MA374 Extra Assignment

Sensitivity Analysis of Exotic Options

Name - Shivendu Mishra
Roll Number - 200123050
email id - m.shivendu@iitg.ac.in

Contents

Problem Description

Exotic options are a class of financial derivatives that have payoffs that are more complex than those of traditional options, such as vanilla call and put options. They are called "exotic" because their payoffs are usually based on more complex financial variables than just the price of the underlying asset.These variables can include interest rates, foreign exchange rates, commodity prices, and even weather conditions.

Exotic options are important because they provide investors with a wide range of financial tools that allow them to manage their risk exposure in more sophisticated ways than traditional options. Exotic options can be customized to meet specific needs of investors, allowing them to hedge against particular risks or take advantage of unique market conditions.

For example, gap options can be used to manage risk in markets that are volatile, providing investors with a way to limit their downside while still participating in the upside potential. Shout options, on the other hand, can be used to capture profits in markets that are trending strongly, allowing investors to lock in gains while still leaving room for further upside.

In the course of MA373, we have learned about many types of exotic options which include gap options, shout options, and binary options. Each of these options has a unique payoff structure and requires a different pricing formula. In this assignment, I have discussed the basic concepts of each of these options and their pricing formulae along with the sensitivity analysis of each of them.

Starting from the next page I have presented the detailed explanation of each of these option,proof of the pricing formulae and the sensitivity analysis plots.

Gap Option

A binary option is a type of financial derivative where the payoff depends on whether the underlying asset's price at maturity is above or below a predetermined strike price. Specifically, a binary call option pays out a fixed amount if the price of the underlying asset at maturity is above the strike price, while a binary put option pays out a fixed amount if the price of the underlying asset at maturity is below the strike price. If the price of the underlying asset is at or near the strike price at maturity, the option may expire worthless.

The formula for the payout of a binary call option is:

$$C_{binary} = \begin{cases} 1 & \text{if } S_T > K \\ 0 & \text{otherwise} \end{cases}$$

where S_T is the price of the underlying asset at maturity and K is the strike price.

The formula for the payout of a binary put option is:

$$P_{binary} = \begin{cases} 1 & \text{if } S_T < K \\ 0 & \text{otherwise} \end{cases}$$

where the variables have the same meaning as in the binary call option formula.

Binary Option Pricing Formulae

The pricing of binary options can also be calculated using numerical methods, such as Monte Carlo simulations, or analytic methods, such as the Black-Scholes model with a modification to account for the binary nature of the option.

The formula for pricing a binary call option using the Black-Scholes model is:

$$C_{binary} = Qe^{-rT}\Phi(d_2)$$

where r is the risk-free interest rate, T is the time to maturity, d_2 is defined as:

$$d_2 = \frac{\ln(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}}$$

and $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

The formula for pricing a binary put option is:

$$P_{binary} = Qe^{-rT}\Phi(-d_2)$$

where P_{binary} is the price of the binary put option and the other variables have the same meanings as in the binary call option formula.

Note that the Black-Scholes model assumptions and limitations discussed for the gap option also apply to binary options. Additionally, binary options may be subject to certain

regulatory requirements and are not available in all jurisdictions.I will now prove the pricing formulae for the binary option:-

$$P = Qe^{-rT}N(d_2)$$

To derive this formula, we first note that the option has a fixed payoff, which means that the value of the option depends only on the probability of it expiring in the money. Using the risk-neutral valuation principle, we can assume that the expected payoff of the option is equal to its present value.

Therefore, we can write:

$$P = e^{-rT}E[Q]$$

To calculate $E[Q]$, we use the Black-Scholes model to derive the distribution of S_T. Since the payoff of the option is a step function, we only need to consider the probability of S_T being greater than or equal to K.

Using the Black-Scholes model, we can derive the following expression for the probability that S_T is greater than or equal to K:

$$N(d_2) = P(S_T >= K)$$

where d_2 is the standard normal distribution variable defined above.

Substituting this expression into the equation for the expected payoff, we get:

$$P = e^{-rT}E[Q]$$

$$= Qe^{-rT}[1 * P(S_T >= K) + 0 * P(S_T < K)]$$

$$= Qe^{-rT}P(S_T >= K)$$

$$= Qe^{-rT}N(d_2)$$

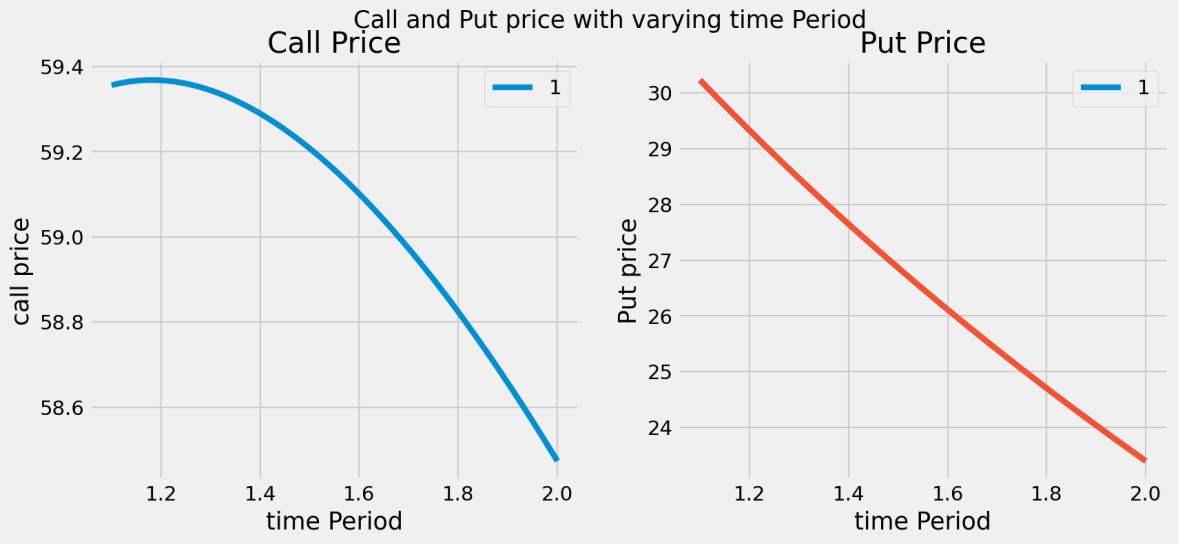
which is the formula for pricing a binary option with strike K and payoff Q. Below is the sensitivity analysis of Binary Option along with all the relevant plots:-

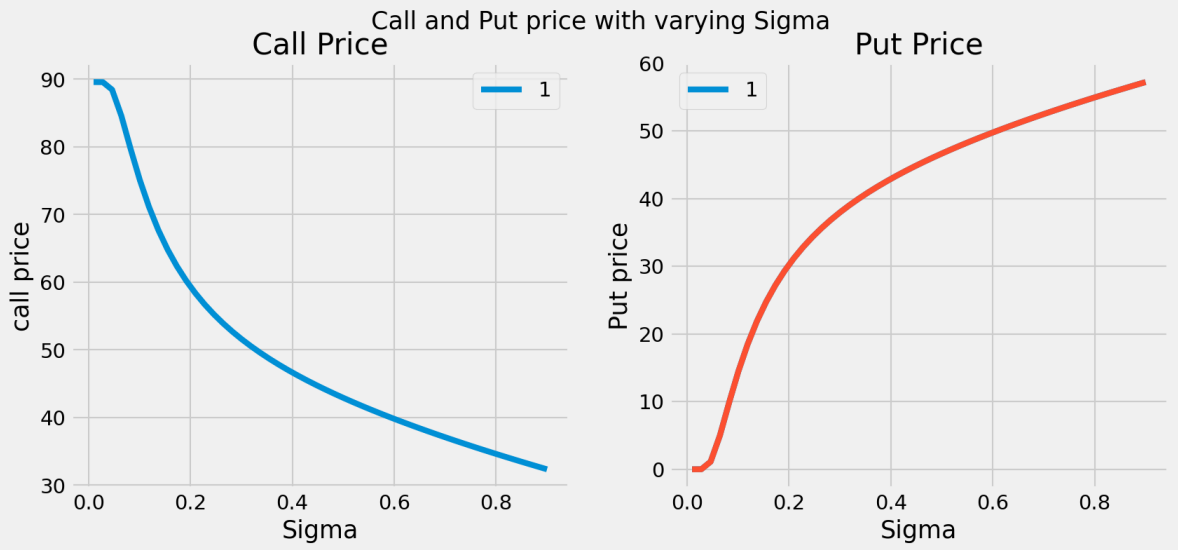
In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use("fivethirtyeight")
from scipy.stats import norm
```

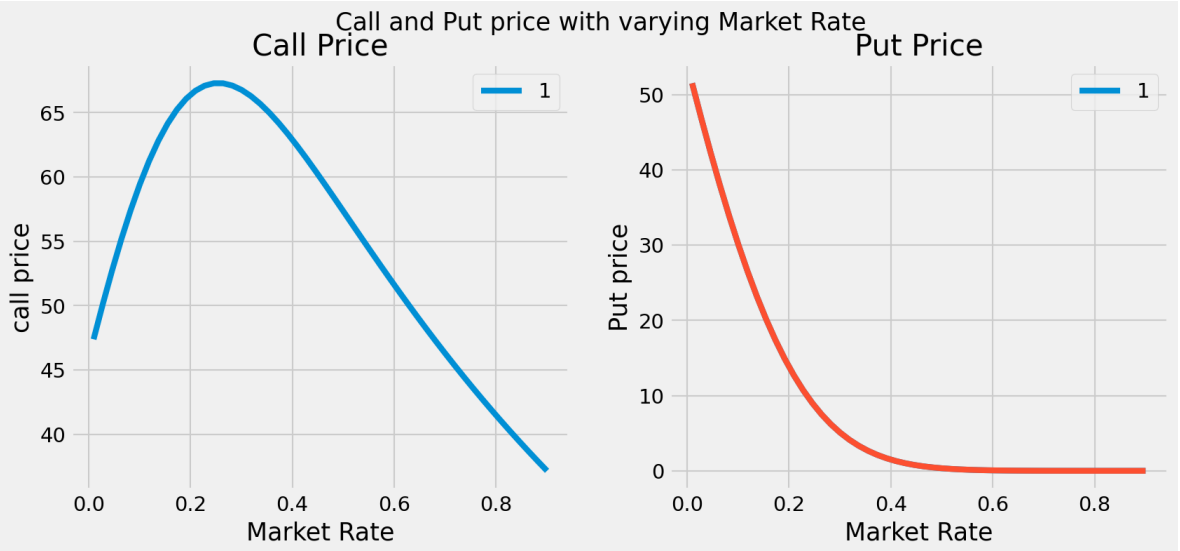
In [ ]:
```
def BinaryOptionCall(t, T, sigma, r, St, K, Q):
    d1=(np.log(St/K)+(r+(sigma*sigma)/2)*(T-t))/(sigma*np.sqrt(T-t))
    d2=d1-sigma*np.sqrt(T-t)
    return Q*np.exp(-r*(T-t))*norm.cdf(d2)

