Black_Scholes_Stock_Puts

September 29, 2021

1 Black Scholes Stock Puts Inputs

[1]:
```
import numpy as np
import scipy.stats as ss
import matplotlib.pyplot as plt
import yfinance as yf
```

[2]:
```
dfo = yf.Ticker("AAPL")
```

[3]:
```
dfo.options
```

[3]:
```
('2020-10-15',
 '2020-04-23',
 '2021-09-16',
 '2020-04-30',
 '2021-01-14',
 '2020-06-18',
 '2021-06-17',
 '2020-12-17',
 '2020-05-21',
 '2022-01-20',
 '2020-09-17',
 '2020-04-16',
 '2022-06-16',
 '2020-05-14',
 '2020-05-07',
 '2020-05-28',
 '2020-07-16')
```

[4]:
```
dfo_exp = dfo.option_chain('2020-05-28')
```

[5]:
```
dfo_exp.puts
```

[5]:

	contractSymbol	lastTradeDate	strike	lastPrice	bid	ask	\
0	AAPL200529P00195000	2020-04-09 18:05:46	195.0	2.15	1.87	2.09	
1	AAPL200529P00200000	2020-04-09 18:38:33	200.0	2.37	1.99	2.67	
2	AAPL200529P00205000	2020-04-09 19:44:51	205.0	2.69	2.29	3.05	

```
3   AAPL200529P00220000 2020-04-09 19:57:24   220.0       4.15   3.65   4.15
4   AAPL200529P00225000 2020-04-09 17:52:28   225.0       4.85   4.30   5.05
5   AAPL200529P00235000 2020-04-09 19:46:45   235.0       6.32   5.90   6.40
6   AAPL200529P00240000 2020-04-09 19:59:54   240.0       7.21   6.85   7.85
7   AAPL200529P00245000 2020-04-09 19:20:38   245.0       8.80   8.00   8.55
8   AAPL200529P00250000 2020-04-09 19:59:54   250.0       9.68   9.25  10.30
9   AAPL200529P00260000 2020-04-09 17:03:54   260.0      13.52  12.55  13.55
10  AAPL200529P00262500 2020-04-09 16:48:56   262.5      14.85  13.40  14.05
11  AAPL200529P00265000 2020-04-09 19:55:35   265.0      14.90  14.35  15.40
12  AAPL200529P00267500 2020-04-09 19:56:34   267.5      15.75  15.75  16.50
13  AAPL200529P00270000 2020-04-09 14:02:21   270.0      16.50  16.35  17.60
14  AAPL200529P00275000 2020-04-09 19:46:45   275.0      19.50  17.70  20.65
15  AAPL200529P00280000 2020-04-09 13:44:29   280.0      21.26  20.55  23.45

    change percentChange  volume  openInterest  impliedVolatility  inTheMoney  \
0     2.15      Infinity       2           NaN           0.633060       False
1     2.37      Infinity       2           NaN           0.618534       False
2     2.69      Infinity      10           NaN           0.599980       False
3     4.15      Infinity       3           NaN           0.540410       False
4     4.85      Infinity       1           NaN           0.531377       False
5     6.32      Infinity      12           NaN           0.506414       False
6     7.21      Infinity      24           NaN           0.507146       False
7     8.80      Infinity       1           NaN           0.478582       False
8     9.68      Infinity       9           NaN           0.477849       False
9    13.52      Infinity       3           NaN           0.452276       False
10   14.85      Infinity      20           NaN           0.434454       False
11   14.90      Infinity       3           NaN           0.437628       False
12   15.75      Infinity      14           NaN           0.432745       False
13   16.50      Infinity       4           NaN           0.426397        True
14   19.50      Infinity      13           NaN           0.431158        True
15   21.26      Infinity       5           NaN           0.423468        True

   contractSize currency
0       REGULAR      USD
1       REGULAR      USD
2       REGULAR      USD
3       REGULAR      USD
4       REGULAR      USD
5       REGULAR      USD
6       REGULAR      USD
7       REGULAR      USD
8       REGULAR      USD
9       REGULAR      USD
10      REGULAR      USD
11      REGULAR      USD
12      REGULAR      USD
13      REGULAR      USD
```

```
14      REGULAR      USD
15      REGULAR      USD
```

