Assignment 5

Group 5 : Sanskruti Mittal, Kaushik Gurav, Xianglin Cheng, Minghao Wang

Import Modules

In [1]:
```
import pandas              as pd
import numpy               as np
import scipy               as scp
import matplotlib.pyplot as plt
import os                  as os
from datetime import date as dd  # for dates
from scipy import optimize
from scipy import stats
import statsmodels.api as sm
```

Datafiles

Part 1

Q1.1

In [2]:
```
r = 0.02                          # risk free rate
d = 0.9524                        # return in down state
u = 1.05                          # return in up state
p = (1+r-d)/(u-d)                 # risk neutral probability of up state
print(f'The probability is {p :.2f}.')
```

```
The probability is 0.69.
```

Because we consider that (u-1)*p* + *(d-1)*(1-p) = rf.

Q 1.2

In [3]:
```
S0 = 100                                            # Fill in stock prices at all nodes
Su = 105
Sd = 95.24
Suu = 110.25
Sud = 100
Sdd = 90.70
K = 95                                             # Strike price
Cuu = max(Suu - K, 0)                              # Calculate call payoff at all nodes
Cud = max(Sud - K, 0)
Cdd = max(Sdd - K, 0)
Cu = (p*Cuu + (1-p)*Cud)/(1+r)
Cd = (p*Cud + (1-p)*Cdd)/(1+r)

Cu1 = Su - K
Cd1 = Sd - K
print(Cu1 , Cd1)

C = (p*Cu + (1-p)*Cd)/(1+r)
print(f'The option price at node u is {Cu :.2f}.')
print(f'The option price at node d is {Cd :.2f}.')
print(f'The option price at node 0 is {C :.2f}.')
```

```
10 0.23999999999999488
The option price at node u is 11.86.
The option price at node d is 3.40.
The option price at node 0 is 9.08.
```

Q 1.3

In [4]:
```
H0 = (Cu - Cd)/(u*S0-d*S0)                          # Fill in the hedge ratios
Hu = (Cuu - Cud)/(u*u*S0-u*d*S0)
Hd = (Cud - Cdd)/(d*u*S0-d*d*S0)
print(f'The hedge ratio at node 0 is {H0 :.2f}.')
print(f'The hedge ratio at node u is {Hu :.2f}.')
print(f'The hedge ratio at node d is {Hd :.2f}.')
```

```
The hedge ratio at node 0 is 0.87.
The hedge ratio at node u is 1.00.
The hedge ratio at node d is 0.54.
```

With the stock price increasing，call option becomes deep in the money so the slope of the value of call option becomes the largest(equal to 1) so hedge ratio increases when stock price increases.

Q 1.4 (Put)

In [5]:
```
S0 = 100                                                   # Fill in stock prices at all nodes
Su = 105
Sd = 95.24
Suu = 110.25
Sud = 100
Sdd = 90.70
Kp = 105                                                    # Strike price

Puu = max(Kp - Suu, 0)                                      # Calculate call payoff at all nodes
Pud = max(Kp - Sud, 0)
Pdd = max(Kp - Sdd, 0)

Pu = (p*Puu + (1-p)*Pud)/(1+r)
Pd = (p*Pud + (1-p)*Pdd)/(1+r)

Pu1 = max(Kp - Su,0)
Pd1 = max(Kp - Sd,0)
#print(Pu1 , Pd1)

P = (p*Pu + (1-p)*max(Pd,Pd1))/(1+r)
print(f'The option price at node u is {Pu :.2f}.')
print(f'The option price at node d is {Pd :.2f}.')
print(f'The option price at node 0 is {P :.2f}.')
```

```
The option price at node u is 1.51.
The option price at node d is 7.70.
The option price at node 0 is 3.96.
```

In [6]:
```
H0 = (Pu - Pd1)/(u*S0-d*S0)                                   # Fill in the hedge ratios
Hu = (Puu - Pud)/(u*u*S0-u*d*S0)
Hd = (Pud - Pdd)/(d*u*S0-d*d*S0)
print(f'The hedge ratio at node 0 is {H0 :.2f}.')
print(f'The hedge ratio at node u is {Hu :.2f}.')
print(f'The hedge ratio at node d is {Hd :.2f}.')
```

```
The hedge ratio at node 0 is -0.85.
The hedge ratio at node u is -0.49.
The hedge ratio at node d is -1.00.
```

Part 2

Q 2.1

In [7]:
```
options_data = pd.read_excel('options_data_2022.xlsx', usecols = "A:E", header = 0)
options_data["ret"] = np.log(options_data['sp_500_index']) - np.log(options_data['sp_500_index'].shift(1))
T = len(options_data)
options_data
```