def BinaryOptionPut(t, T, sigma, r, St, K, Q):
    d1=(np.log(St/K)+(r+(sigma*sigma)/2)*(T-t))/(sigma*np.sqrt(T-t))
    d2=d1-sigma*np.sqrt(T-t)
    return Q*np.exp(-r*(T-t))*norm.cdf(-d2)
```

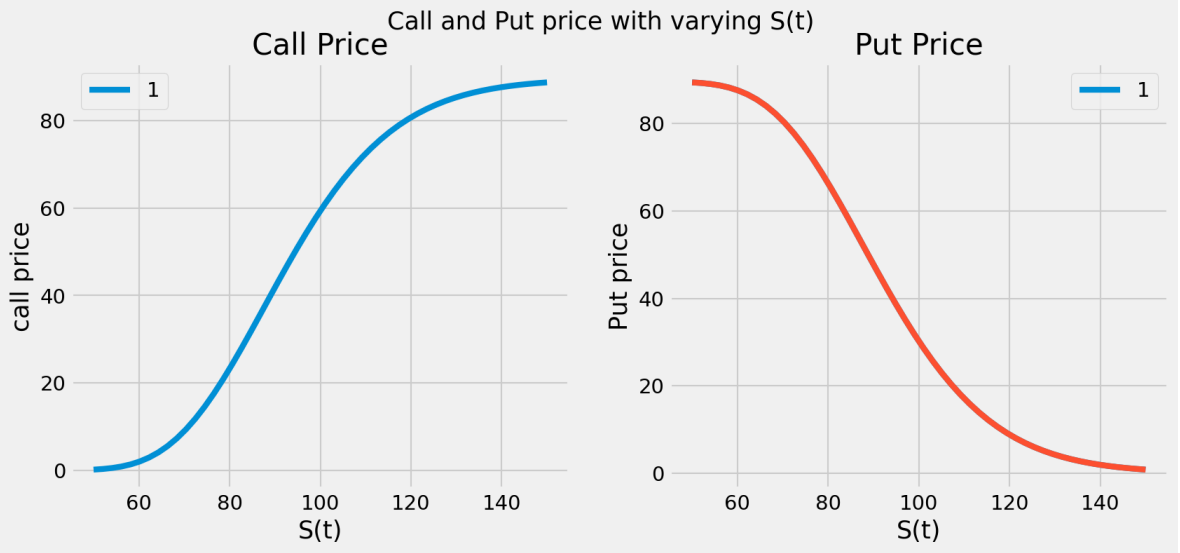
In [ ]:
```
# def optionPrice(t, T, sigma, r, St, K, Q):
varying = []
varying.append(np.linspace(0,1,50))
varying.append(np.linspace(1.1,2,50))
varying.append(np.linspace(0.01,0.90,50))
varying.append(np.linspace(0.01,0.90,50))
varying.append(np.linspace(50,150,50))
varying.append(np.linspace(50,150,50))
varying.append(np.linspace(50,150,50))

names = ["start time", "time Period", "Sigma", "Market Rate", "S(t)", "St
for i in range(1,6):
    fig, ax = plt.subplots(nrows = 1,ncols = 2)
    for st in [1]:
        call_p = []
        put_p = []
        params = [0, 1.1, 0.2, 0.1, 100, 100, 100]
        for ii in range(len(varying[i])):
            params[i]=varying[i][ii]
            call_p.append(BinaryOptionCall(params[0],params[1],params[2],
            put_p.append(BinaryOptionPut(params[0],params[1],params[2],pa
            fig.suptitle("Call and Put price with varying "+names[i])
        ax[0].plot(varying[i],call_p)
        ax[1].plot(varying[i],put_p)
    fig.set_size_inches(12, 5)
    fig.set_dpi(150)
    ax[0].set_title("Call Price")
    ax[0].set_xlabel(names[i])
    ax[0].set_ylabel("call price")
    ax[1].plot(varying[i],put_p)
    ax[1].set_title("Put Price")
    ax[1].set_xlabel(names[i])
    ax[1].set_ylabel("Put price")
    ax[0].legend([1])
    ax[1].legend([1])
    plt.show()

