[33]:
```
import numpy as np
import pandas as pd
from pandas_datareader import data as pdr
from datetime import datetime, date, timedelta

from math import log, sqrt, pi, exp
from scipy.stats import norm
#from pandas import DataFrame

import yfinance as yfin
yfin.pdr_override()

pd.set_option('display.float_format', lambda x: f'{x:,.2f}')
```

Assignment - Black-Scholes Calculator Black-Scholes Calculator

- Build a Black Scholes Calulator in Python or R (n.b. Open Source Models are available for free download)
- Select an Asset in your portfolio and using the calculator that you created price a European option on that asset
- Present your work including the code that you have developed, the results of the pricing and attribution of resources that you have used including any open sourrce code.
- Post a summary on the Forum and comment on the work of your coleagues.

Variables

[43]:
```
stock = 'PEX'
expiry = '12-18-2024'
strike_price_call = 30
strike_price_put = 30
today = datetime.now()

one_year_ago = today.replace(year=today.year-1)
```

Functions

[35]:
```
def d1(S,K,T,r,sigma):
    return(log(S/K)+(r+sigma**2/2.)*T)/(sigma*sqrt(T))

def d2(S,K,T,r,sigma):
    return d1(S,K,T,r,sigma)-sigma*sqrt(T)

def bs_call(S,K,T,r,sigma):
    return S*norm.cdf(d1(S,K,T,r,sigma))-K*exp(-r*T)*norm.cdf(d2(S,K,T,r,sigma))

def bs_put(S,K,T,r,sigma):
    return K*exp(-r*T)-S*bs_call(S,K,T,r,sigma)
```

[ ]:

Load Data

[36]:
```
df = pdr.get_data_yahoo(stock, start=one_year_ago, end=today)
df
```

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

[36]:

Date	Open	High	Low	Close	Adj Close	Volume
2022-10-21	23.63	24.07	23.63	24.07	21.46	2200
2022-10-24	24.19	24.43	24.06	24.39	21.75	5400
2022-10-25	24.54	25.34	24.54	25.34	22.60	4100
2022-10-26	25.72	25.78	25.61	25.61	22.84	1400
2022-10-27	25.62	25.76	25.48	25.48	22.72	3600
...	...	...	...	...	...	...
2023-10-16	26.00	26.00	26.00	26.00	26.00	200
2023-10-17	25.83	25.92	25.83	25.92	25.92	100
2023-10-18	25.69	25.69	25.49	25.49	25.49	1400
2023-10-19	25.47	25.47	25.16	25.17	25.17	43900
2023-10-20	24.85	24.98	24.85	24.89	24.89	2600

251 rows × 6 columns

Calculate Black Scholes

[37]:
```
df = df.sort_values(by="Date")
df = df.dropna()
df['returns'] = df.Close.pct_change()
df
```

[37]:

Date	Open	High	Low	Close	Adj Close	Volume	returns
2022-10-21	23.63	24.07	23.63	24.07	21.46	2200	NaN
2022-10-24	24.19	24.43	24.06	24.39	21.75	5400	0.01
2022-10-25	24.54	25.34	24.54	25.34	22.60	4100	0.04
2022-10-26	25.72	25.78	25.61	25.61	22.84	1400	0.01
2022-10-27	25.62	25.76	25.48	25.48	22.72	3600	-0.01
...	...	...	...	...	...	...	...
2023-10-16	26.00	26.00	26.00	26.00	26.00	200	0.02
2023-10-17	25.83	25.92	25.83	25.92	25.92	100	-0.00
2023-10-18	25.69	25.69	25.49	25.49	25.49	1400	-0.02
2023-10-19	25.47	25.47	25.16	25.17	25.17	43900	-0.01
2023-10-20	24.85	24.98	24.85	24.89	24.89	2600	-0.01

251 rows × 7 columns

[38]:
```
sigma = np.sqrt(252) * df['returns'].std()
sigma
```

[38]:
```
0.19433064514881673
```

[39]:
```
uty = (pdr.get_data_yahoo("^TNX", start=today.replace(day=today.day-1), end=today)['Close'].iloc[-1])/100
uty
```

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

[39]:
```
0.04923999786376953
```

[40]:
```
lcp = df['Close'].iloc[-1]
lcp
```

[40]:
```
24.889999389648438
```

[41]:
```
t = (datetime.strptime(expiry, "%m-%d-%Y") - datetime.utcnow()).days / 365
t
```

[41]:
```
1.158904109589041
```

[44]:
```
print('The Call Option Price is: ', round(bs_call(lcp, strike_price_call, t, uty, sigma),4))
print('The Put Option Price is: ', round(bs_put(lcp, strike_price_put, t, uty, sigma),4))
```

```
The Call Option Price is:  0.9046
The Put Option Price is:  5.8209
```

[ ]: