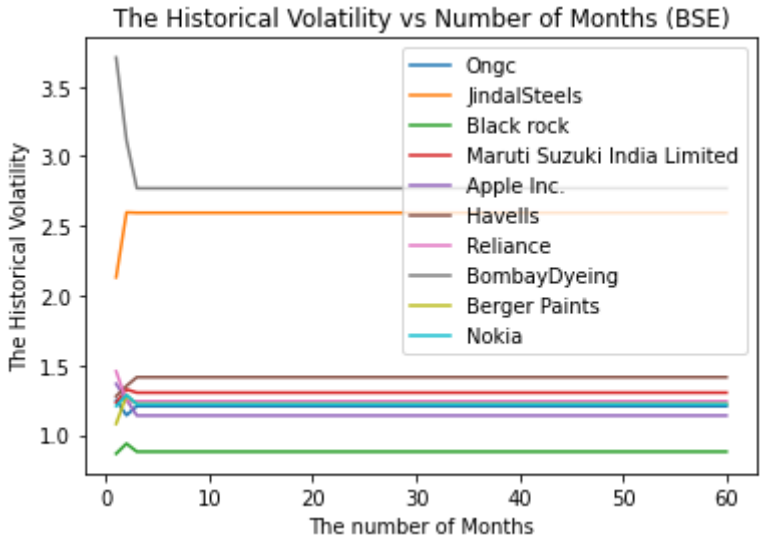
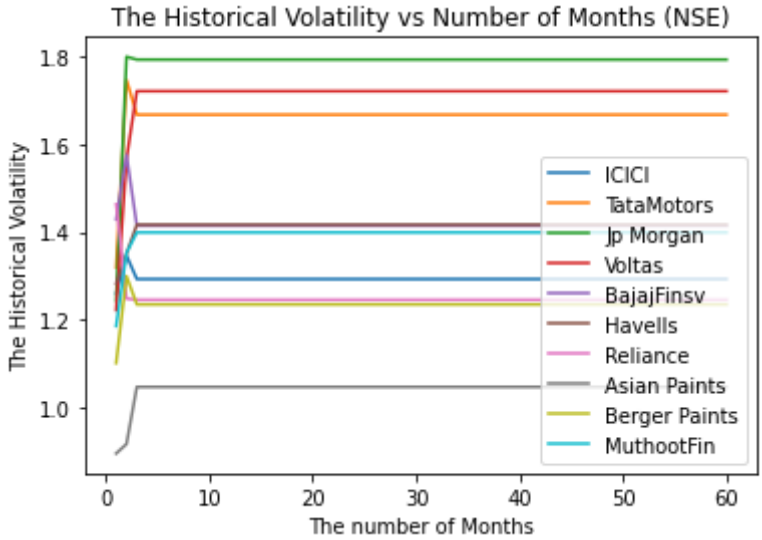
```
#Question No: 03
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from math import erf

plt.rcParams.update({'figure.max_open_warning': 0})

def clean_data(X):
    y = [x for x in X if not np.isnan(x)]
    return y

def read_data(filename):
    df = pd.read_csv(filename)
    df.set_index('Date',inplace=True)
    data = df.to_dict()
    for key, vals in data.items():
        data[key] = clean_data(list(vals.values()))
    return data

def historical_volatility(prices, duration):
    required_price = prices[-duration:]
    R = []
    for i in range(1, len(required_price)):
        r = (required_price[i] - required_price[i-1])/required_price[i-1]
        R.append(r)


    var = np.var(R)
    sigma_d = np.sqrt(var)
    sigma_a = np.sqrt(252)*sigma_d
    return sigma_a

def D_plus_and_minus(x, tau, sig, K, r):
    D_plus = (1/(sig*np.sqrt(tau)))*(np.log(x/K) + tau*(r + (sig*sig)/2))
    D_minus = (1/(sig*np.sqrt(tau)))*(np.log(x/K) + tau*(r - (sig*sig)/2))

    return D_plus, D_minus

def Normal(x):
  return 0.5*(1 + erf(x/np.sqrt(2)))

def Call(t, x, T, sig, K, r):
    if x == 0:
        return 0
    if t == T:
        return max(x - K, 0)

    tau = T-t
    D_plus, D_minus = D_plus_and_minus(x, tau, sig, K, r)

    price = x*Normal(D_plus) - K*np.exp(-r*tau)*Normal(D_minus)

    return price
```

```
    return price

def Put(t, x, T, sig, K, r):
    call = Call(t,x,T,sig,K,r)
    put = call + K*np.exp(-r*(T-t)) - x

    return put

bse_data = read_data('bsedata1.csv')
nse_data = read_data('nsedata1.csv')

r = 0.05
T = 0.5
t = 0
n_days_month = 20

A = np.arange(0.5, 1.51, 0.1)

for company, prices in bse_data.items():
    months = range(1, 61)
    vols = []
    for i in range(1, 61):
        days_in_month = i*n_days_month
        vol = historical_volatility(prices, days_in_month)
        vols.append(vol)

    plt.plot(months, vols, label = company)

plt.xlabel("The number of Months")
plt.ylabel("The Historical Volatility")
plt.title("The Historical Volatility vs Number of Months (BSE)")
plt.legend(loc = "best")
plt.show()
plt.savefig("q3_BSE_volatility")
plt.clf()

for company, prices in nse_data.items():
    months = range(1, 61)
    vols = []
    for i in range(1, 61):
        days_in_month = i*n_days_month
        vol = historical_volatility(prices, days_in_month)
        vols.append(vol)

    plt.plot(months, vols, label = company)

plt.xlabel("The number of Months")
plt.ylabel("The Historical Volatility")
plt.title("The Historical Volatility vs Number of Months (NSE)")
plt.legend(loc = "best")
plt.show()
plt.savefig("q3_NSE_volatility")
plt.clf()

for company, prices in bse_data.items():
    S0 = prices[-1]
```

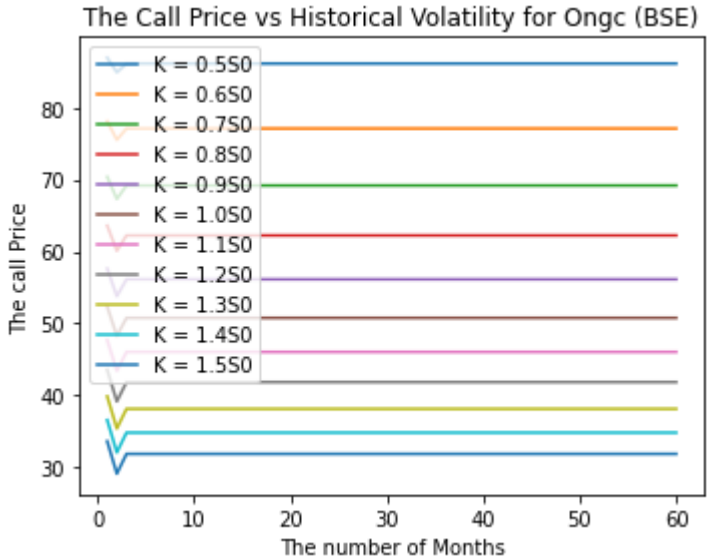
```
    S0 = prices[-1]
    months = range(1,61)
    fig = plt.figure(figsize=(5.6,4.2))
    for a in A:
        K = a*S0
        call_prices= []
        for i in range(1,61):
            days_in_month = i*n_days_month
            sig = historical_volatility(prices, days_in_month)
            call = Call(t, S0, T, sig, K, r)
            call_prices.append(call)

        plt.plot(months, call_prices, label = "K = %.1fS0"%a)

    plt.xlabel("The number of Months")
    plt.ylabel("The call Price")
    plt.title(f"The Call Price vs Historical Volatility for {company} (BSE)")
    plt.legend(loc = "best")
    plt.show()
    plt.clf()

for company, prices in nse_data.items():
    S0 = prices[-1]
    months = range(1,61)
    fig = plt.figure(figsize=(5.6,4.2))
    for a in A:
        K = a*S0
        call_prices= []
        for i in range(1,61):
            days_in_month = i*n_days_month
            sig = historical_volatility(prices, days_in_month)
            call = Call(t, S0, T, sig, K, r)
            call_prices.append(call)

        plt.plot(months, call_prices, label = "K = %.1fS0"%a)

    plt.xlabel("The number of Months")
    plt.ylabel("The call Price")
    plt.title(f"The call Price vs Historical Volatility for {company} (NSE)")
    plt.legend(loc = "best")
    plt.show()
    plt.clf()

for company, prices in bse_data.items():
    S0 = prices[-1]
    months = range(1,61)
    fig = plt.figure(figsize=(5.6,4.2))
    for a in A:
        K = a*S0
        put_prices= []
        for i in range(1,61):
            days_in_month = i*n_days_month
            sig = historical_volatility(prices, days_in_month)
            put = Put(t, S0, T, sig, K, r)
            put_prices.append(put)
```

```
        plt.plot(months, put_prices, label = "K = %.1fS0"%a)

    plt.xlabel("The number of Months")
    plt.ylabel("The Put Price")
    plt.title(f"The Put Price vs Historical Volatility for {company} (BSE)")
    plt.legend(loc = "best")
    plt.show()
    plt.clf()

for company, prices in nse_data.items():
    S0 = prices[-1]
    months = range(1,61)
    fig = plt.figure(figsize=(5.6,4.2))
    for a in A:
        K = a*S0
        put_prices= []
        for i in range(1,61):
            days_in_month = i*n_days_month
            sig = historical_volatility(prices, days_in_month)
            put = Put(t, S0, T, sig, K, r)
            put_prices.append(put)

        plt.plot(months, put_prices, label = "K = %.1fS0"%a)

    plt.xlabel("The number of Months")
    plt.ylabel("The Put Price")
    plt.title(f"The Put Price vs Historical Volatility for {company} (NSE)")
    plt.legend(loc = "best")
    plt.show()
    plt.clf()
```

