# Terminology

1. **Artificial Intelligence** – Algorithm/code to enable machines to mimic, devlop and demonstrate human cognitive behaviour.
2. **Machine Learning** – Techniques and processes to help machines acquire AI, Computer learns/improves by repeating prcesses over and over, tweeking at each repetition unitl the disiredd result.
   1. Supervised Learning: help from a data scientist
   2. Unsupervised Learning: by recognising patterns in input data
3. **Deep Learning**: Drawing meaningful inferences from large volumes of dtat. Requires artificial neuro-networks.

**Traditional computing**: inputs 🡪the programmer tells the computer, the process (rules & conditions) 🡪desired output

**ML** – the programmer tells the computer desired result🡪 and leave the computer to figure out the process( rules & Conditions)

# DEFINITIONS

* **Finance = Banking + Insurance + Asset Management**
* Contributes 20% of all global business revenue
* **Quants**: traders involved in quantitative analysis and related trading activities
* **A Portfolio** is a collection of investments owned by an individual
* **A Fund** is a pool of investments managed by a professional fund manager. Individuals buy ‘units’ of the fund and the manager invests the money
* **An Index:** is a sample og the market that is repr……
* **Bid:** (buy) = $99
* **Ask:** (sell) = $101
* **Spread**: = $2
* Difference between the money market & stock market is **liquidity**
* **Risk** is to holding an asset for an uncertain period of time. During this time the holder is exposed to the risk that the price may move against him.
* Stochastic: having a random probability distribution or pattern that may be analysed statistically but may not be predicted precisely.
* **Monte Carlo** method. ... **Monte Carlo** methods, or **Monte Carlo** experiments, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems that might be deterministic in principle
* **Monte Carlo Simulation** is a mathematical technique that generates random variables for modelling risk or uncertainty of a certain system. The random variables or inputs are modelled on the basis of probability distributions such as normal, log normal,

**GROSTTEN – MIGROM MODELS**

**GROSSMAN & MILLER (1988) MODEL**

## algorithmic trading

Advantages over discretionary trading:

1. Mechanical vs emotional
2. Time constraints
3. Back-testing
4. All contribute to consistency

Algorithmic trading Development

1. Idea
2. Code
3. Back-Testing
4. Optimise
5. Safeguards eg stop losses/take profits
6. Optimise
7. Paper trades(demo)
8. Launch – start small
9. Scale Up

Monitor & optimise

# financial time series

1. Stationarity & Linerity

# CLASSICAL financial models

1. Black-Scholes-Merton Option Formula
2. Monte Carlo Simulations

## ASSUMPTIONS:

1. Stationarity & Linearity
2. Martingale & Markov Properties: whatever happened yesterday has no impact or influence on today. Today, apocalypse; tomorrow, sunshine.
3. Independency
4. Gaussian Distribution returns are normally distributed and do not follow trends
5. Market Efficiency

The **Markov** property states that a stochastic process essentially has "no memory". ... The **Martingale** property states that the future expectation of a stochastic process is equal to the current value, given all known information about the prior events.

## MONTE CARLO SIMULATIONS - CALCULATION OF THE PRESENT VALUE OF STOCK

### Dividend Discount Model:

Where *V* is the present value of a stock,*g* the growth rate over time *n*

### GORDON GROWTH MODEL:

* In all cases r > g, significantly
* distribution for valuing mature companies with a constant dividend eg Coca Cola

**Compare current Market Price of the stock with its calculated Present Value to see if the stock is under or over-valued!**

#### Derivation of the Gordon Model

Asset growth:

*Let g* represent the rate of growth:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Year | 0 | 1 | 2 | 3 | n |
| Value (v) | *x* | *x(1+g)* | *x(1+g)2* | *x(1+g)3* | *x(1+g)n* |

Present Value of Stock is :

Where g is the rate of growth discounted at the rate of r

And r > g

Using the Geometric Progression:

Sn = A + AR + AR2 + AR3 + …+ ARn-1

#### Uniform Distribution

import random; import numpy as np; import seaborn as sns;

from matplotlib import pyplot as plt

d0 = 100

data = [ ]

for i in range(10000):

g = random.uniform(0.05, 0.08) # uniform distribution

r = random.uniform(0.09, 0.10)

v = d0 \* (1 + g)/(r - g) # Gordon Growth Model

data.append(v)

print ('Mean Price', np.mean(data)); print ('Median Price', np.median(data))

print ('STD', np.std(data)); print ('Min Price', np.min(data))

print ('Max Price', np.max(data))

sns.distplot(data); plt.ylabel("Frequency"); plt.xlabel("Market Value") plt.show()

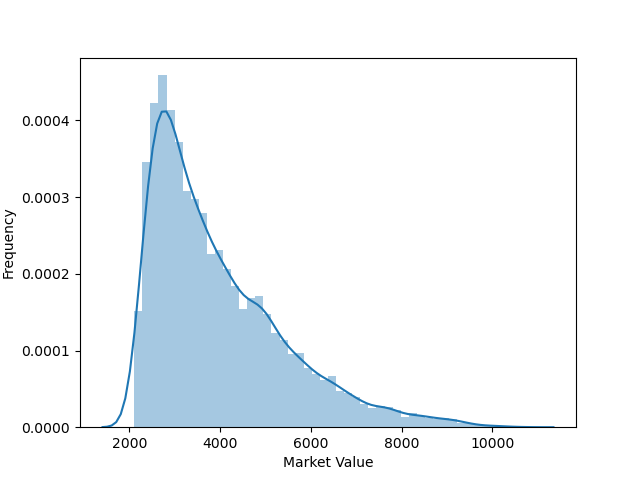
Mean Price 3967.41

Median Price 3552.58

STD 1468.95

Min Price 2106.91

Max Price 10651.02



#### Normal Distribution

Take the median price. 50% of values are above this price and 50% below.

68% of values are within ±1.0 σ

95% of values are within ±2.0 σ

Add /replace the code below:

d0 = 100 # initial price ($100)

g\_mean = 0.05 # average growth rate from the company history is 5%

g\_sd = 0.01 # standard deviation from the companies history is 1%

r = 0.1 # rate of return

g = norm.ppf(random.uniform(0, 1), loc=g\_mean, scale=g\_sd) # normal distribution

v = d0 \* (1 + g)/(r - g) # Gordon Growth Model

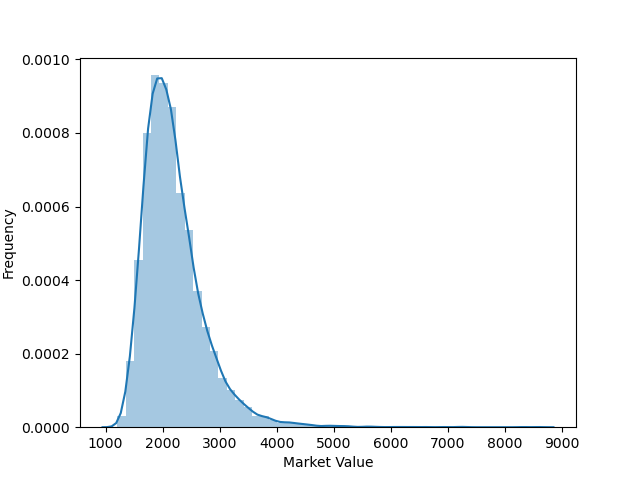
Mean Price 3967.41

Median Price 3552.58

STD 1468.95

Min Price 2106.91

Max Price 10651.02



### H – MODEL WITH NORMAL DISTRIBUTION

* The H-model is an approximation of the Dividend Discount Model
* Assumes 2 growth rates that change over time.
* Start with a short-term growth rate which linearly decreases to a mature rate over time
* E.g. *Tesla* – initial high growth rate that slows (mature)

Where *gS* is the short term growth rate and *gL* I s the long term growth rate.