for i in range(7):
    for j in range(i+1,7):
        fig, ax = plt.subplots(nrows = 1,ncols = 2,subplot_kw={"projectio
        call_p = np.zeros((50,50))
        put_p = np.zeros((50,50))
        p_axis, q_axis = np.meshgrid(varying[i],varying[j])
        params = [0, 1.1, 0.2, 0.1, 100, 100, 100]
        for ii in range(len(p_axis)):
            for jj in range(len(q_axis)):
                params[i]=varying[i][ii]
                params[j]=varying[j][jj]
                call_p[ii][jj]=BinaryOptionCall(params[0],params[1],param
                put_p[ii][jj]=BinaryOptionPut(params[0],params[1],params[
        call_p=call_p.T
        put_p=put_p.T
        fig.suptitle("Call and Put price with varying "+names[i]+" and "+
        fig.set_size_inches(12, 5)
        fig.set_dpi(150)
        ax[0].plot_surface(p_axis,q_axis,call_p,cmap='inferno')
        ax[0].set_title("Call Price")
        ax[0].set_xlabel(names[i])
        ax[0].set_ylabel(names[j])
        ax[0].set_zlabel("call price")
```

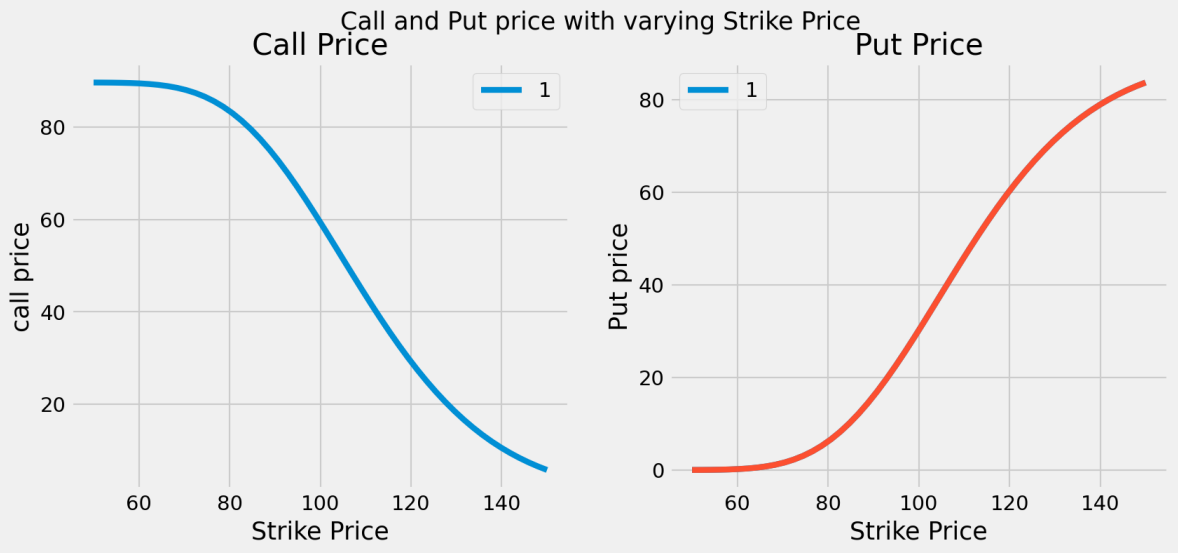
```
        ax[1].plot_surface(p_axis,q_axis,put_p,cmap='viridis')
        ax[1].set_title("Put Price")
        ax[1].set_xlabel(names[i])
        ax[1].set_ylabel(names[j])
        ax[1].set_zlabel("Put price")
        ax[0].view_init(15,-135)
        ax[1].view_init(15,45)
        plt.show()
```

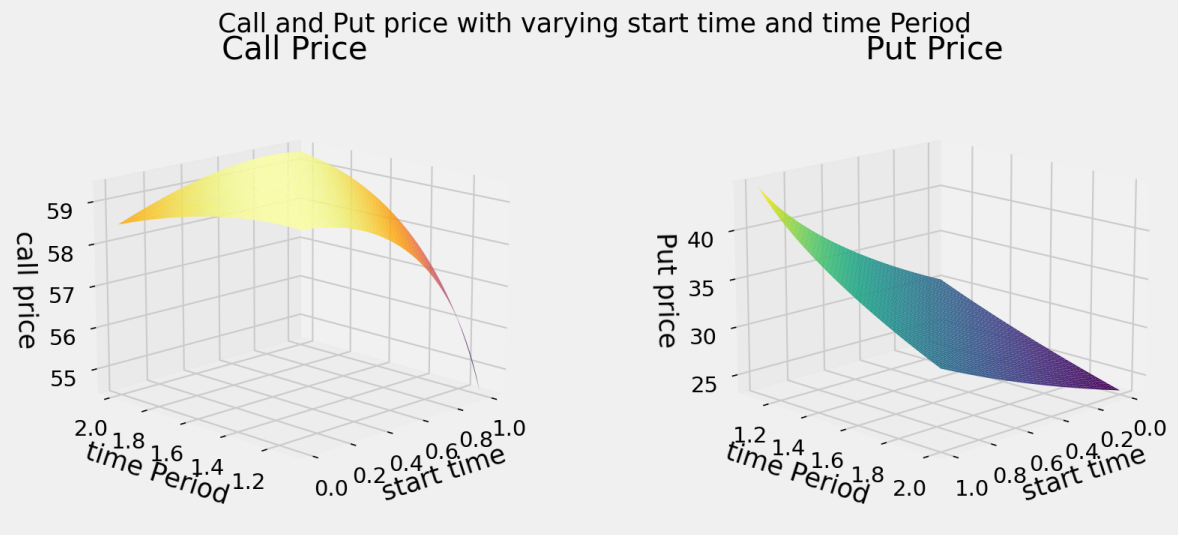


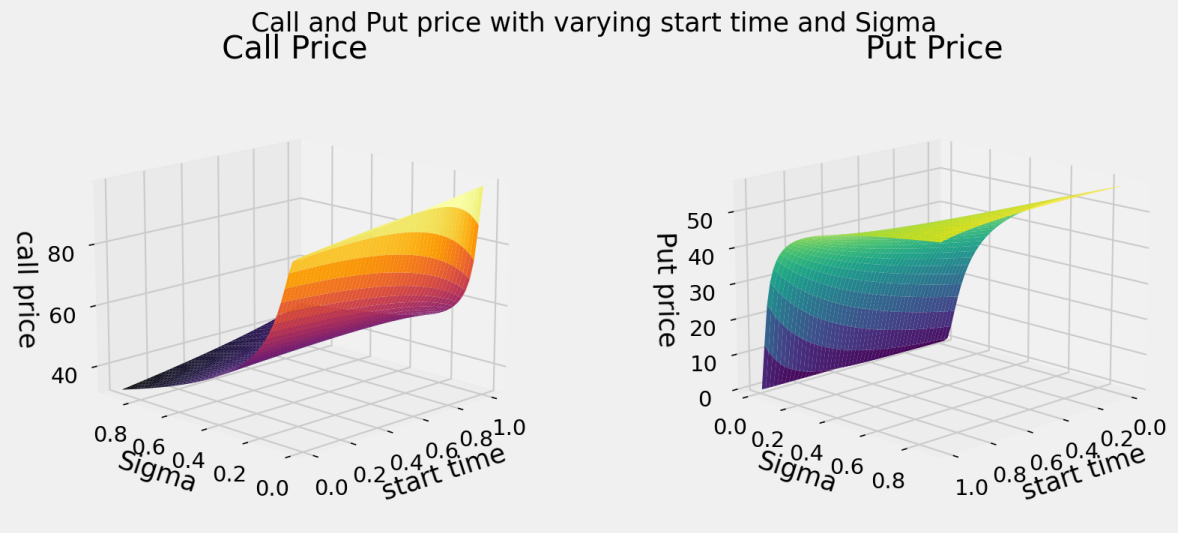


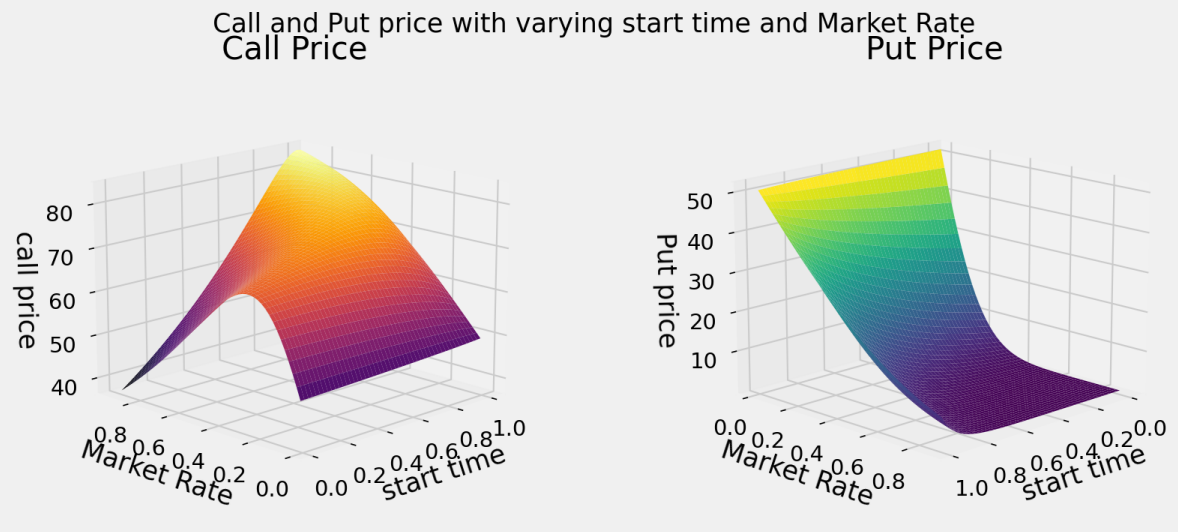


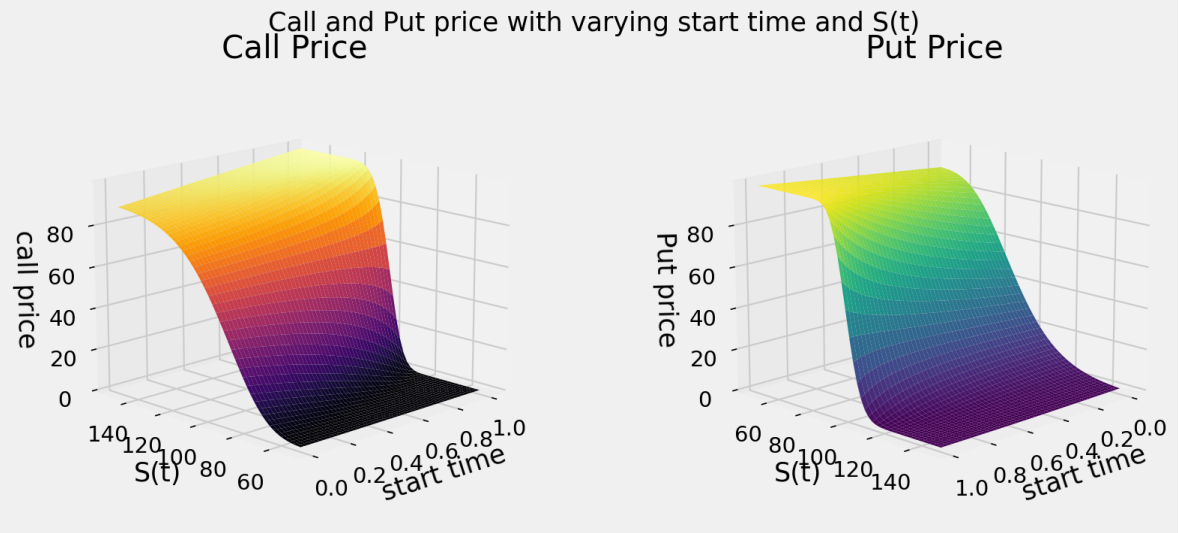

Call and Put price with varying S(t)
Call Price
Put Price
call price
Put price
S(t)
S(t)
1
1

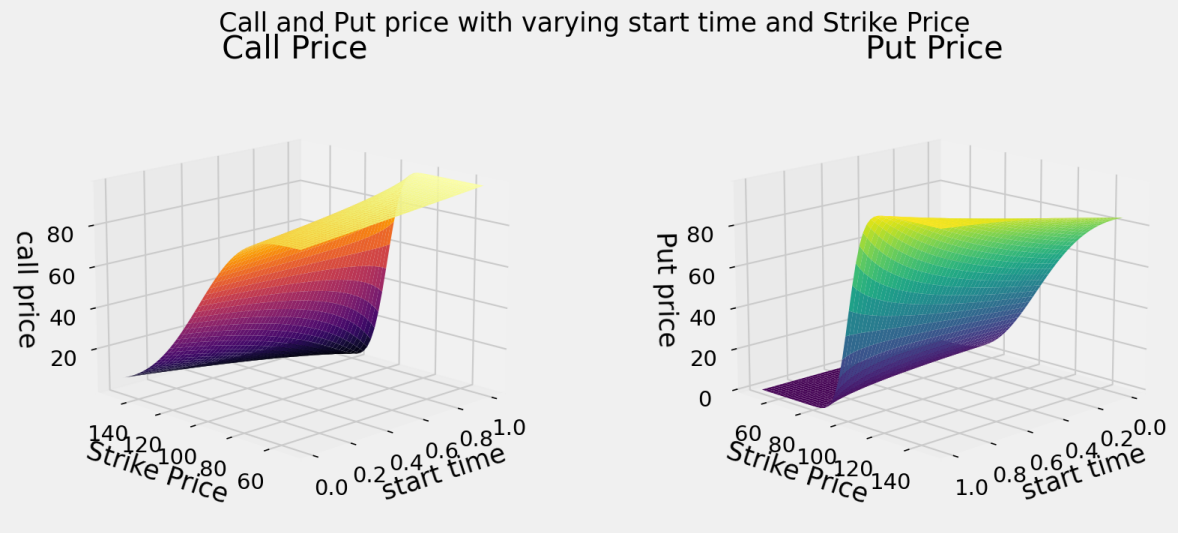

Call and Put price with varying Strike Price
Call Price
Put Price
call price
Put price
Strike Price
Strike Price
1
1

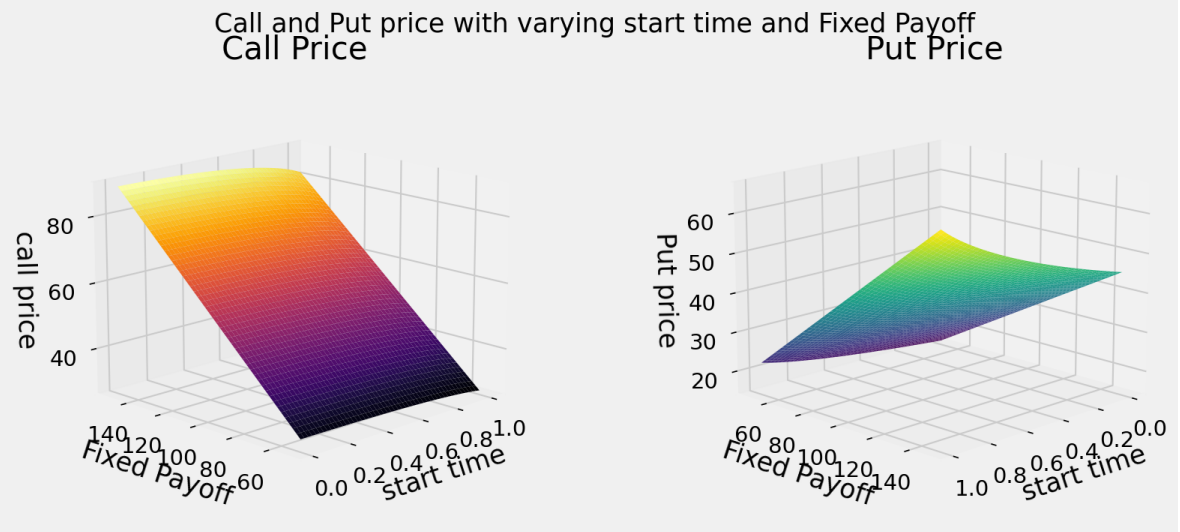

Call and Put price with varying start time and time Period
Call Price
Put Price
call price
Put price
time Period
start time
time Period
start time

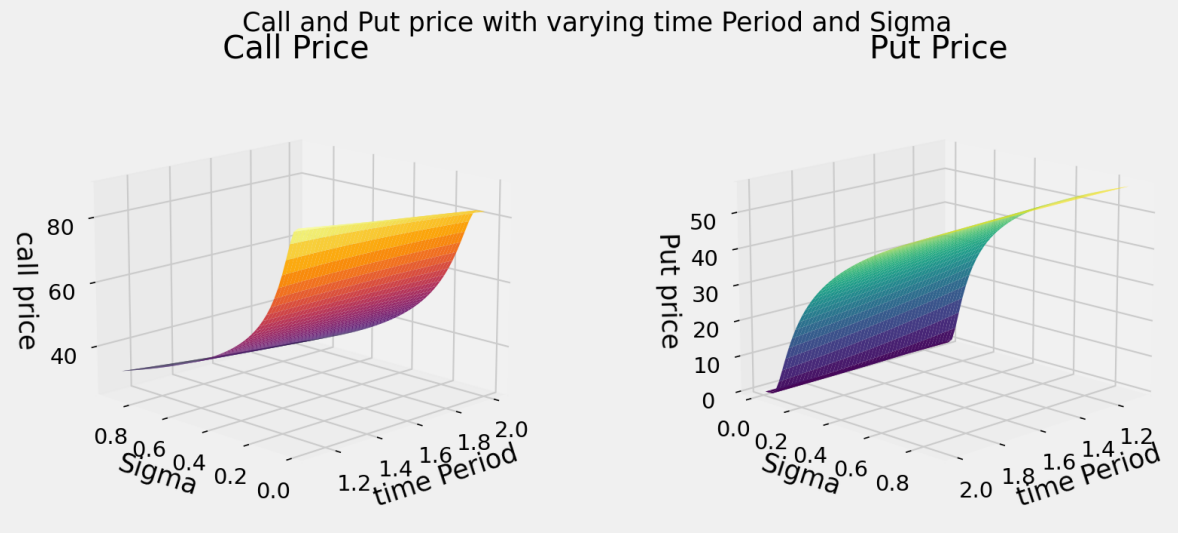

Call and Put price with varying start time and Sigma
Call Price
Put Price
call price
Put price
Sigma
start time

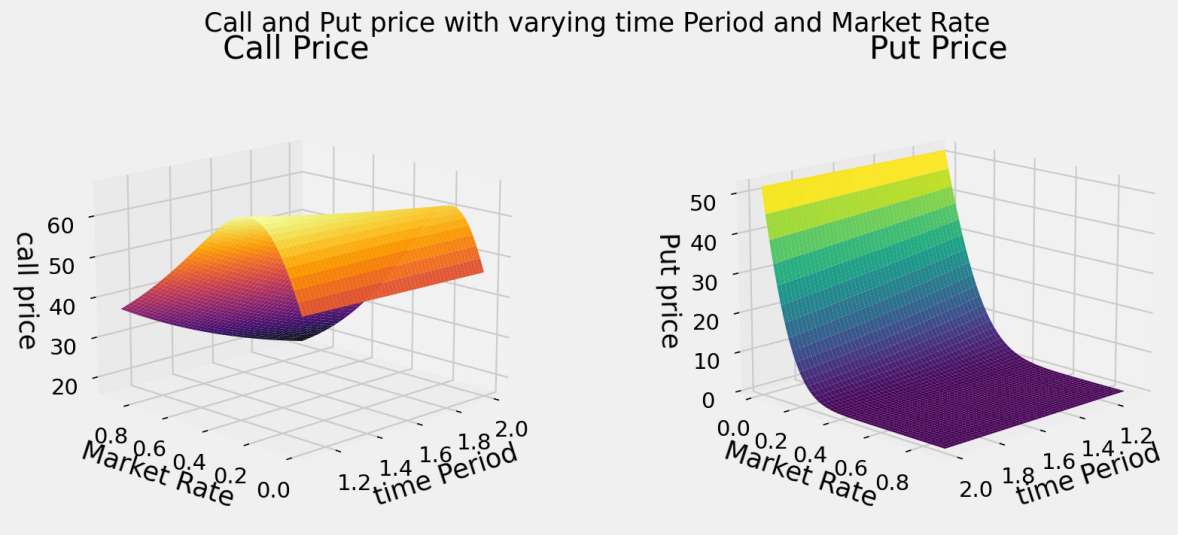

Call and Put price with varying start time and Market Rate
Call Price
Put Price
call price
Put price
Market Rate
start time

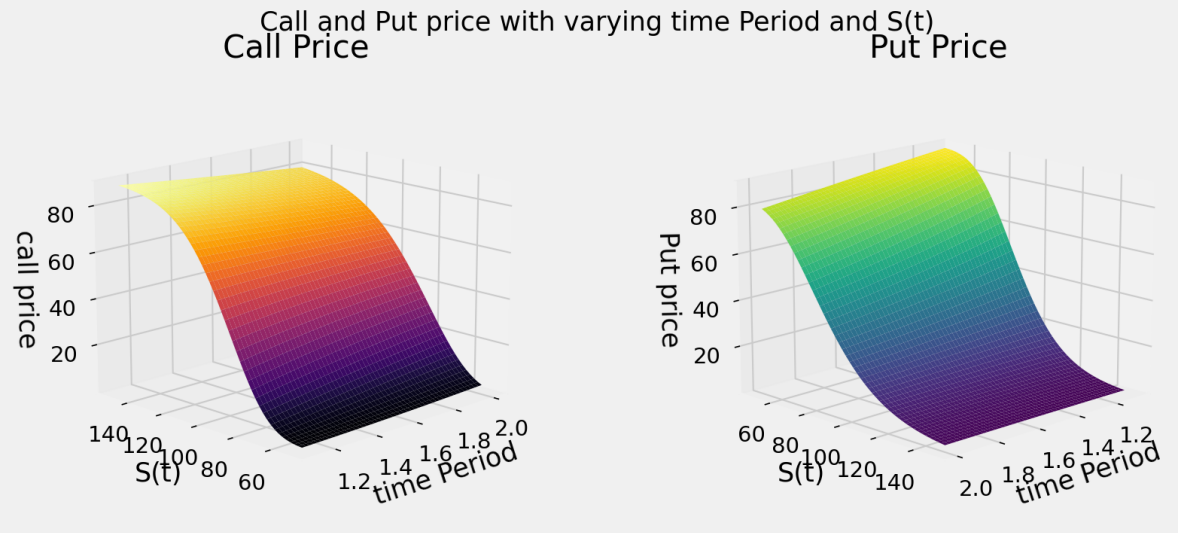

Call and Put price with varying start time and S(t)
Call Price
Put Price
call price
Put price
S(t)
start time

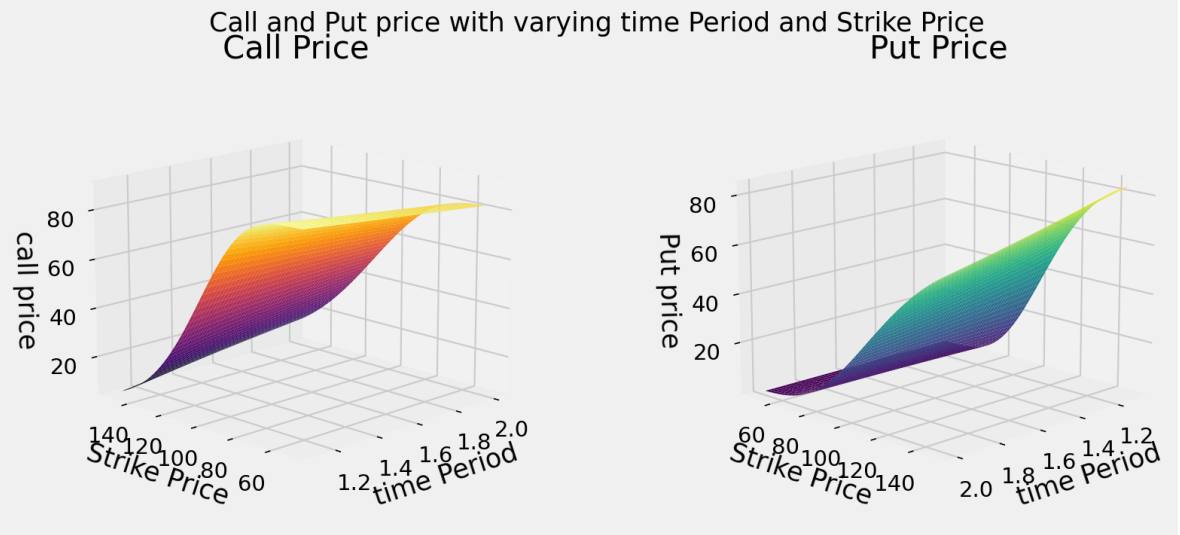

Call and Put price with varying start time and Strike Price
Call Price
Put Price
call price
Put price
Strike Price
start time

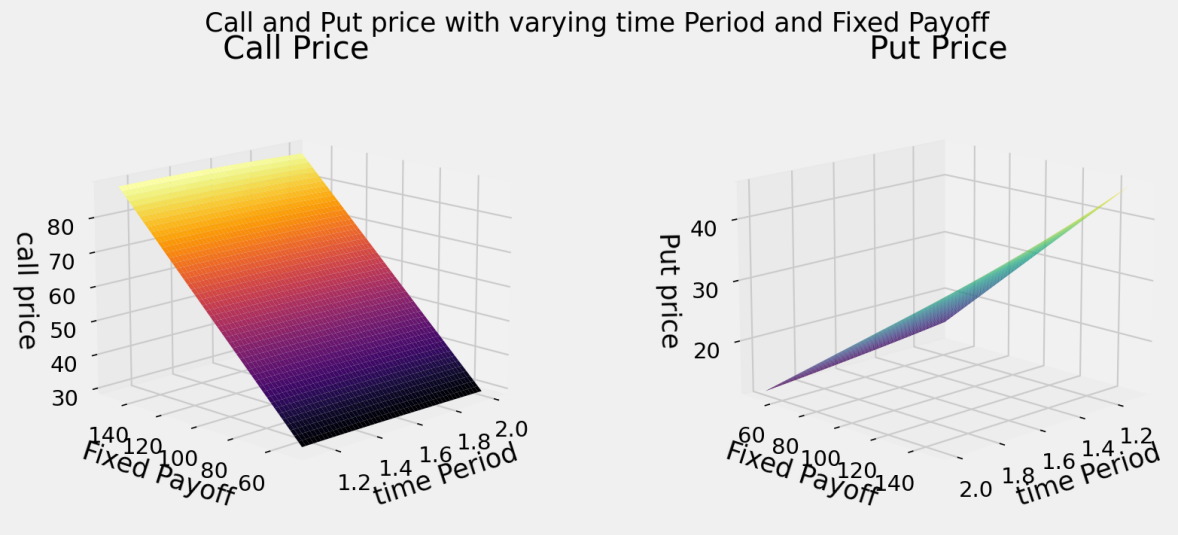

Call and Put price with varying start time and Fixed Payoff
Call Price
Put Price
call price
Put price
Fixed Payoff
start time

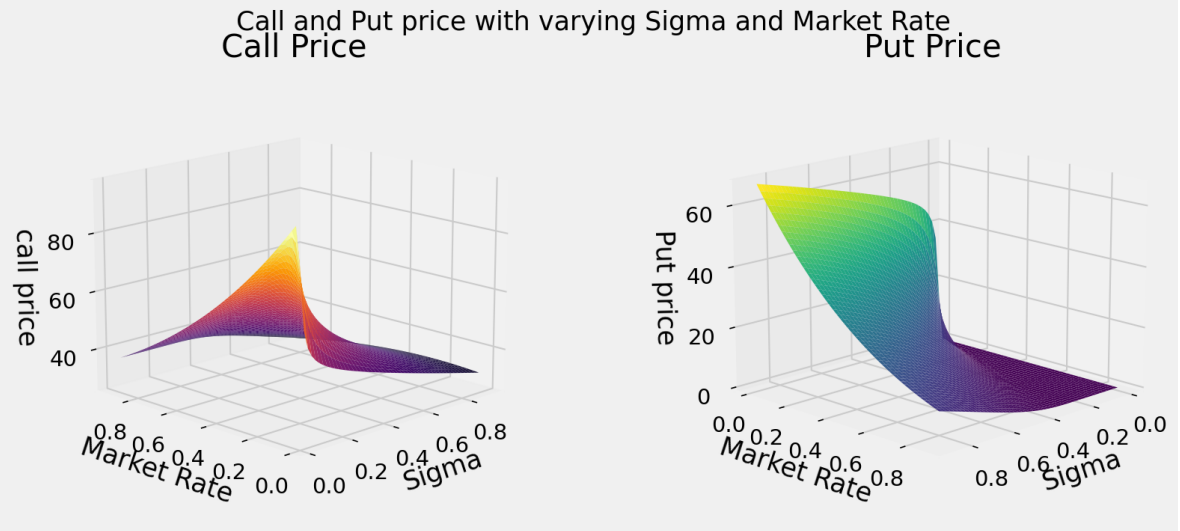

Call and Put price with varying time Period and Sigma
Call Price
Put Price
call price
Put price
Sigma
time Period

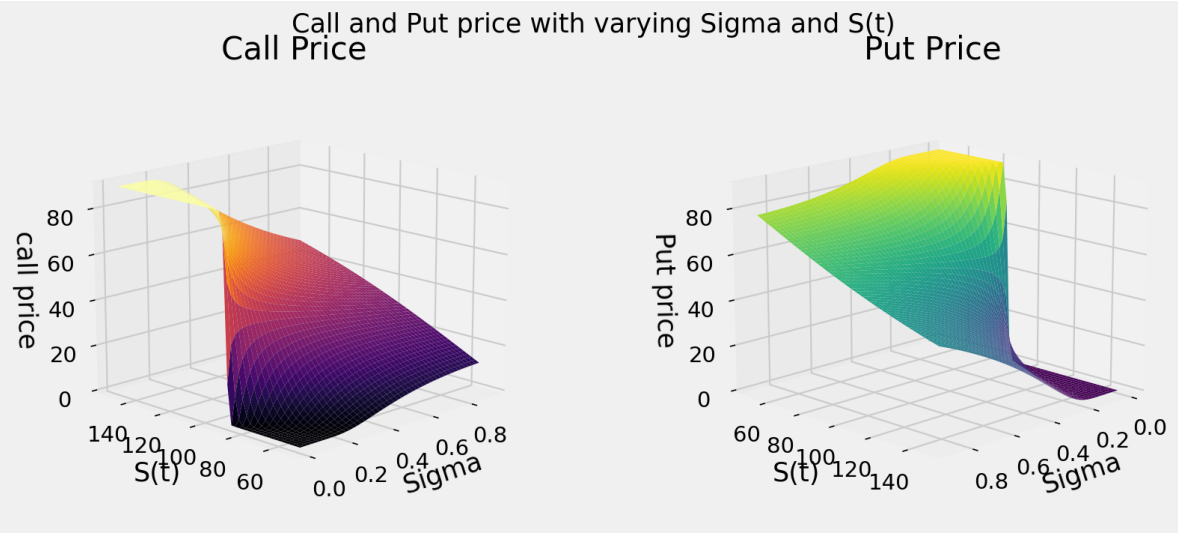

Call and Put price with varying time Period and Market Rate
Call Price
Put Price
call price
Put price
Market Rate
time Period

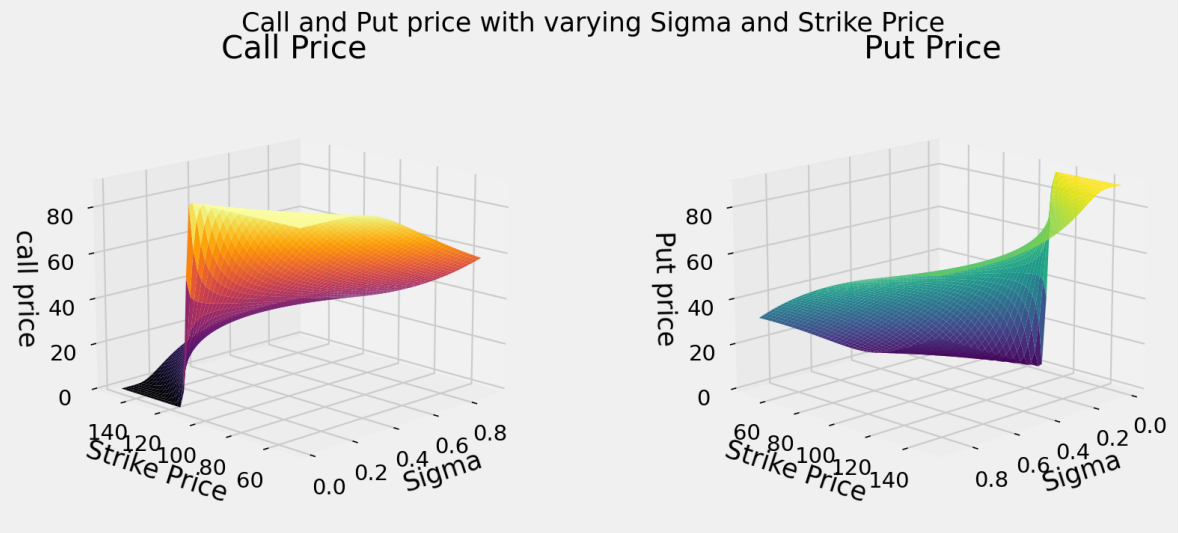