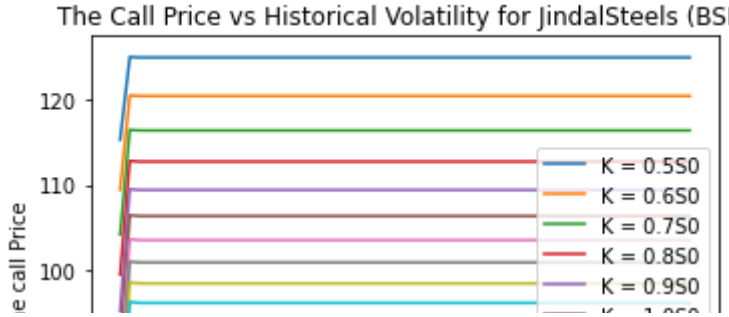


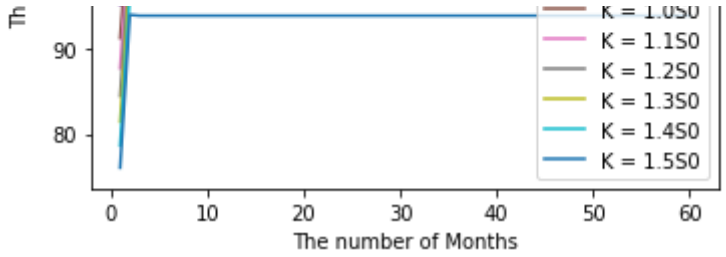


<Figure size 432x288 with 0 Axes>

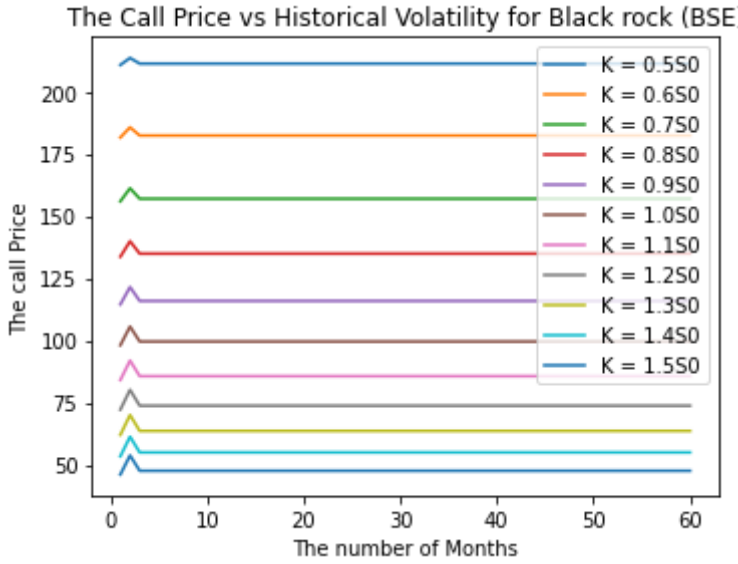


<Figure size 432x288 with 0 Axes>

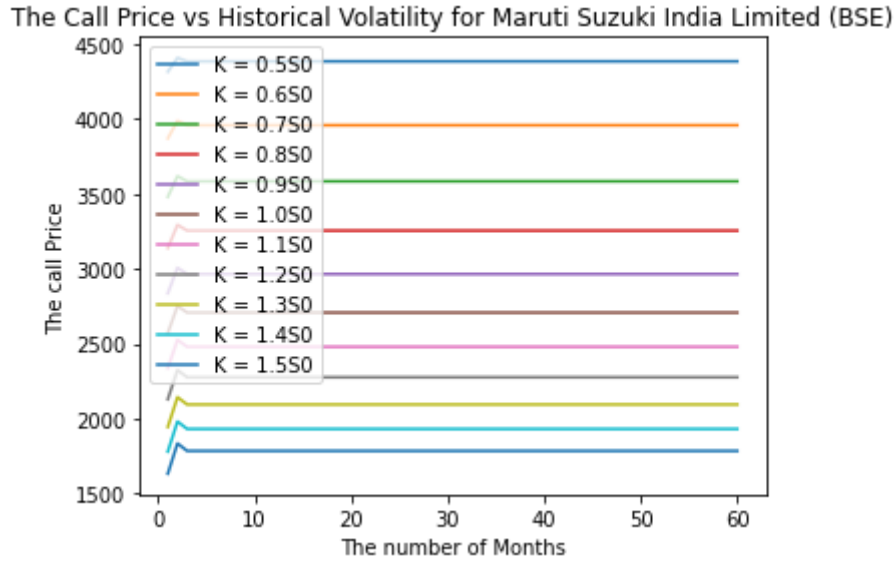




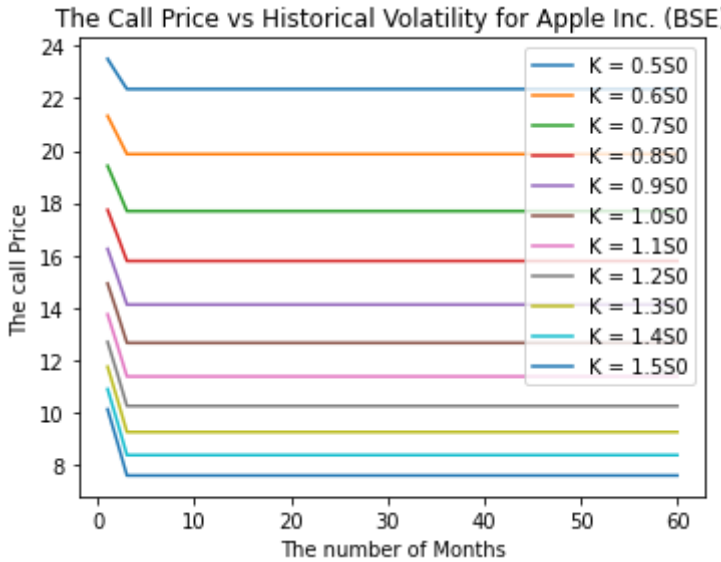
<Figure size 432x288 with 0 Axes>



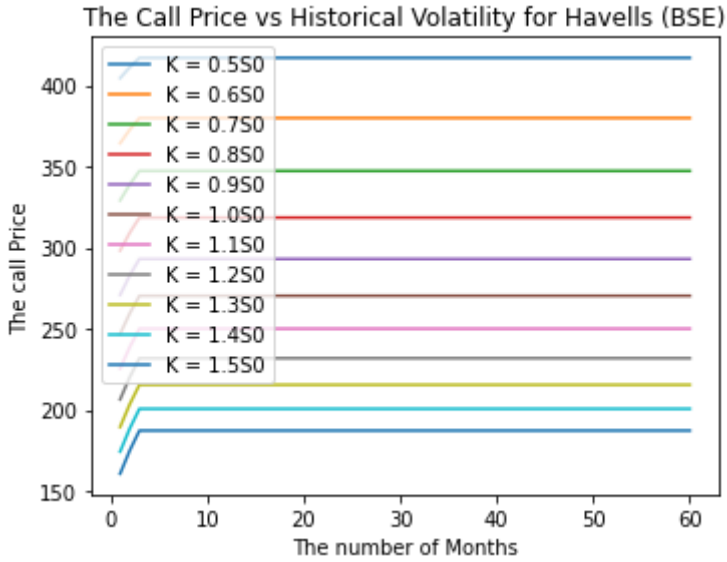
<Figure size 432x288 with 0 Axes>



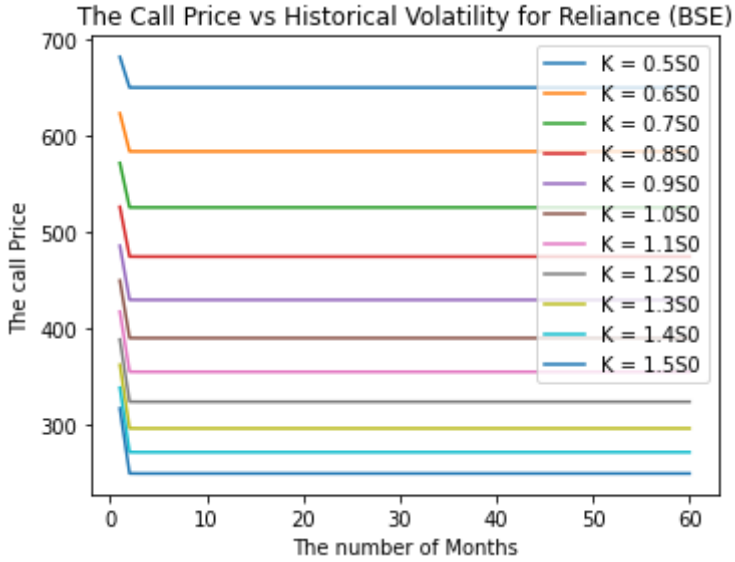
<Figure size 432x288 with 0 Axes>



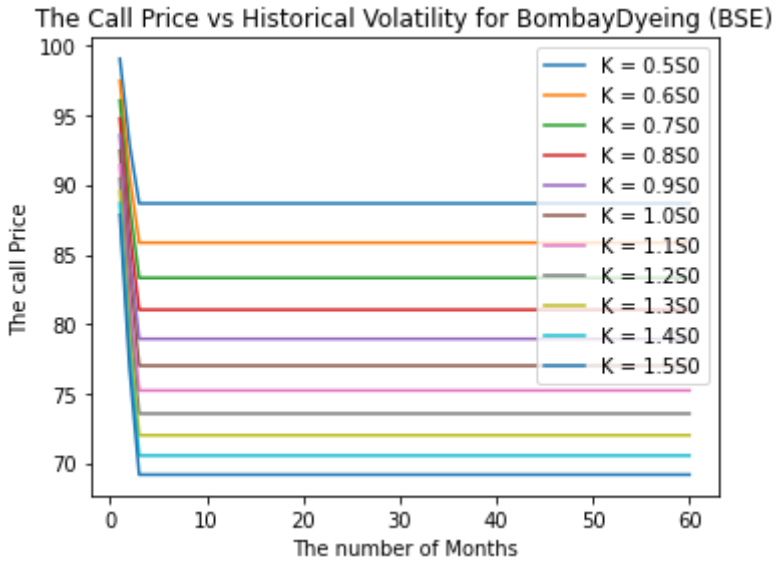
