In [1]:
```
import numpy as np
import pandas as pd
from pandas_datareader import data as wb
from scipy.stats import norm
%matplotlib inline
import yfinance as yf
```

Black Scholes Formula

S = Stocks current price

K = Strike price at which the option can be exercised

T = The option's time tile expiration

r = Risk-free rate

s = Standard Deviation

N = Normal Distribution

e = exponential term

C = Call prenium

$$Formula$$

$$d_1 = \frac{\ln(\frac{S}{K}) + (r + \frac{stdev^2}{2})t}{s \cdot \sqrt{t}}$$

$$d_2 = d_1 - s \cdot \sqrt{t} = \frac{\ln(\frac{S}{K}) + (r - \frac{stdev^2}{2})t}{s \cdot \sqrt{t}}$$

In [2]:
```
def d1(S, K, r, s, T):
    return (np.log(S / K) + (r + s ** 2/2)*T) / (s * np.sqrt(T))

def d2(S, K, r, s, T):
    return (np.log(S / K) + (r - s ** 2/2)*T) / (s * np.sqrt(T))
```

In [3]:
```
norm.cdf(0)
```

Out[3]: 0.5

In [4]:
```
norm.cdf(0.25)
```

Out[4]: 0.5987063256829237

In [5]:
```
norm.cdf(0.75)
```

Out[5]: 0.7733726476231317

In [6]:
```
norm.cdf(9)
```

Out[6]:
```
1.0
```

$$\mathbf{C} = SN(d_1) - Ke^{-rt}N(d_2)$$

In [7]:
```
def BSM(S, K, r, s, T):
    return (S * norm.cdf(d1(S,K, r, s, T))) - (K * np.exp(-r * T) * norm.cdf(d2
(S, K, r, s, T)))
```

In [8]:
```
#call data of a stock

ticker = 'MSFT'
data = pd.DataFrame()
data[ticker] = wb.DataReader(ticker, data_source='yahoo', start='2016-1-1', end='
2019-7-29')['Adj Close']
```

In [9]:
```
S = data.iloc[-1]
S
```

Out[9]:
```
MSFT    141.029999
Name: 2019-07-29 00:00:00, dtype: float64
```

In [10]:
```
log_returns = np.log(1 + data.pct_change())
```

In [11]:
```
s = log_returns.std() * 250 ** 0.5
s
```

Out[11]:
```
MSFT    0.221814
dtype: float64
```

In [24]:
```
# Change K to the strike price you want

r = 0.025
K = 160.0
T = 1
```

In [25]:
```
d1(S, K, r, s, T)
```

Out[25]:
```
MSFT   -0.345337
dtype: float64
```

In [26]:
```
d2(S, K, r, s, T)
```

Out[26]:
```
MSFT   -0.567151
dtype: float64
```

In [27]:
```
BSM(S, K, r, s, T)
```

Out[27]:
```
MSFT    6.94288
Name: 2019-07-29 00:00:00, dtype: float64
```

In [ ]:
```

```