Call and Put price with varying time Period and S(t)
Call Price
Put Price
call price
Put price
S(t)
time Period

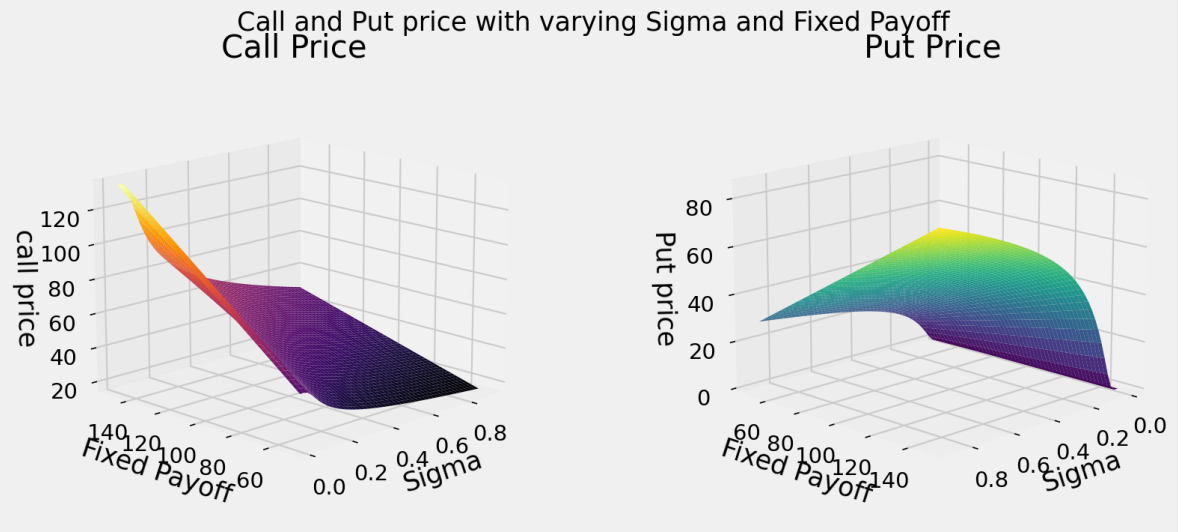

Call and Put price with varying time Period and Strike Price
Call Price
Put Price
call price
Put price
Strike Price
time Period

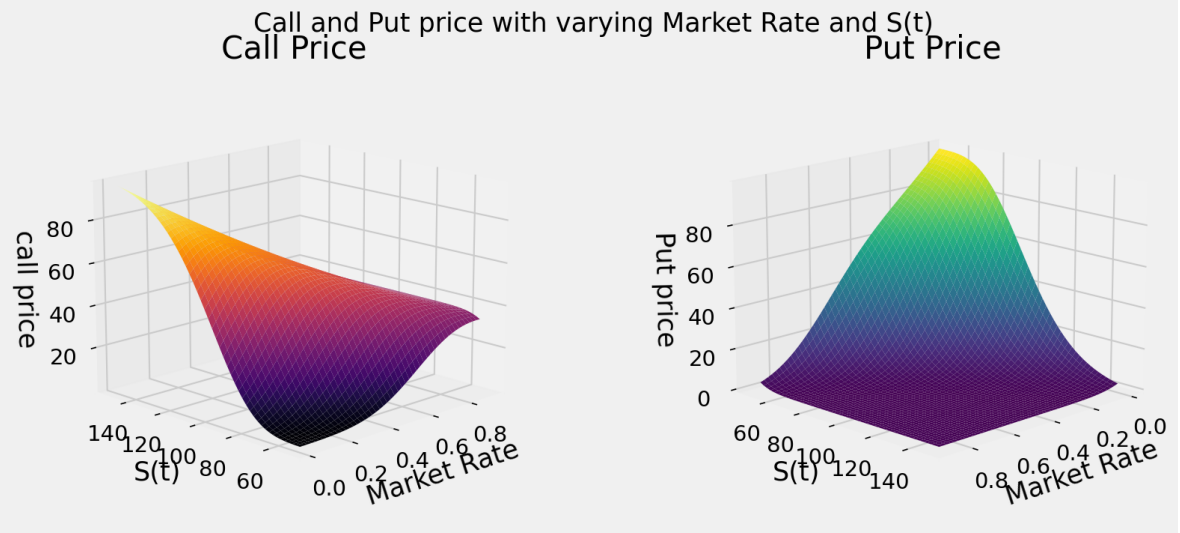

Call and Put price with varying time Period and Fixed Payoff
Call Price
Put Price
call price
Fixed Payoff
time Period
Put price
Fixed Payoff
time Period

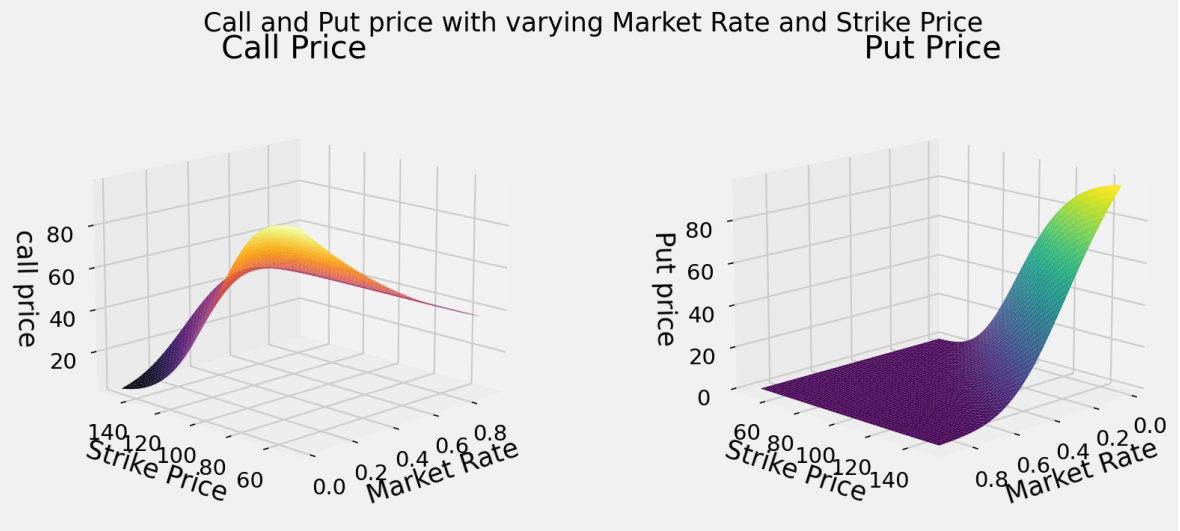

Call and Put price with varying Sigma and Market Rate
Call Price
Put Price
call price
Market Rate
Sigma
Put price
Market Rate
Sigma


Call and Put price with varying Sigma and S(t)
Call Price
Put Price
call price
S(t)
Sigma
Put price
S(t)
Sigma


Call and Put price with varying Sigma and Strike Price
Call Price
Put Price
call price
Put price
Strike Price
Sigma

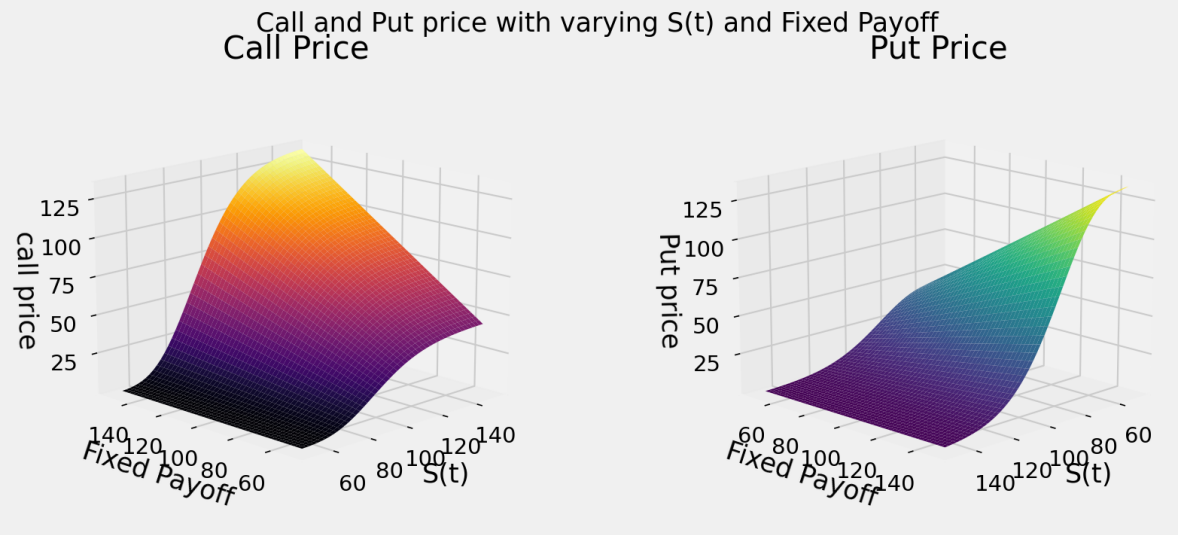

Call and Put price with varying Sigma and Fixed Payoff
Call Price
Put Price
call price
Put price
Fixed Payoff
Sigma


Call and Put price with varying Market Rate and S(t)
Call Price
Put Price
call price
Put price
S(t)
Market Rate


Call and Put price with varying Market Rate and Strike Price
Call Price
Put Price
call price
Put price
Strike Price
Market Rate


Call and Put price with varying Market Rate and Fixed Payoff
Call Price
Put Price
call price
Put price
Fixed Payoff
Market Rate


Call and Put price with varying S(t) and Strike Price
Call Price
Put Price
call price
Put price
Strike Price
S(t)


Call and Put price with varying S(t) and Fixed Payoff
Call Price
Put Price
call price
Put price
125
100
75
50
25
140
120
100
80
60
Fixed Payoff
S(t)


Call and Put price with varying Strike Price and Fixed Payoff
Call Price
Put Price
call price
Put price
125
100
75
50
25
140
120
100
80
60
Fixed Payoff
Strike Price

Gap Option

A gap option is a type of financial derivative where the payoff depends on the difference between the price of the underlying asset and a predetermined strike price. Specifically, a gap call option pays out if the price of the underlying asset at maturity is above the strike price by more than a certain amount, called the "gap." Conversely, a gap put option pays out if the price of the underlying asset at maturity is below the strike price by more than the gap.

The formula for the payout of a gap call option is:

$$\max(S_T - K_2, 0)$$

where S_T is the price of the underlying asset at maturity, and K_2 is the strike price plus the gap. If the difference between S_T and K_2 is less than zero, the option expires worthless.

The formula for the payout of a gap put option is:

$$\max(K_2 - S_T, 0)$$

where the variables have the same meaning as in the gap call option formula.

Gap Option Pricing Formulae

The pricing of gap options can be calculated using numerical methods, such as Monte Carlo simulations, or analytic methods, such as the Black-Scholes model with a modification to account for the gap.

The formula for pricing a gap call option using the Black-Scholes model is:

$$C_{gap} = S_0 N(d_1) - K_2 e^{-rT} N(d_2)$$