H is the half-life of time *t* years, of short term growth.

E.g. S&P 500

from scipy.stats import norm

d0 = 163 # historical dividend last year

gL\_mean = 0.025 # average GDP growth rate

gL\_sd = 0.00001 # standard deviation

gS\_mean = 0.0 # over next year Zero grow

gS\_sd = 0.00001 # standard deviation

r = 0.08 # historical rate of return

t = 1 # 1 year short-term growth rate

H = t/2

gL = norm.ppf(random.uniform(0, 1), loc=gL\_mean, scale=gL\_sd)

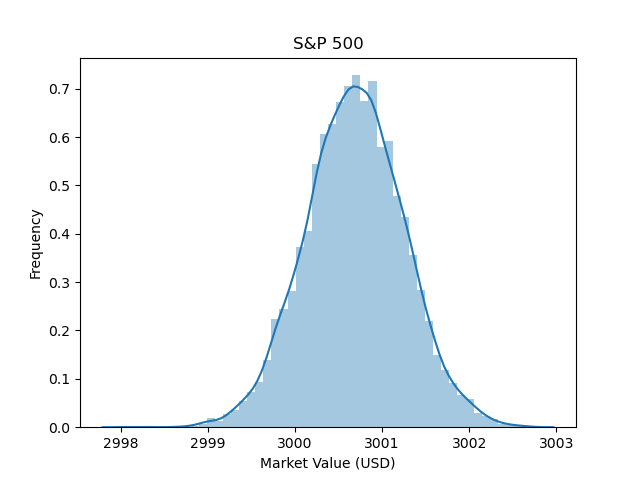
gS = norm.ppf(random.uniform(0, 1), loc=gS\_mean, scale=gS\_sd)

v = ((d0\*H\*(gS-gL)) + (d0\*(1+gL)))/(r-gL) # H - Growth Model

Mean Price 3000.68 Min Price 2998.45

Median Price 3000.69 Max Price 3002.87

STD 0.56



Compare with Today’s Annual price:

S&P 500 Index

INDEXSP: .INX

3,066.59 +25.28 (0.83%)

15 Jun, 17:18 GMT-4 ·

Source: Google

### 3 - STAGE MODEL WITH NORMAL DISTRIBUTION

pv = d0\*(1+gM)/(1+r) + ((d0\*(1+gM)\*H\*(gS-gL))/(r-gL) + (d0\*(1+gM)\*(1+gL))/(r-gL))/(1+r)

from scipy.stats import norm

d0 = 163 # historical dividend last year

gL\_mean = 0.025 # average GDP growth rate

gL\_sd = 0.00001 # standard deviation

gM\_mean = 0.005 # over next year -ve growth

gM\_sd = 0.00001 # standard deviation

gS\_mean = 0.025 # over next year 15% grow

gS\_sd = 0.00001 # standard deviation

r = 0.08 # historical rate of return

t = 1 # 1 year short-term growth rate

H = t/2

data = [] # List

for i in range(10000):

gL = norm.ppf(random.uniform(0, 1), loc=gL\_mean, scale=gL\_sd)

gM = norm.ppf(random.uniform(0, 1), loc=gM\_mean, scale=gM\_sd)

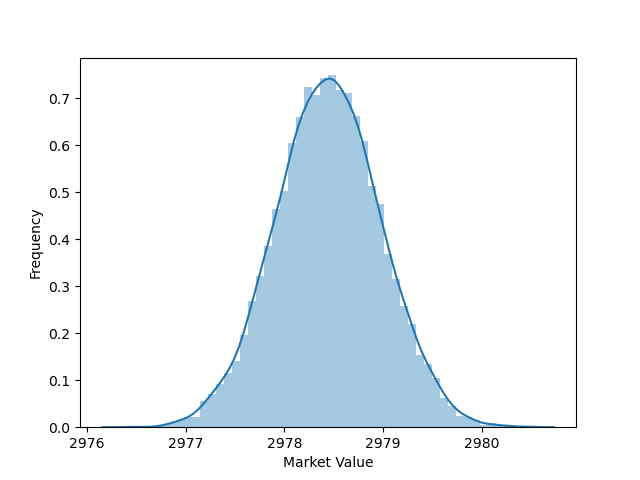
gS = norm.ppf(random.uniform(0, 1), loc=gS\_mean, scale=gS\_sd)