<Figure size 432x288 with 0 Axes>



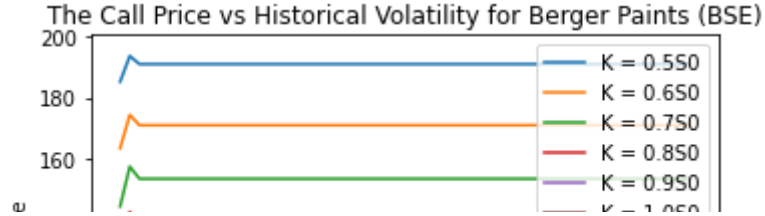
<Figure size 432x288 with 0 Axes>

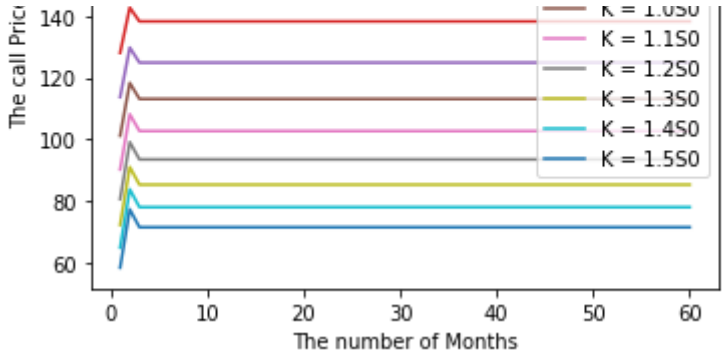


<Figure size 432x288 with 0 Axes>

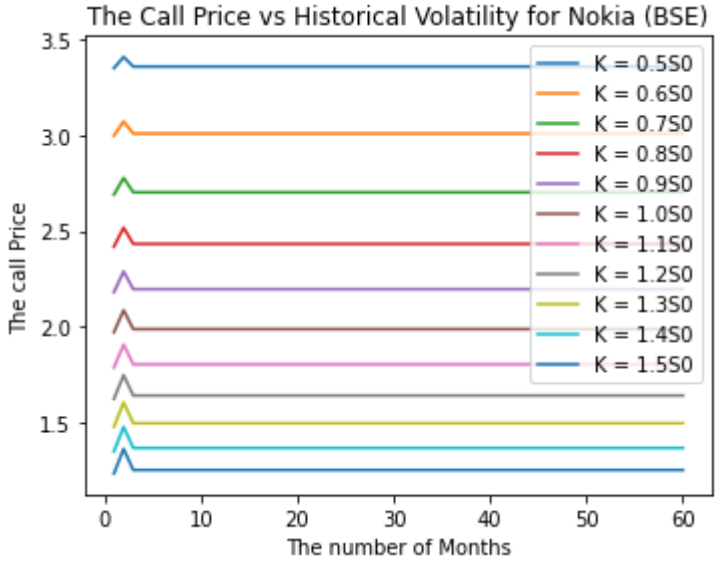


<Figure size 432x288 with 0 Axes>

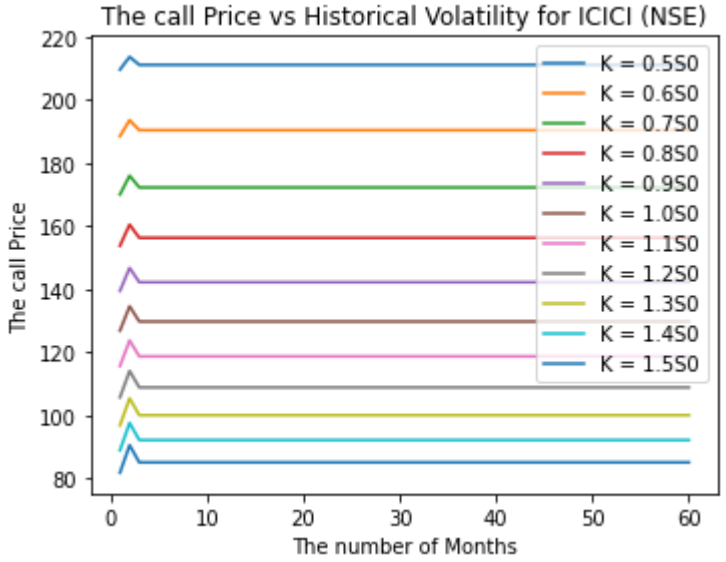




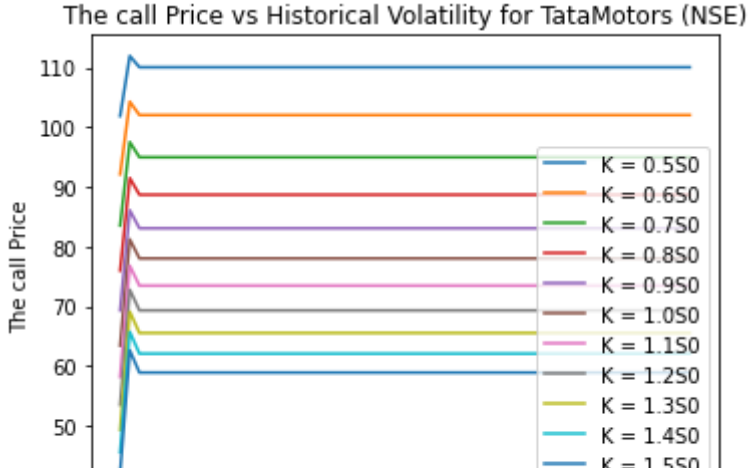
<Figure size 432x288 with 0 Axes>

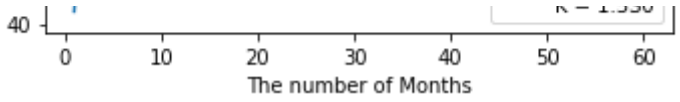


<Figure size 432x288 with 0 Axes>

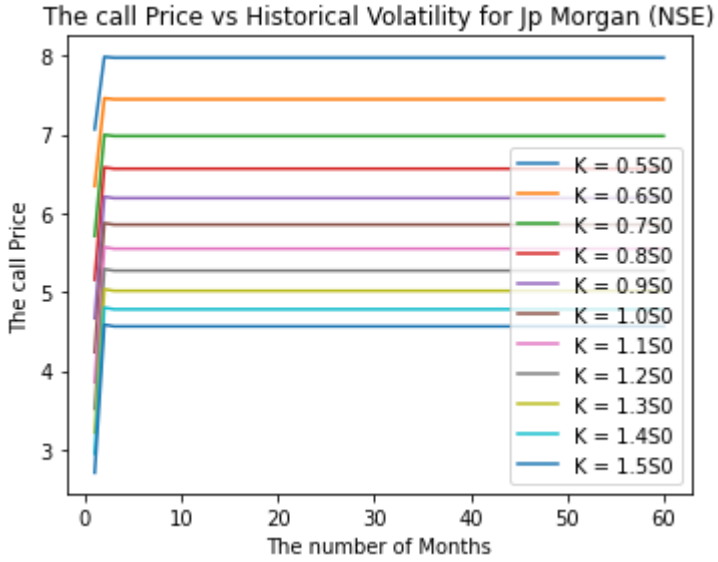