Out[7]:

	Date	vix	sp_500_index	call_strike=4000	put_strike=4000	ret
0	2020-11-02	37.130000	3310.239990	NaN	NaN	NaN
1	2020-11-03	35.550000	3369.159912	NaN	NaN	0.017643
2	2020-11-04	29.570000	3443.439941	NaN	NaN	0.021808
3	2020-11-05	27.580000	3510.449951	NaN	NaN	0.019273
4	2020-11-06	24.860000	3509.439941	NaN	NaN	-0.000288
...	...	...	...	...	...	...
497	2022-10-24	29.850000	3797.340000	NaN	NaN	0.011812
498	2022-10-25	28.459999	3859.110000	NaN	NaN	0.016136
499	2022-10-26	27.280001	3830.600000	NaN	NaN	-0.007415
500	2022-10-27	27.389999	3807.300000	NaN	NaN	-0.006101
501	2022-10-28	25.750000	3901.060000	129.0	185.32	0.024328

502 rows × 6 columns

In [8]:
```
def BScholes(S,K,maturity, r, sigma, delta):
    d1 = (np.log(S*np.exp(-delta*maturity)/K) + (r + sigma**2/2)*maturity)/(sigma*np.sqrt(maturity))
    d2 = d1 - sigma*np.sqrt(maturity)
    Nd1 = stats.norm.cdf(d1)
    Nd2 = stats.norm.cdf(d2)
    Nd1neg = stats.norm.cdf(-d1)
    Nd2neg = stats.norm.cdf(-d2)

    callprice = S*np.exp(-delta*maturity)*Nd1 - np.exp(-r*maturity)*K*Nd2
```

```
    putprice = np.exp(-r*maturity)*K*Nd2neg - S*np.exp(-delta*maturity)*Nd1neg
    return callprice, putprice
```

In [9]:
```
sigma = options_data["ret"].iloc[-100:].std()*np.sqrt(252)    #calculate the annualized vol using the ret from 
S = 3901                                              # Fill these in
K = 4000
maturity = 50/252
r = 0.04
delta = 0.016
callprice = BScholes(S,K,maturity, r, sigma, delta)[0]
putprice = BScholes(S,K,maturity, r, sigma, delta)[1]
print(f'The call price is {callprice :.2f}.')
print(f'The put price is {putprice :.2f}.')
```

```
The call price is 136.98.
The put price is 216.73.
```

Q 2.2

In [10]:
```
sigma200 = options_data["ret"].iloc[-200:].std()*np.sqrt(252)         #calculate the annualized vol using 200 d
callprice200 = BScholes(S,K,maturity, r, sigma200, delta)[0]
putprice200 = BScholes(S,K,maturity, r, sigma200, delta)[1]

print(f'The call price with 200 days vol is {callprice200 :.2f}.')
print(f'The put price with 200 days vol is {putprice200 :.2f}.')
```

```
The call price with 200 days vol is 136.09.
The put price with 200 days vol is 215.83.
```

With the sigma increasing, the price of opitons will increase.

Q 2.3

The value of the Vix on 10/28/2022 was 25.75%.

In [11]:
```
print(f'The 200 days vol is {sigma200*100 :.2f} percent.')
print(f'The 100 days vol is {sigma*100 :.2f} percent.')
```

```
The 200 days vol is 24.80 percent.
The 100 days vol is 24.93 percent.
```

The VIX implies higher volatility than my historical volatility estimates.

Q 2.4

In [12]:
```
call = 129
put = 185
def callfunc(x):                                     # callfunc takes input the volatility x and ouptputs the 
    return BScholes(S,K,maturity, r, x, delta)[0]

def putfunc(x):
    return BScholes(S,K,maturity, r, x, delta)[1]     # Fill in the output for the putfunc

impliedvol_call = (scp.optimize.minimize(lambda x: (callfunc(x) - call)**2, x0 = 0.2).x)[0]
impliedvol_put = (scp.optimize.minimize(lambda x: (putfunc(x) - put)**2, x0 = 0.2).x)[0]             # Perform

print(f'The implied vol with call is {impliedvol_call*100 :.2f} percent.')
print(f'The implied vol with put is {impliedvol_put*100 :.2f} percent.')
```

```
The implied vol with call is 23.77 percent.
The implied vol with put is 20.29 percent.
```

The implied vol with call is higher than that with put because investors expect the stock price will increase and the call price is likely to be higher than its intrinsic value, which means that there is a higher implied volatility.