d1 = d0\*(1+gM)

v1 = d1/(1+r)

v2 = ((d1\*H\*(gS-gL))/(r-gL) + (d1\*(1+gL))/(r-gL))/(1+r)

v = v1 + v2



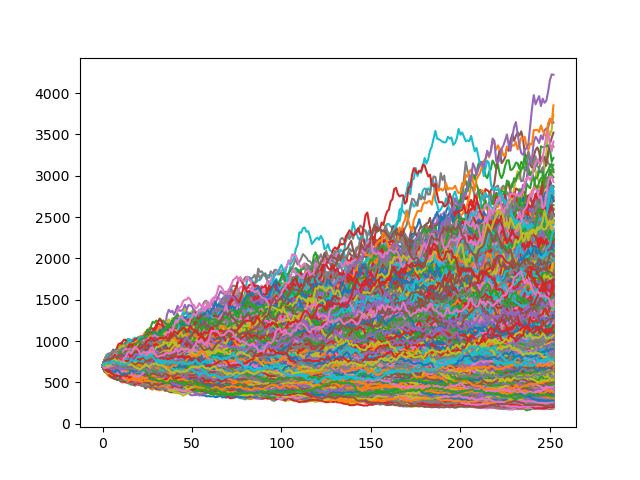
Mean Price 2978.46 Min Price 2976.39

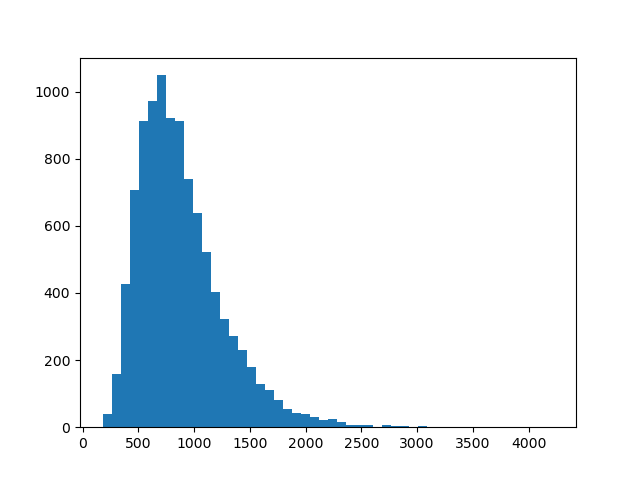
Median Price 2978.45 Max Price 2980.39

STD 0.52

## MONTE CARLO SIMULATIONS - PREDICTING FUTURE STOCK PRICES

Initial: Price $698.98. 10,000 iterations





Mean Price 881.61 Min Price 179.11

Median Price 809.18 Max Price 4222.47

STD 390.53

5% quantile = 404.4

95% quantile = 1607.83

* Problem: this is random calculation does not take into account previous behaviour of stock prices!

# NEW RESEARCH IN finACIAl TIME SERIES ANALYTSIS

1. Non-Stationarity & Non-Linearity: yesterday does matter
2. Fractinality & Momentum: using wavelet transforms, data follows fractal patterns, follow trends and display momentum
3. No Independency
4. Trends follow Log-Normal Distributions leading to conclusion no 5.
5. Efficient Markets do not exist.

## Fourier anaysis

* Transformation of Raw Signal (Time Domain) into processed signal (frequency domain) to obtain frequency spectrum
* High frequencies equals rapid change
* Requires periodic signals, so inapplicable in financial market data analysis

## wavelet anaysis