<Figure size 432x288 with 0 Axes>

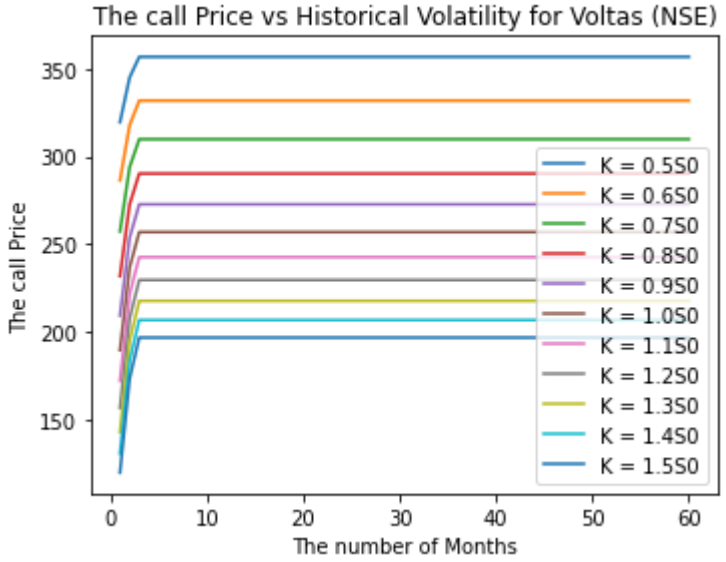




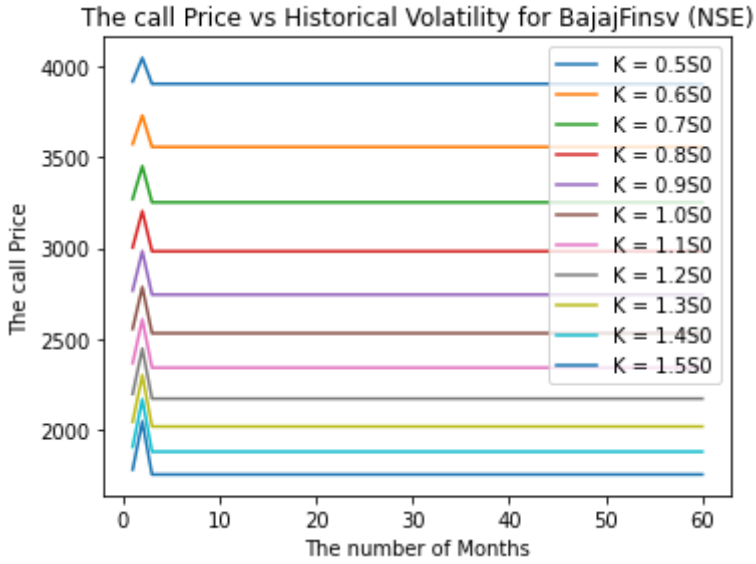
<Figure size 432x288 with 0 Axes>



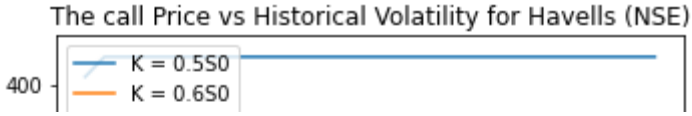
<Figure size 432x288 with 0 Axes>

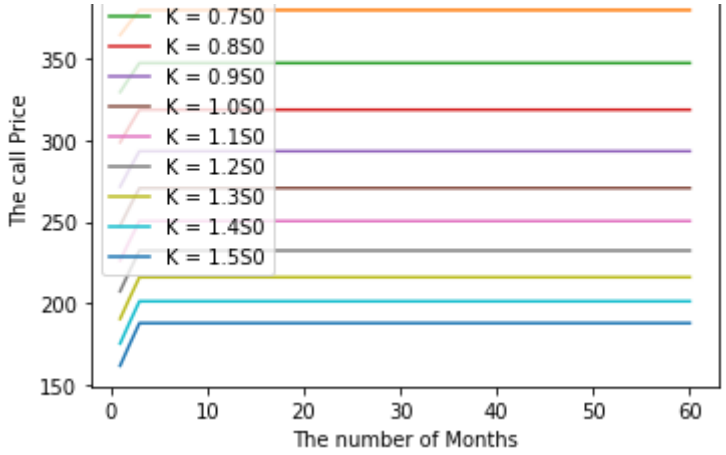


<Figure size 432x288 with 0 Axes>

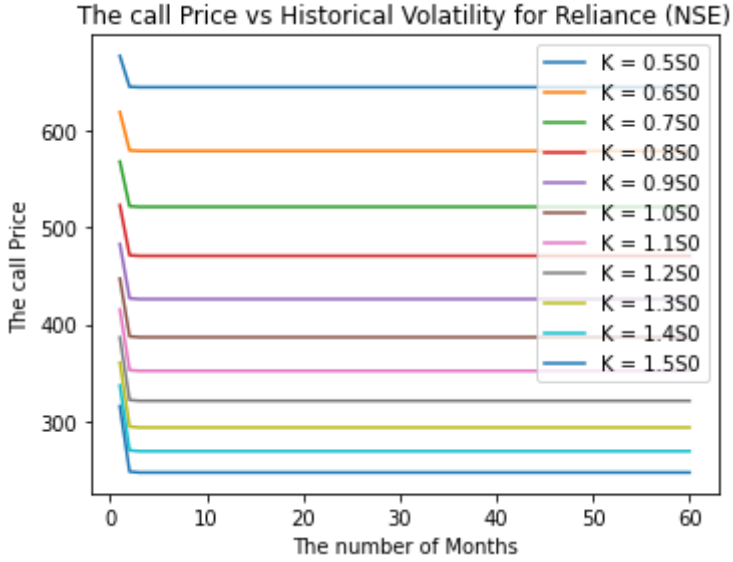


<Figure size 432x288 with 0 Axes>

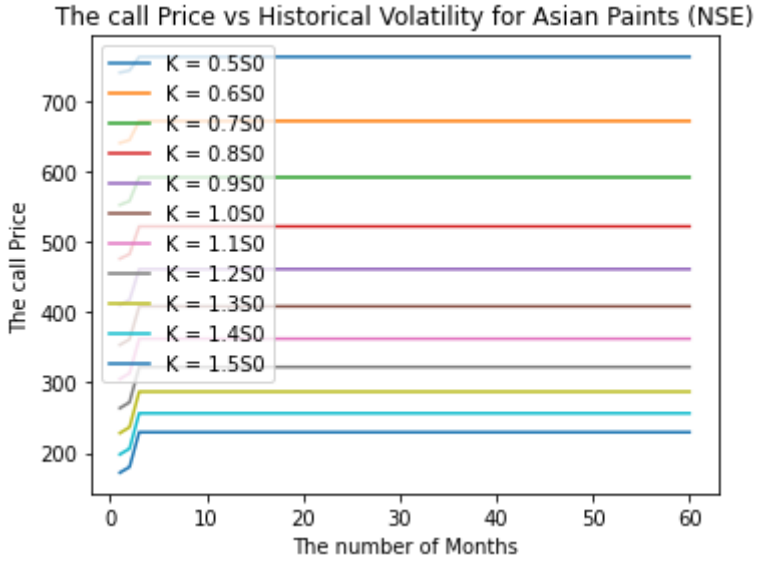




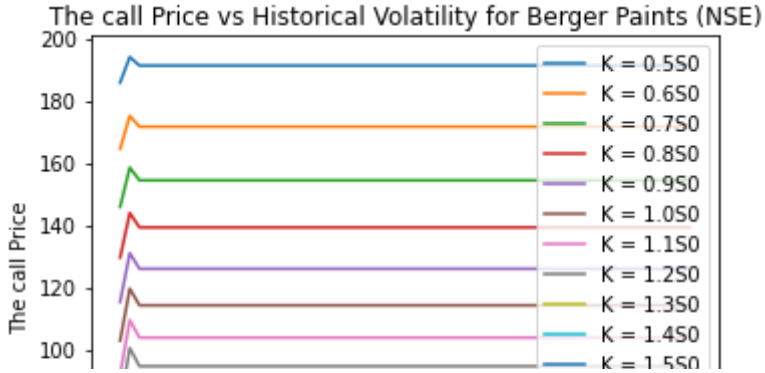
<Figure size 432x288 with 0 Axes>

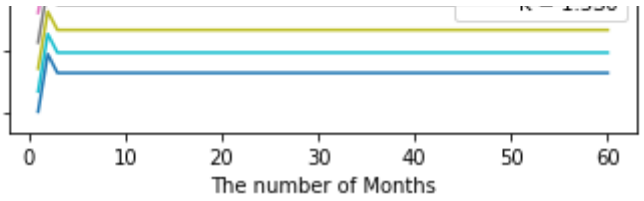