where S_0 is the current price of the underlying asset, K_2 is the strike price plus the gap, r is the risk-free interest rate, T is the time to maturity, σ is the volatility of the underlying asset, $N(\cdot)$ is the cumulative distribution function of the standard normal distribution, $\phi(\cdot)$ is the probability density function of the standard normal distribution, and

$$d_1 = \frac{\ln(S_0/K_2) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

The formula for pricing a gap put option is similar, but with some changes to account for the put option payoff:

$$P_{gap} = K_2 e^{-rT} N(-d_2) - S_0 N(-d_1)$$

where P_{gap} is the price of the gap put option and the other variables have the same meanings as in the gap call option formula.

Note that the Black-Scholes model assumes that the underlying asset follows a log-normal distribution, which may not always hold in practice. Additionally, the model assumes that the risk-free interest rate and volatility are constant over the life of the option, which may not always be the case. As such, the pricing of gap options using the Black-Scholes model may have limitations and may not always accurately reflect market prices.I wil now prove this pricing formulae:-

$$P = e^{-rT}(S_0 e^{(b-r-\frac{1}{2}\sigma^2)T} N(d_1') - Ke^{-rT}N(d_1) - Ge^{-rT}N(d_2))$$

To derive this formula, we first note that the gap option has a fixed payoff if the underlying asset price is above the strike price plus the gap level, and pays nothing if the underlying asset price is below the strike price. The payoff for a gap option can be represented as follows:

$$Q = \begin{cases} S_T - K - G, & \text{if } S_T > K + G \\ 0, & \text{otherwise} \end{cases}$$

where Q is the payoff, S_T is the price of the underlying asset at expiry, K is the strike price, and G is the gap level.

The price of a gap option can be calculated using the Black-Scholes model with an adjustment for the gap level.

Using the risk-neutral valuation principle, we can assume that the expected payoff of the option is equal to its present value. Therefore, we can write:

$$P = e^{-rT}E[Q]$$

To calculate $E[Q]$, we need to derive the distribution of S_T. Using the Black-Scholes model with an adjustment for the gap level, we can derive the following expressions for d_1' and d_2:

$$d_1' = \frac{\ln(S_0/(K+G)) + (b + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}$$

and

$$d_2 = d_1' - \sigma\sqrt{T}$$

Substituting these expressions into the equation for the expected payoff, we get:

$$P = e^{-rT}E[Q]$$

$$= e^{-rT}[(S_0 - K - G)N(d_1')e^{-(b-r)T} - (S_0 - K)e^{-rT}N(d_1) - Ge^{-rT}N(d_2)]$$

$$= e^{-rT}(S_0 e^{(b-r-\frac{1}{2}\sigma^2)T} N(d_1') - Ke^{-rT}N(d_1) - Ge^{-rT}N(d_2))$$

which is the formula for pricing a gap option with strike K and gap level G.

Below is the sensitivity analysis of Binary Option along with all the relevant plots:-

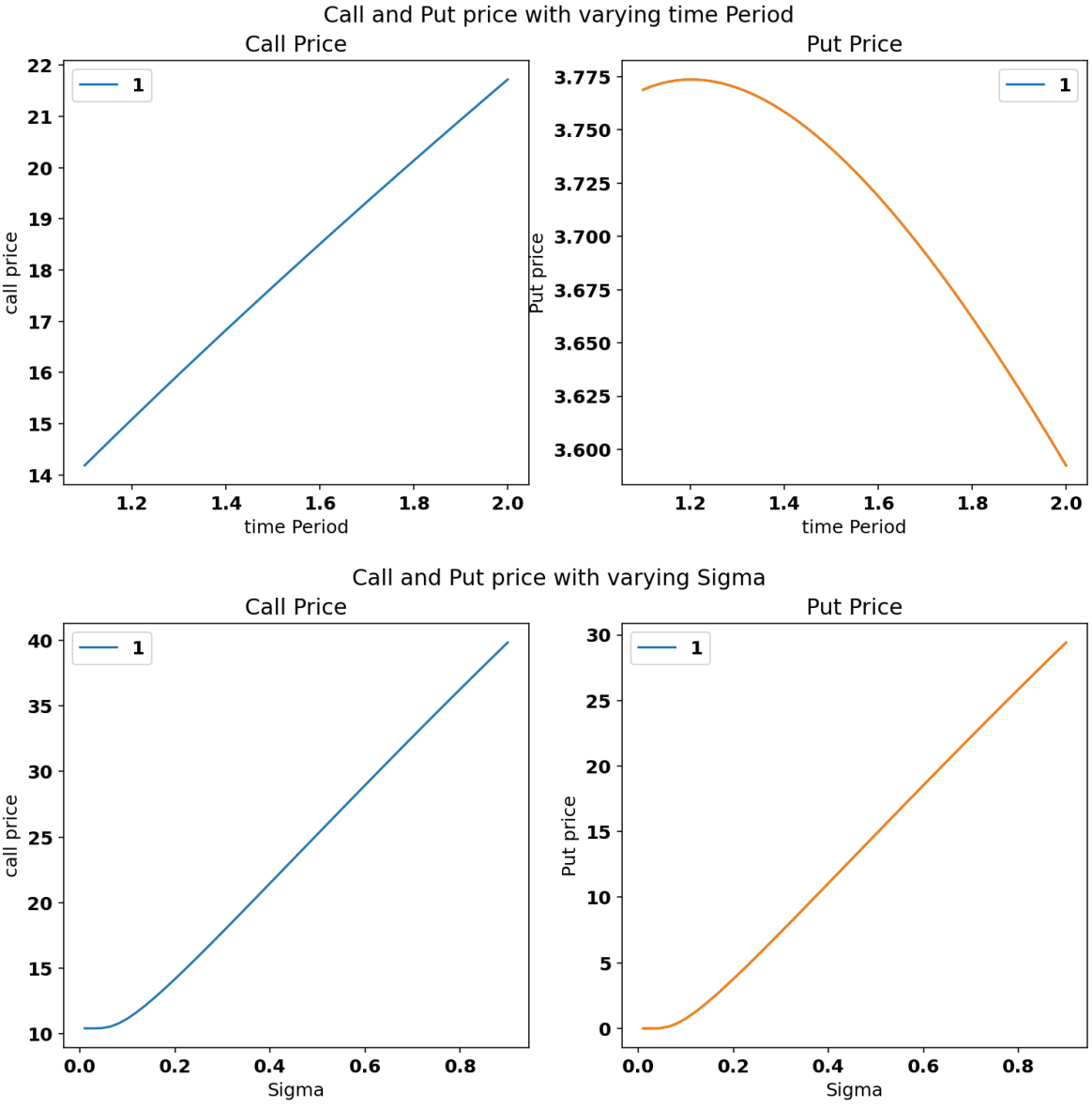
In [ ]:
```
import numpy as np
```

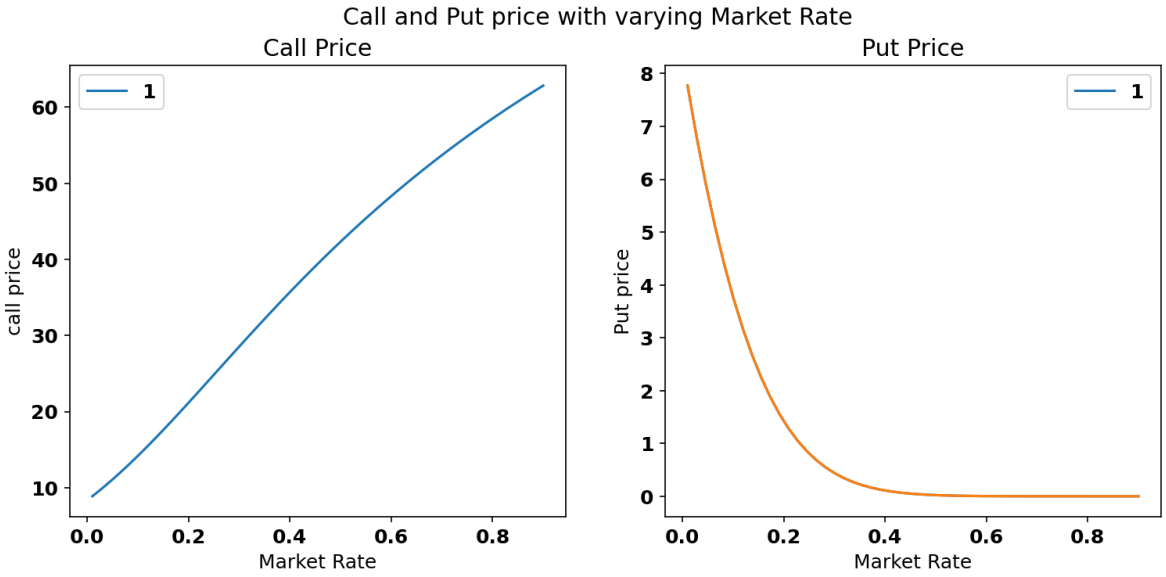
In [ ]:
```
def gapOptionCall(t, T, sigma, r, St, K, Q):
    d1=(np.log(St/K)+(r+(sigma*sigma)/2)*(T-t))/(sigma*np.sqrt(T-t))
    d2=d1-sigma*np.sqrt(T-t)
    return St*norm.cdf(d1) - K*math.exp(-r*(T-t))*norm.cdf(d2)

