.asp) up to $11. Each time you are successful, you continue to bet the same $1 until you lose. The next flip is a loser, and you bring your account equity back to $10. On the next bet, you wager $2 hoping that if the coin lands on heads, you will recoup your previous losses and bring your [net profit](https://www.investopedia.com/ask/answers/122414/net-income-same-profit.asp) and loss to zero. Unfortunately, it lands on tails again and you lose another $2, bringing your total equity down to $8. So, according to martingale strategy, on the next bet you wager double the prior amount to $4. Thankfully, you hit a winner and gain $4, bringing your total equity back up to $12. As you can see, all you needed was one winner to get back all of your previous losses.

However Once again, you have $10 to wager, with a starting bet of $1. In this scenario, you immediately lose on the first bet and bring your balance down to $9. You double your bet on the next wager, lose again and end up with $7. On the third bet, your wager is up to $4 and your losing streak continues, bringing you down to $3. You do not have enough money to double down, and the best you can do is bet it all. If you lose, you are down to zero and even if you win, you are still far from your initial $10 starting capital.

This is good when you have almost infinite fund but you don`t so it does n`t works.

Anyway the [code](martingale_startegy.py) for this