<Figure size 432x288 with 0 Axes>

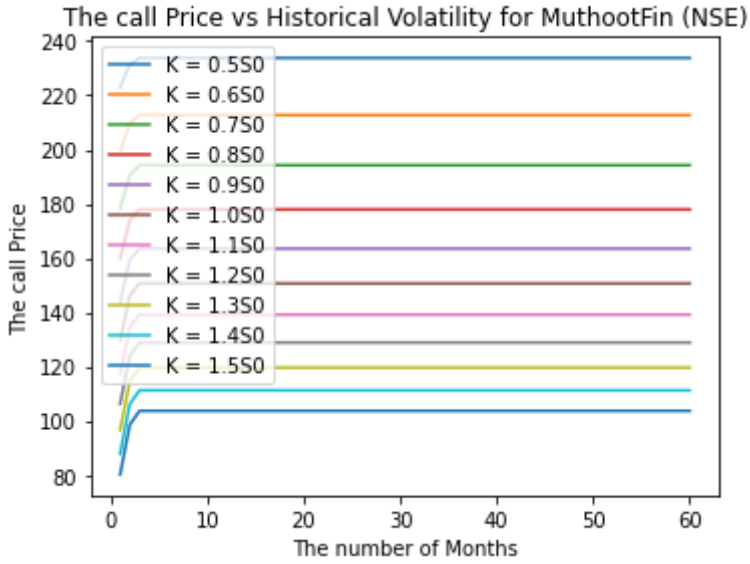


<Figure size 432x288 with 0 Axes>

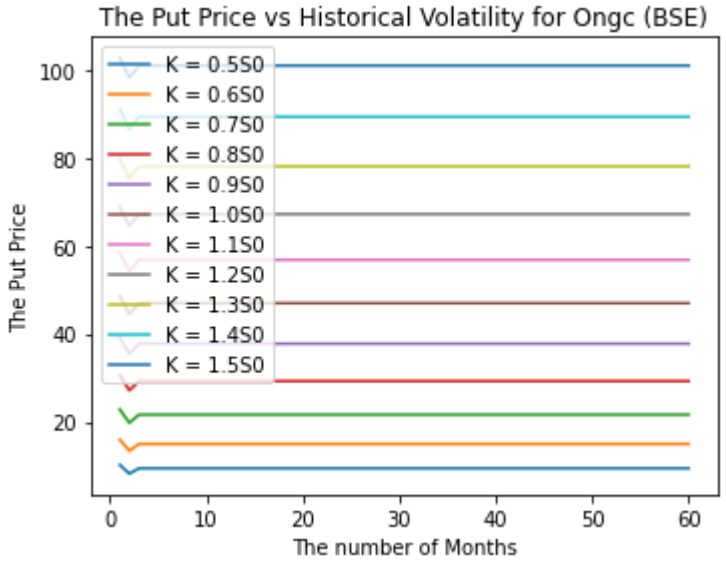




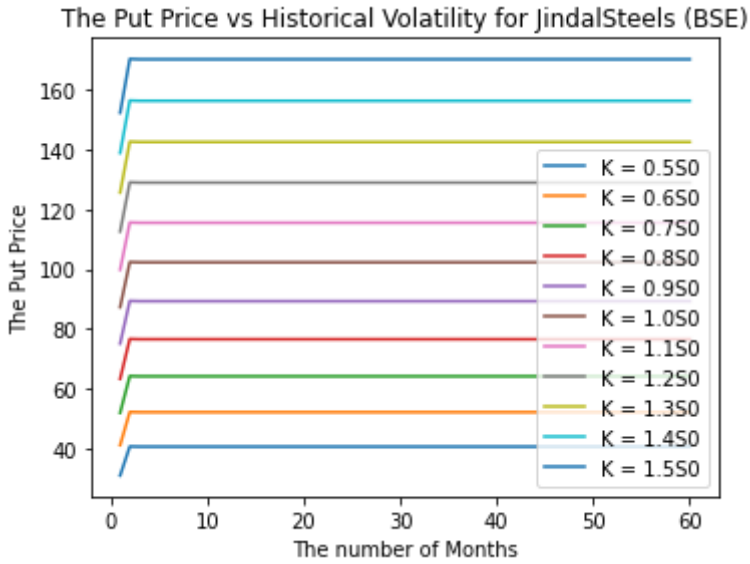
<Figure size 432x288 with 0 Axes>



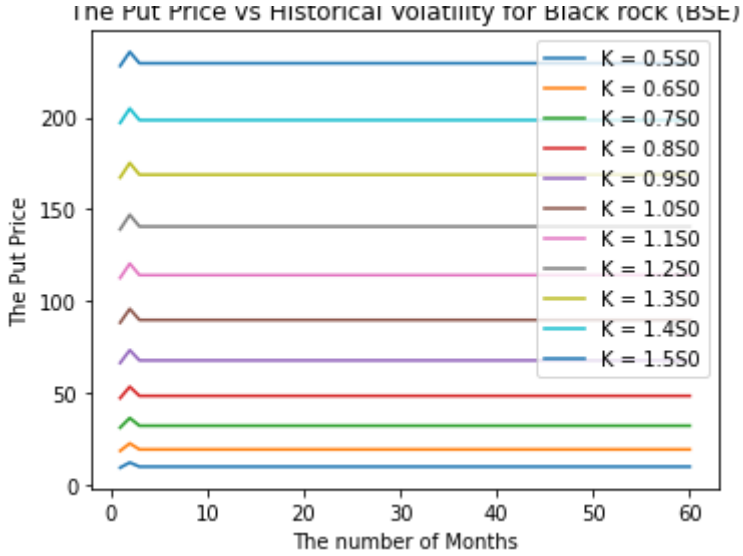
<Figure size 432x288 with 0 Axes>



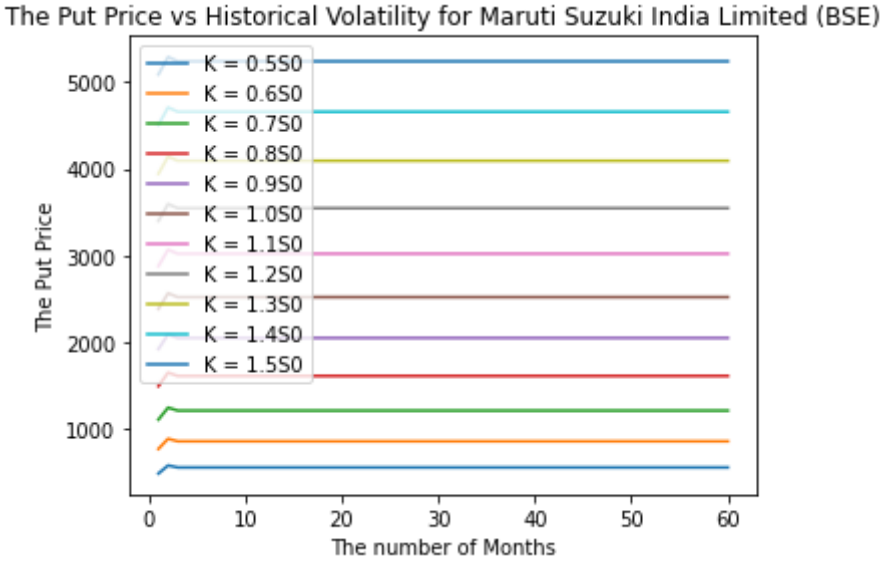
<Figure size 432x288 with 0 Axes>



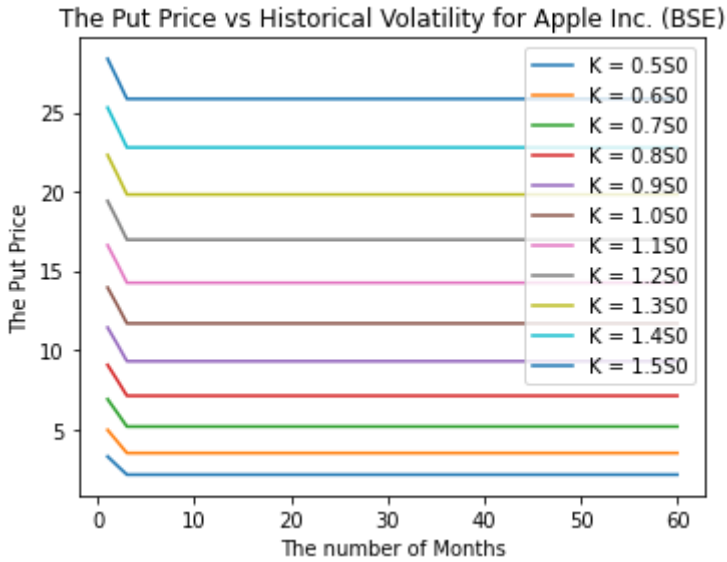
<Figure size 432x288 with 0 Axes>



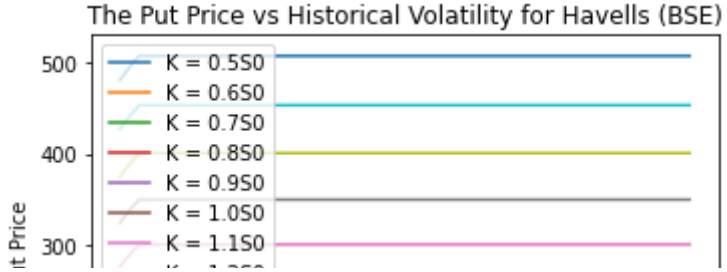
<Figure size 432x288 with 0 Axes>

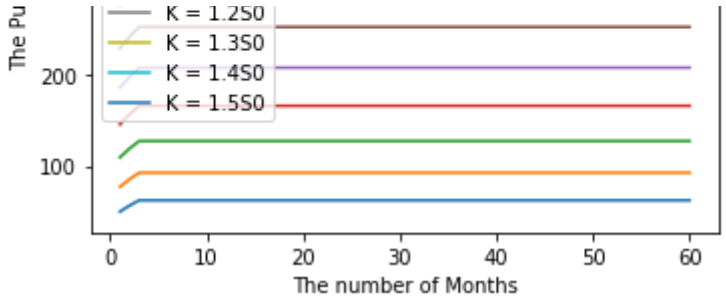


<Figure size 432x288 with 0 Axes>

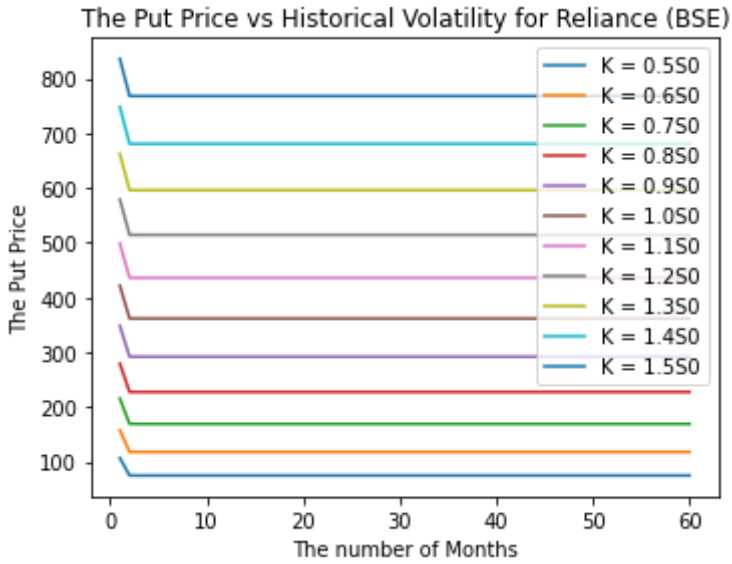


<Figure size 432x288 with 0 Axes>

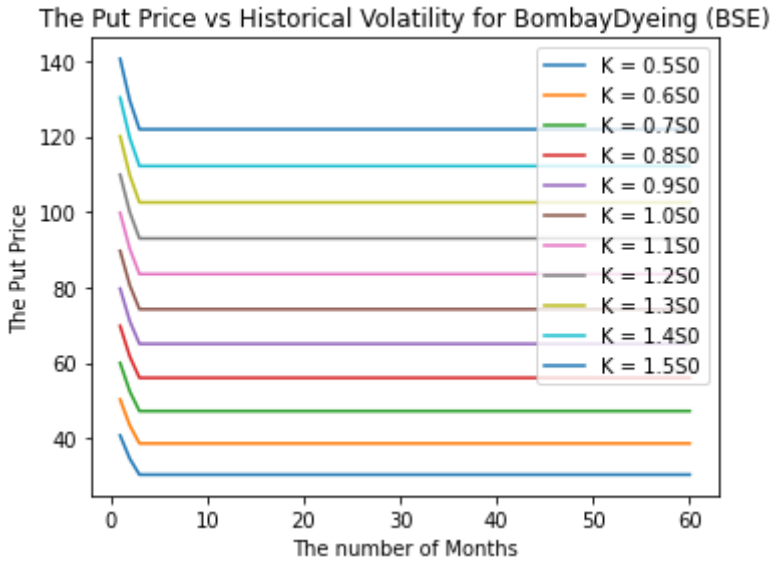




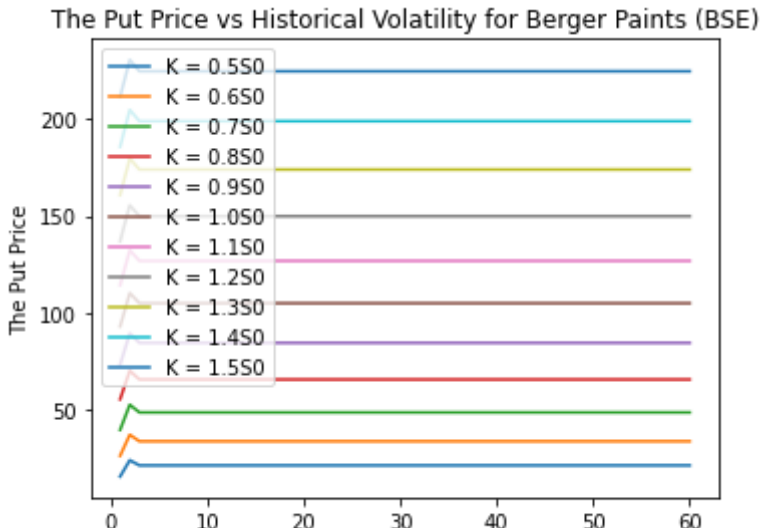
<Figure size 432x288 with 0 Axes>

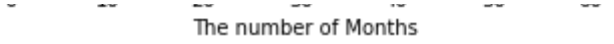


<Figure size 432x288 with 0 Axes>

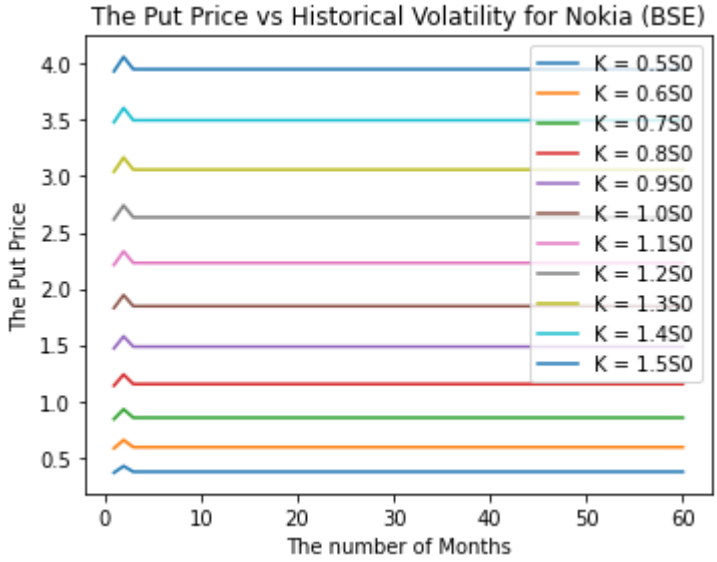


<Figure size 432x288 with 0 Axes>

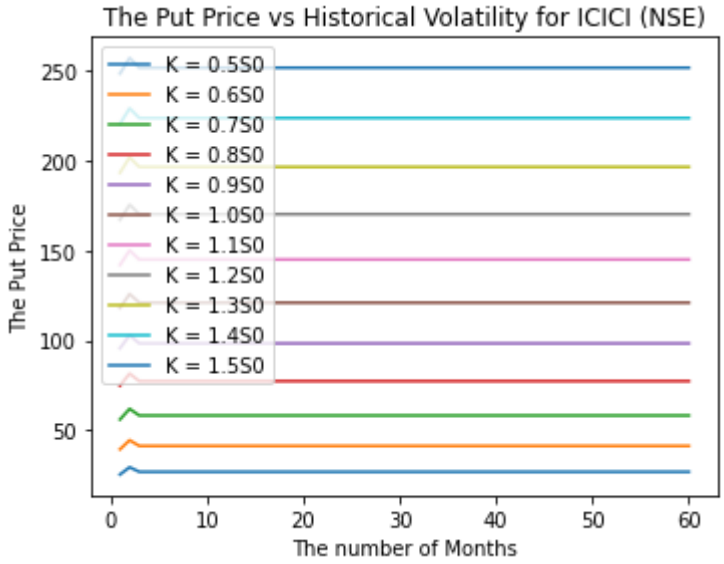