def gapOptionPut(t, T, sigma, r, St, K, Q):
    d1=(np.log(St/K)+(r+(sigma*sigma)/2)*(T-t))/(sigma*np.sqrt(T-t))
    d2=d1-sigma*np.sqrt(T-t)
    return -St*norm.cdf(-d1) +K*math.exp(-r*(T-t))*norm.cdf(-d2)
```

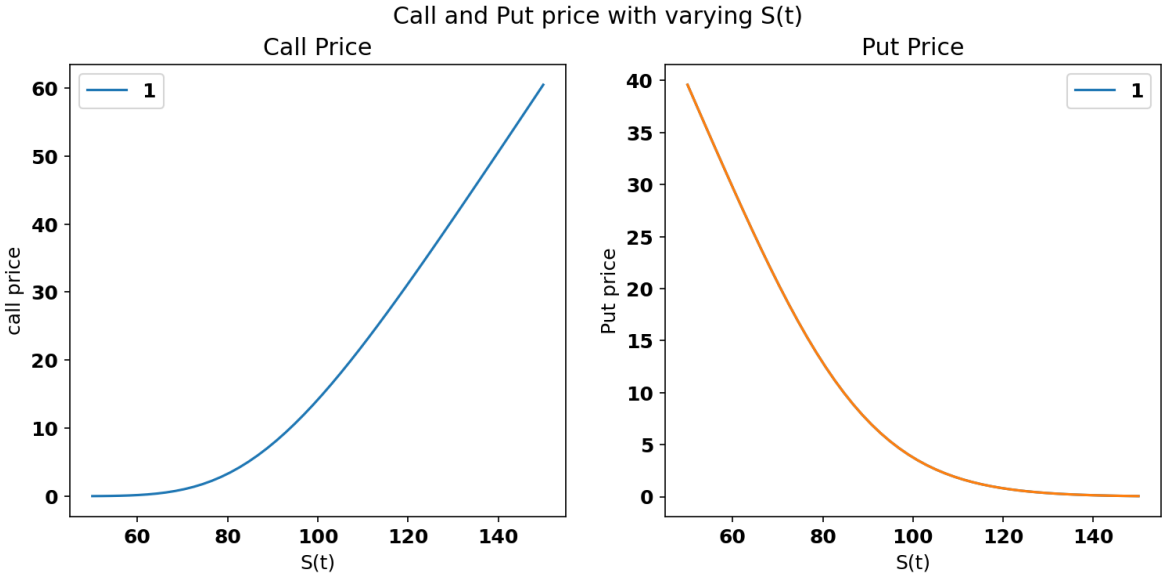
In [ ]:
```
# def optionPrice(t, T, sigma, r, St, K, Q):
varying = []
varying.append(np.linspace(0,1,50))
varying.append(np.linspace(1.1,2,50))
varying.append(np.linspace(0.01,0.90,50))
varying.append(np.linspace(0.01,0.90,50))
varying.append(np.linspace(50,150,50))
varying.append(np.linspace(50,150,50))
varying.append(np.linspace(50,150,50))

