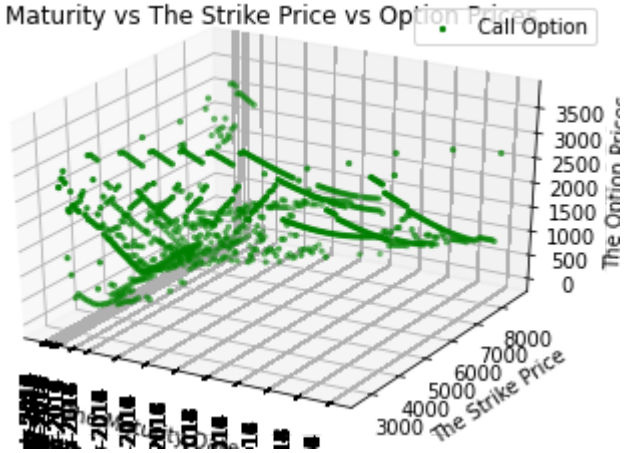
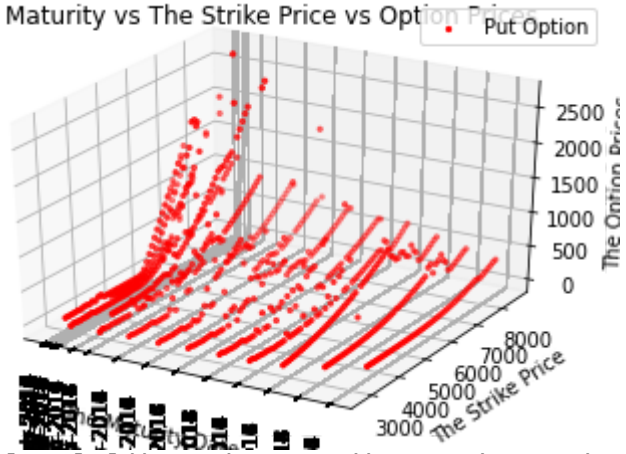
```
from IPython.core.display import display, HTML

import numpy as np
from pandas import read_csv, to_datetime
from scipy.stats import norm

from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

fields=['Expiry', 'Strike Price', 'Put Price', 'Call Price']
original_data = read_csv('OptionNifty.csv', usecols=fields, index_col=False)

optionData = read_csv("OptionNifty.csv")
stock_data = read_csv("nsedata1.csv")
optionData['Date2'] = to_datetime(optionData['Date'])
stock_data['Date2'] = to_datetime(stock_data['Date'])
stock_data = stock_data[['Date2','Close']]
data = optionData.merge(stock_data,on='Date2')
data.head()

number_sample = 1000
mask = np.random.randint(0, len(data), number_sample)
data = data.loc[mask]

data.head()

len(data)

import matplotlib.dates as mdates
plot_data = original_data[:number_sample]

dates = to_datetime(plot_data['Expiry'])
x = to_datetime(dates)
x = mdates.date2num(x)

y = plot_data['Strike Price']
z_call = plot_data['Call Price']
z_put = plot_data['Put Price']

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x, y, z_call, c='g', marker='.', label='Call Option')

plt.xticks(x, data['Expiry'], rotation=90)
ax.set_xlabel('The Maturity Date')
ax.set_ylabel('The Strike Price')
ax.set_zlabel('The Option Prices')
ax.legend()
plt.title("Maturity vs The Strike Price vs Option Prices")
plt.show()
plt.clf()
fig = plt.figure()
```

```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x, y, z_put, c='r', marker='.', label='Put Option')

plt.xticks(x, data['Expiry'], rotation=90)
ax.set_xlabel('The Maturity Date')
ax.set_ylabel('The Strike Price')
ax.set_zlabel('The Option Prices')
ax.legend()
plt.title("Maturity vs The Strike Price vs Option Prices")
plt.show()
plt.clf()

def get_call(S, K, r, t, sig):
    d1 = (np.log(S/K)+t*(r+(sig**2)/2))/(sig*(t**0.5))
    d2 = d1-sig*(t**0.5)
    Nd1 = norm.cdf(d1)
    Nd2 = norm.cdf(d2)
    C = S*Nd1 - K*np.exp(-r*t)*Nd2
    return C

def get_put(S, K, r, t, sig):
    d1 = (np.log(S/K)+t*(r+(sig**2)/2))/(sig*(t**0.5))
    d2 = d1-sig*(t**0.5)
    Nd1 = norm.cdf(-d1)
    Nd2 = norm.cdf(-d2)
    P = K*np.exp(-r*t)*Nd2 - S*Nd1
    return P
def q(Price, St, K, r, t, sig, option='Call'):
    if option is 'Call':
        return get_call(St, K, r, t, sig)-Price
    else:
        return get_put(St, K, r, t, sig)-Price
def Secant(Price, St, K, r, t, option='Call'):
    x_0 = 0.1
    x_1 = 0.2

    tol = 0.00001
    n = 100
    alpha = 0.1
    for i in range(n):
        x2 = x_1 - q(Price, St, K, r, t, x_1, option)*(x_1-x_0)/(q(Price, St, K, r
        x_0 = x_1
        x_1 = x2

        if abs(q(Price, St, K, r, t, x_1, option)) < tol:
            break
    return x_1

from datetime import datetime

n = len(data)
sigma_c = np.zeros(n)
for i in range(n):
    St = data.iloc[-i]['Close']
```

```
    r = 0.05
    init_date=data.iloc[-i]['Date']
    expiry_date=data.iloc[-i]['Expiry']

    date_format = "%d-%b-%Y"
    d0 = datetime.strptime(init_date, date_format)
    d1 = datetime.strptime(expiry_date, date_format)
    t = (d1-d0).days/252
    K = data.iloc[-i]['Strike Price']
    P = data.iloc[-i]['Put Price']
    C = data.iloc[-i]['Call Price']

    sigma_c[i] = Secant(C, St, K, r, t, 'Call')
    if abs(sigma_c[i]) > 10:
        sigma_c[i] = np.nan


data.head()

data['Volatility']=sigma_c
data.drop(['Date2'], axis=1)
data.to_csv('result.csv', index=False)

def plotVolatility(data):
    dates = to_datetime(data['Expiry'])
    x = to_datetime(dates)
    x = mdates.date2num(x)

    y = data['Strike Price']
    z = data['Volatility']

    fig = plt.figure()
    ax = fig.add_subplot( projection='3d')

    ax.scatter(x, y, z, marker='.')

    plt.xticks(x, data['Expiry'], rotation=90)
    ax.set_xlabel('The Maturity Date')
    ax.set_ylabel('The Strike Price')
    ax.set_zlabel('The Volatility')
    ax.legend()
    plt.title('Maturity vs The Strike Price vs Volatility')
    plt.show()

plotVolatility(data)
```




```
<Figure size 432x288 with 0 Axes>
```




```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:69: RuntimeWarni
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:69: RuntimeWarni
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:96: RuntimeWarni
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:69: RuntimeWarni
No handles with labels found to put in legend.
<Figure size 432x288 with 0 Axes>
```