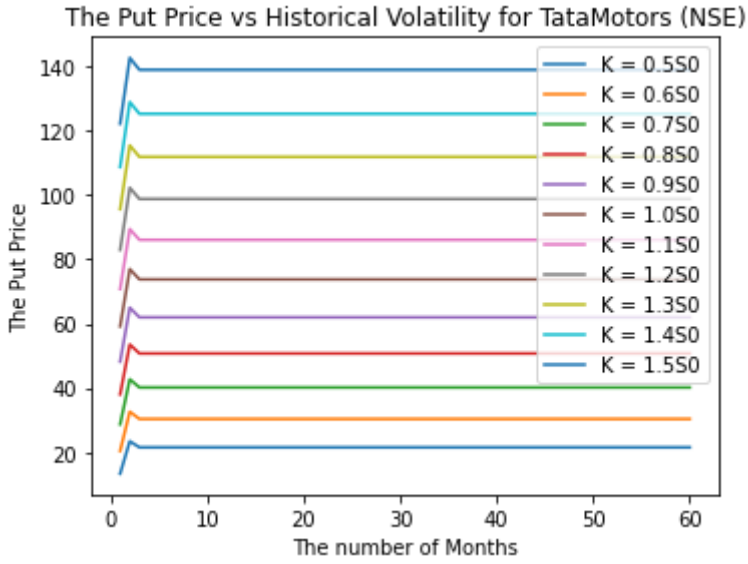
<Figure size 432x288 with 0 Axes>



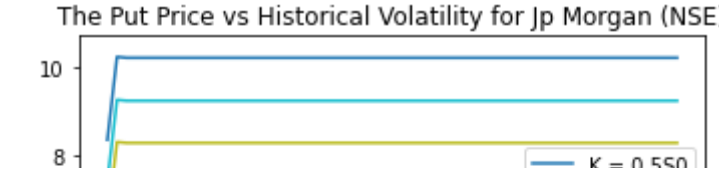
<Figure size 432x288 with 0 Axes>

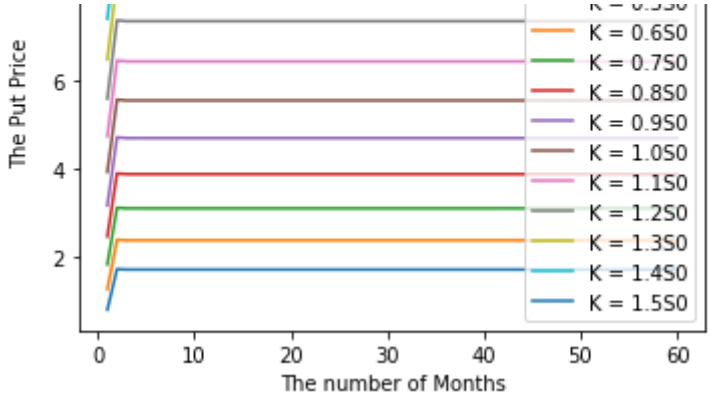


<Figure size 432x288 with 0 Axes>

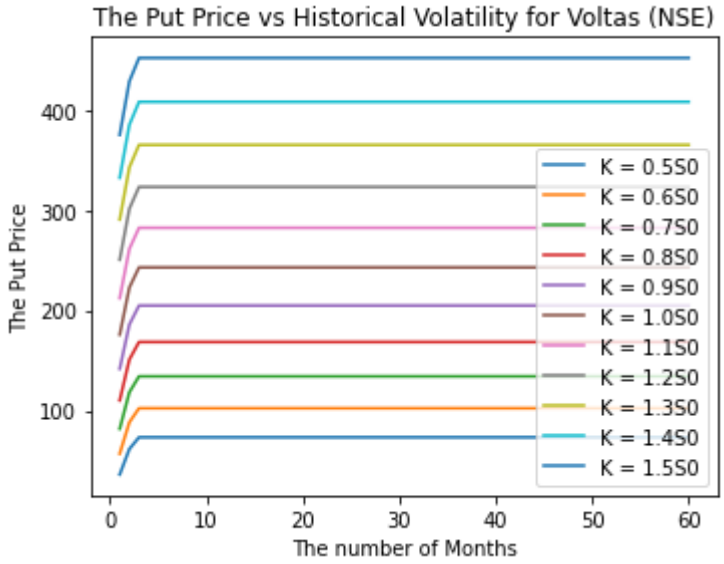


<Figure size 432x288 with 0 Axes>

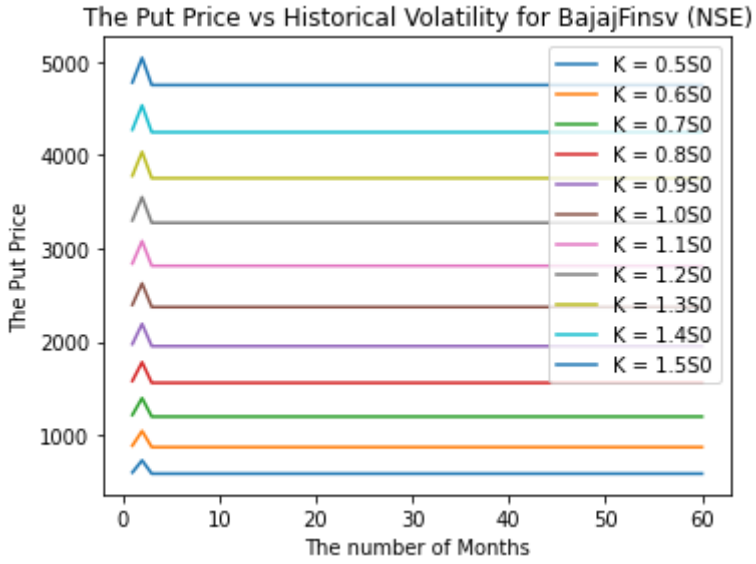




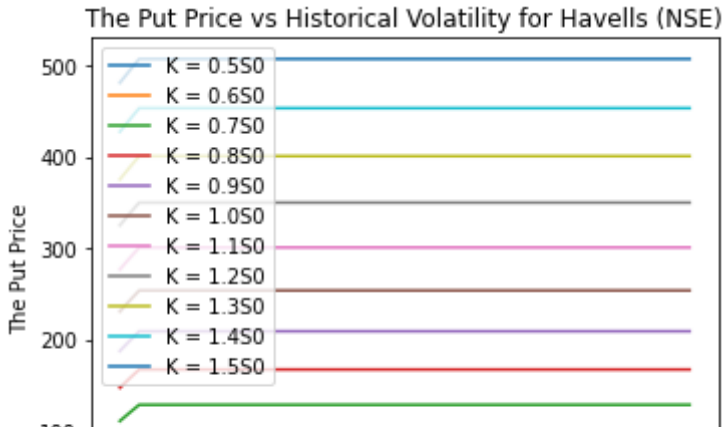
<Figure size 432x288 with 0 Axes>

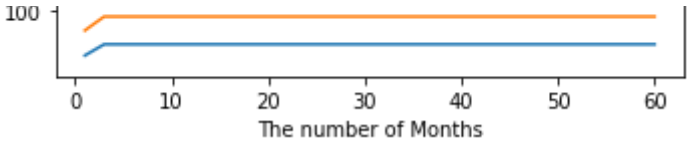


<Figure size 432x288 with 0 Axes>

