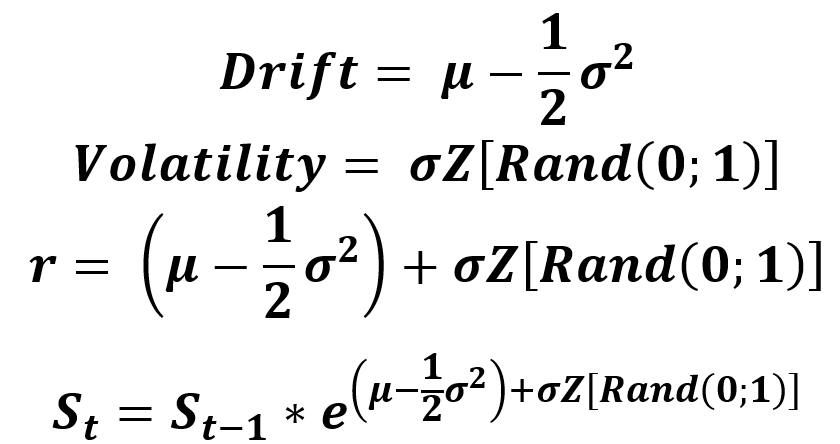
*How would you model a Stock Price as a Random Walk?*

Stock prices can be modeled as random walks using Monte Carlo Simulations. A Monte Carlo simulation can run an enormous number of trials with different random numbers generated from an underlying distribution for the uncertain variables. To understand how it is done, we need to understand Brownian Motion.

Brownian motion is a stochastic process used for modeling random behavior over time. Brownian motion has two main components:

Drift — the direction that rates of returns have had in the past. That is, the expected return of the stock. You may ask yourself: why is the variance multiplied by 0.5? Because historical values are eroded in the future.

Volatility — the historical volatility multiplied by a random, standard normal variable.



The procedure to predict stock price is:

Import the stock price data

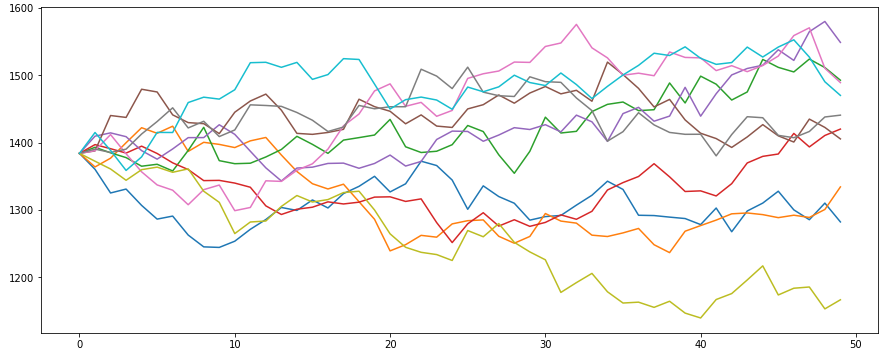
Compute the log returns

Compute the drift

Compute the variance and daily returns

Simulating the stock price for every trial.

We may get something like below for 10 trials (the paths will change every time we run the code as a different set of random numbers are generated):



Here is the python code to generate this

*https://medium.com/analytics-vidhya/monte-carlo-simulations-for-predicting-stock-prices-python-a64f53585662*