names = ["start time", "time Period", "Sigma", "Market Rate", "S(t)", "Th
for i in range(1,6):
    fig, ax = plt.subplots(nrows = 1,ncols = 2)
    for st in [1]:
        call_p = []
        put_p = []
        params = [0, 1.1, 0.2, 0.1, 100, 100, 100]
        for ii in range(len(varying[i])):
            params[i]=varying[i][ii]
            call_p.append(gapOptionCall(params[0],params[1],params[2],par
            put_p.append(gapOptionPut(params[0],params[1],params[2],param
            fig.suptitle("Call and Put price with varying "+names[i])
        ax[0].plot(varying[i],call_p)
        ax[1].plot(varying[i],put_p)
    fig.set_size_inches(12, 5)
    fig.set_dpi(150)
    ax[0].set_title("Call Price")
    ax[0].set_xlabel(names[i])
    ax[0].set_ylabel("call price")
    ax[1].plot(varying[i],put_p)
    ax[1].set_title("Put Price")
    ax[1].set_xlabel(names[i])
    ax[1].set_ylabel("Put price")
    ax[0].legend([1])
    ax[1].legend([1])
    plt.show()

for i in range(7):
    for j in range(i+1,7):
        fig, ax = plt.subplots(nrows = 1,ncols = 2,subplot_kw={"projectio
        call_p = np.zeros((50,50))
        put_p = np.zeros((50,50))
        p_axis, q_axis = np.meshgrid(varying[i],varying[j])
        params = [0, 1.1, 0.2, 0.1, 100, 100, 100]
        for ii in range(len(p_axis)):
            for jj in range(len(q_axis)):
                params[i]=varying[i][ii]
                params[j]=varying[j][jj]
                call_p[ii][jj]=gapOptionCall(params[0],params[1],params[2
                put_p[ii][jj]=gapOptionPut(params[0],params[1],params[2],
        call_p=call_p.T
        put_p=put_p.T
        fig.suptitle("Call and Put price with varying "+names[i]+" and "+
        fig.set_size_inches(12, 5)
        fig.set_dpi(150)
        ax[0].plot_surface(p_axis,q_axis,call_p,cmap='inferno')
        ax[0].set_title("Call Price")
        ax[0].set_xlabel(names[i])
        ax[0].set_ylabel(names[j])
        ax[0].set_zlabel("call price")
```

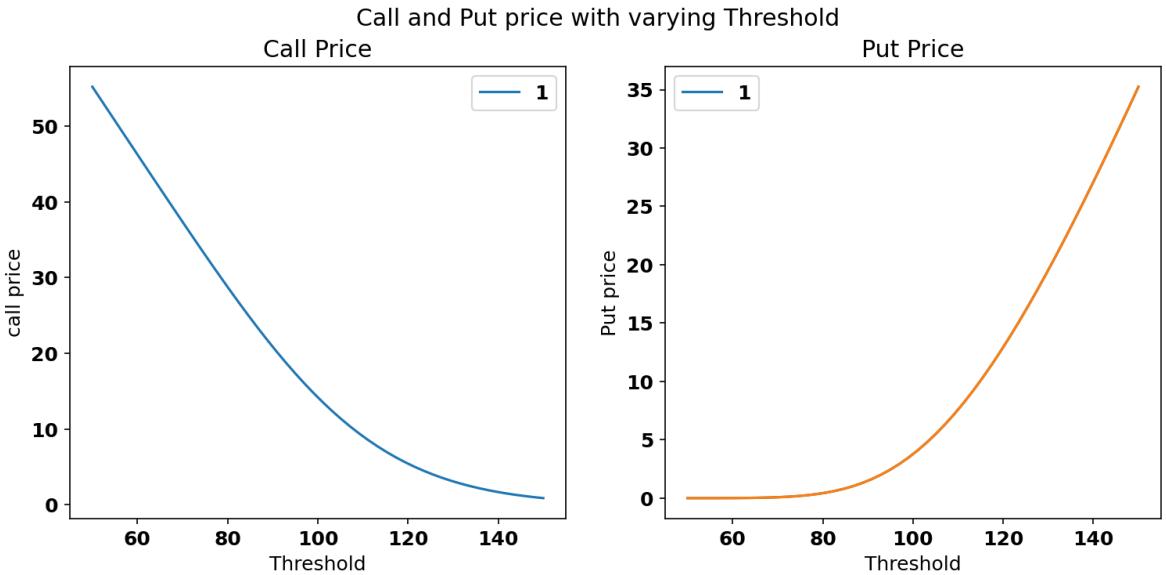
```
        ax[1].plot_surface(p_axis,q_axis,put_p,cmap='viridis')
        ax[1].set_title("Put Price")
        ax[1].set_xlabel(names[i])
        ax[1].set_ylabel(names[j])
        ax[1].set_zlabel("Put price")
        ax[0].view_init(15,-135)
        ax[1].view_init(15,45)
        plt.show()
```

