Computational Finance

Random number

Linear congruential generators

Technique to **create random number** such that the $n + 1$ th number is defined recursively by the n th number. The method **helps to** generate from and form many **distribution tests**. where a, c, and m are integers. A "seed" value, $X0$, needs to be defined at the beginning in order to start generating the sequence.

```
In[ ]: def lin_con_gen(n, a, c, m, x_0):
           """generates n numbers from the linear congruential generator with parameters a,c,m
           and seed x_0"""
           random_list = [None]*n
           random_list[0] = (a*x_0+c)%m
           for i in range(1,n):
               random_list[i] = (a*random_list[i-1]+c)%m
           return random_list
```

Generating uniform random numbers with SciPy

```
import numpy as np
from scipy.stats import uniform

#generating uniform random variables
unif_array = uniform.rvs(size = 1000)
```

sampling from more complex distributions such as exponential

Inverse transform method (reversal)

```
#generating uniform random variables
unif_array = uniform.rvs (size = 10000)

#generating exponential random variables
lambda_val = 1
expo_array = -math.log(1-unif_array)/lambda_val
```

Generate normal reversal random number

```
from scipy.stats import norm

#generating standard normal variables
norm_array = norm.rvs(size = 1000)
```

In Black-Scholes model the distribution is **log-normally**.
Hence need **multivariate normal distribution (MVN)**

```
Sigma = np.array([[1,0.5],[0.5,1]])
L = np.linalg.cholesky(Sigma)
```

```
uncorr_norms = norm.rvs(size = 2)
corr_norms = np.matmul(L, uncorr_norms)
```

corr_norms contains the two correlated random normals. Correlated numbers like these can be used to simulate correlated stock prices in a Black-Scholes world by noting that, within this framework, stock prices are log-normally distributed.

Computational Finance

Monte Carlo simulation

Method for **estimating** the value of **integrals is Monte Carlo simulation**.
Monte Carlo estimation is a technique which **relies on random sampling**
Example: Find integral of

$$\int_{-\infty}^{\infty} e^{-x^2}\, dx.$$

```
np.random.seed(0)

mexp_est = [None]*50
mexp_std = [None]*50

for i in range (1,51):
    norm_array = norm.rvs(size = i*1000)
    exp_value = np.exp(-norm_array**2/2)*np.sqrt(2*np.pi)
    mexp_est[i-1] = np.mean(exp_value)
    mexp_std[i-1] = np.std(exp_value)/np.sqrt(i*1000)

plt.plot ([np.sqrt(np.pi)]*50)
plt.plot (mexp_est, '.')
plt.plot (np.sqrt(np.pi)+np.array (mexp_std)*3, 'r')
plt.plot (np.sqrt(np.pi)-np.array(mexp_std)*3, 'r')
plt.xlabel ("Sample Size")
plt.ylabel ("Value")
plt.show ()

```

The Black-Scholes market model

The Black-Scholes market model provides a stochastic (integral method performed on random simulations) differential equation which models the changes in a given stock's price over time.

It performs sorf operations using monte carlo simulation for pricing options.
It has a mathematical formula which considers all options parameters, enabling it to provide correct option pricing.
Stock price simulation in Python through the use of an example: pricing a vanilla European call option. We will need the following information: More details on WQU 2nd chapter option pricing.

- Continuously compounded, risk-free interest rate, r
- Stock volatility, σ
- Initial stock price, S_0
- Option strike price, K
- Option maturity, T

$$\boxed{f(S_T) = e^{-rT}(S_T - K)^+.}$$

Value at Risk : Keep on calculating VAR is important if we want to hold stocks longer.
(I/P: Consider share price and volatility for VAR operations)
Monte Carlo VaR models make use of Monte Carlo methods to project asset values, in order to get some idea of what happens to your portfolio value in the future.

Computational Finance

Historical VaR models: Estimate VaR based on historical simulation.
We can work out two days' worth of returns for that share. If we then sample from these two returns, we can then project the value of our share one day into the future. In doing so, we are assuming that the company will be subject to similar conditions as it was in the past, so the past returns will be reflective of future returns.

Credit Valuation Adjustment and Merton model helps to identify the risk of churn/default. It also uses simulation methods and few assumptions to identify rate of churn.

Advanced technique for option pricing.
Fourier analysis for practical value in forecasting stock market price. Because Fourier analysis seeks to break down repetitive waveforms into harmonic components and the stock market doesn't move in a well-defined and repetitive manner

Fourier transform: It is a manipulation of an integrable function

Calibration
Mostly model's parameters as being given but if we need to find a set of parameter values based on some dataset is known as **calibration**.