[6]:
```
symbol = 'AAPL'
start = '2019-12-01'
end = '2020-04-02'
```

[7]:
```
df = yf.download(symbol,start,end)
```

```
[**********************100%**********************]  1 of 1 completed
```

[8]:
```
df.head()
```

[8]:
```
              Adj Close       Close        High         Low        Open  \
Date
2019-12-02  263.534546  264.160004  268.250000  263.450012  267.269989
2019-12-03  258.835724  259.450012  259.529999  256.290009  258.309998
2019-12-04  261.120270  261.739990  263.309998  260.679993  261.070007
2019-12-05  264.951172  265.579987  265.890015  262.730011  263.790009
2019-12-06  270.069031  270.709991  271.000000  267.299988  267.480011

               Volume
Date
2019-12-02  23621800
2019-12-03  28607600
2019-12-04  16795400
2019-12-05  18606100
2019-12-06  26518900
```

[9]:
```
df.tail()
```

[9]:
```
              Adj Close       Close        High         Low        Open  \
Date
2020-03-26  258.440002  258.440002  258.679993  246.360001  246.520004
2020-03-27  247.740005  247.740005  255.869995  247.050003  252.750000
2020-03-30  254.809998  254.809998  255.520004  249.399994  250.740005
2020-03-31  254.289993  254.289993  262.489990  252.000000  255.600006
2020-04-01  240.910004  240.910004  248.720001  239.130005  246.500000

               Volume
Date
2020-03-26  63021800
2020-03-27  51054200
2020-03-30  41994100
2020-03-31  49250500
2020-04-01  44054600
```

[10]:
```
returns = df['Adj Close'].pct_change().dropna()
```

[11]:
```
from datetime import datetime
from dateutil import relativedelta

d1 = datetime.strptime(start, "%Y-%m-%d")
d2 = datetime.strptime('2020-05-28', "%Y-%m-%d")
delta = relativedelta.relativedelta(d2,d1)
print('How many years of investing?')
print('%s years' % delta.years)
```

```
How many years of investing?
0 years
```

[12]:
```
maturity_days = (df.index[-1] - df.index[0]).days
print('%s days' % maturity_days)
```

```
121 days
```

[13]:
```
S0 = df['Adj Close'][-1]
K = dfo_exp.puts['strike'][6]
r = 0.1
sigma = returns.std()
T = maturity_days/252
```

[14]:
```
print("S0\tCurrent Stock Price:", S0)
print("K\tStrike Price:", K)
print("r\tContinuously compounded risk-free rate:", r)
print("sigma\tVolatility of the stock price per year:", sigma)
print("T\tTime to maturity in trading years:", T)
```

```
S0      Current Stock Price: 240.91000366210938
K       Strike Price: 240.0
r       Continuously compounded risk-free rate: 0.1
sigma   Volatility of the stock price per year: 0.0369388726875486
T       Time to maturity in trading years: 0.4801587301587302
```

[15]:
```
def d1(S0, K, r, sigma, T):
    d1 = (np.log(S0/K) + (r + sigma**2 / 2) * T)/(sigma * np.sqrt(T))
    return d1
```

[16]:
```
def d2(S0, K, r, sigma, T):
    d2 = (np.log(S0 / K) + (r - sigma**2 / 2) * T) / (sigma * np.sqrt(T))
    return d2
```

[17]:
```
def BlackScholesCall(S0, K, r, sigma, T):
```

```
    BSC = S0 * ss.norm.cdf(d1(S0, K, r, sigma, T)) - K * np.exp(-r * T) * ss.
↪norm.cdf(d2(S0, K, r, sigma, T))
    return BSC
```

[18]:
```
def BlackScholesPut(S0, K, r, sigma, T):
    BSP = K * np.exp(-r * T) * ss.norm.cdf(-d2(S0, K, r, sigma, T)) - S0 * ss.
↪norm.cdf(-d1(S0, K, r, sigma, T))
    return BSP
```

[19]:
```
Put_BS = BlackScholesPut(S0, K, r, sigma, T)
Put_BS
```

[19]:
```
0.04785864194304423
```