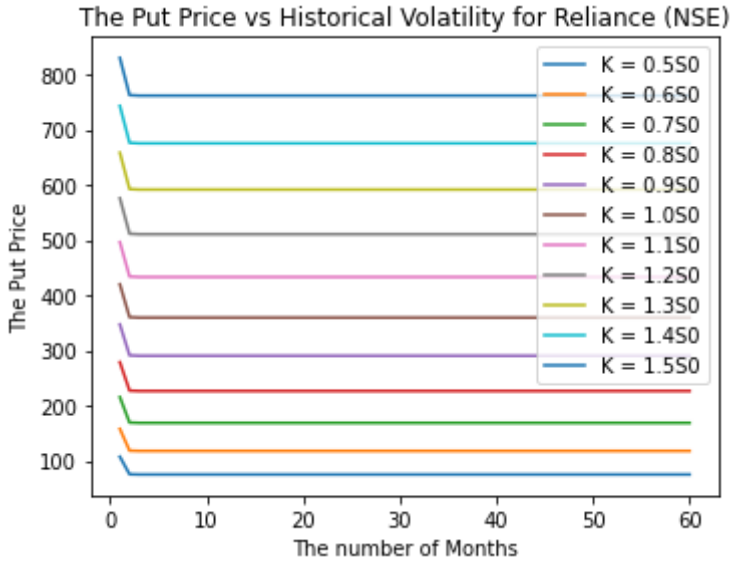


<Figure size 432x288 with 0 Axes>

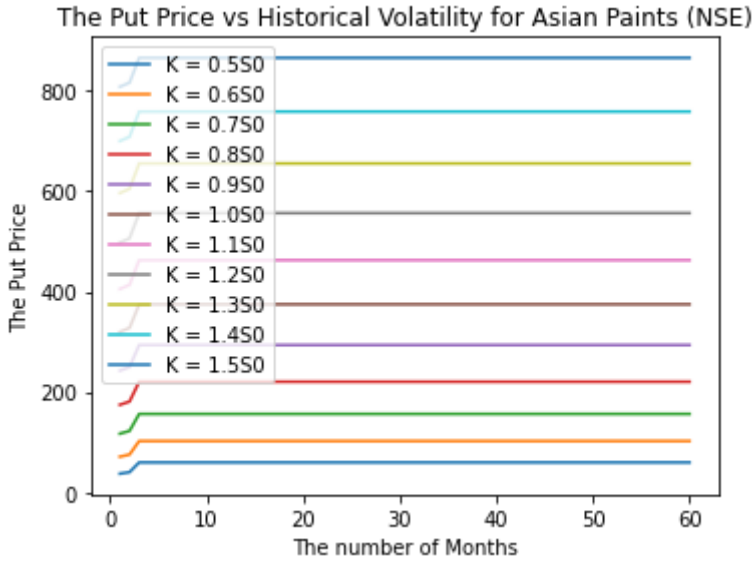




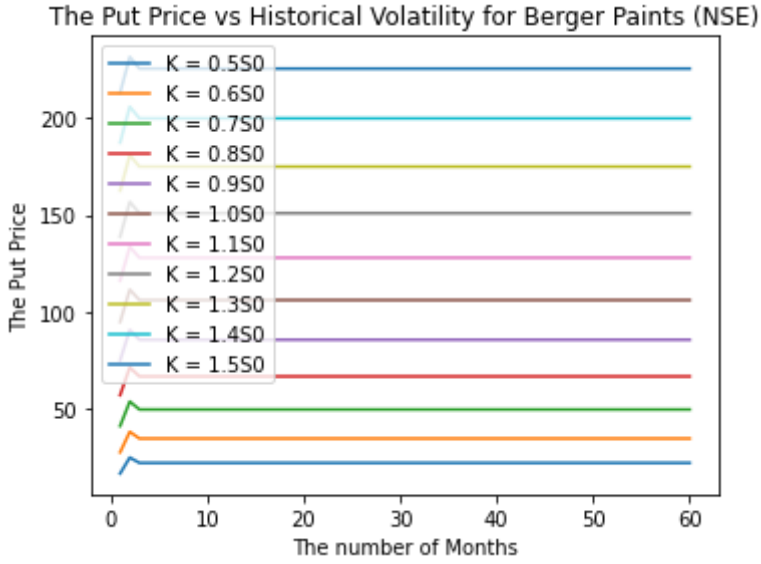
<Figure size 432x288 with 0 Axes>



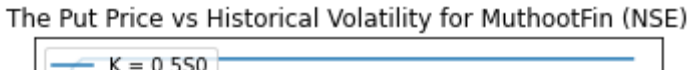
<Figure size 432x288 with 0 Axes>

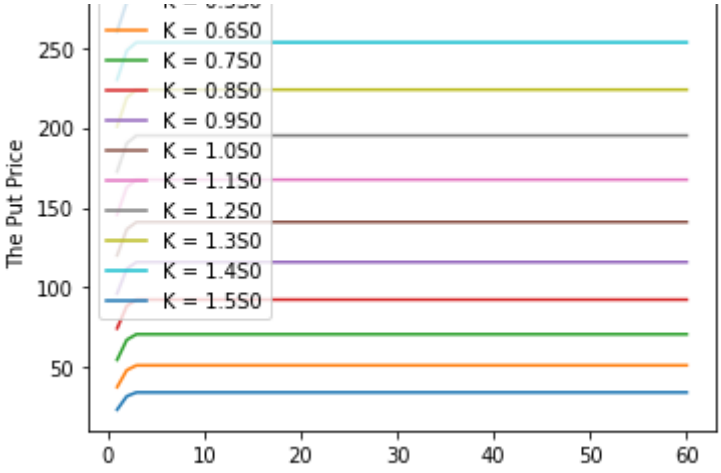


<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>




K = 0.650
K = 0.750
K = 0.850
K = 0.950
K = 1.050
K = 1.150
K = 1.250
K = 1.350
K = 1.450
K = 1.550
The Put Price
250
200
150
100
50
0
10
20
30
40
50
60