```
findfont: Font family ['normal'] not found. Falling back to DejaVu Sans.
findfont: Font family ['normal'] not found. Falling back to DejaVu Sans.
findfont: Font family ['normal'] not found. Falling back to DejaVu Sans.
```



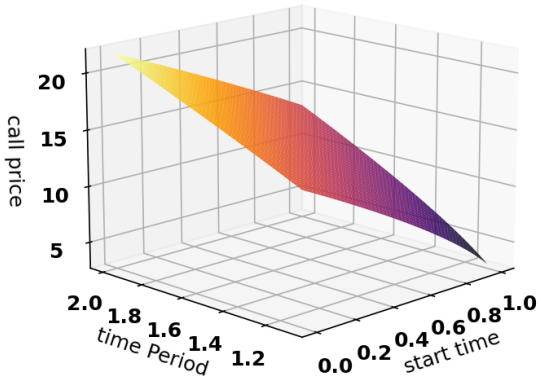

Call and Put price with varying Market Rate
Call Price
call price
Market Rate
1
Put Price
Put price
Market Rate
1


Call and Put price with varying S(t)
Call Price
call price
S(t)
1
Put Price
Put price
S(t)
1

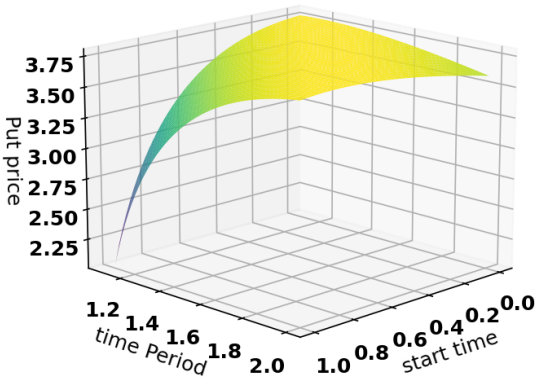

Call and Put price with varying Threshold
Call Price
call price
Threshold
1
Put Price
Put price
Threshold
1

Call and Put price with varying start time and time Period

Call Price

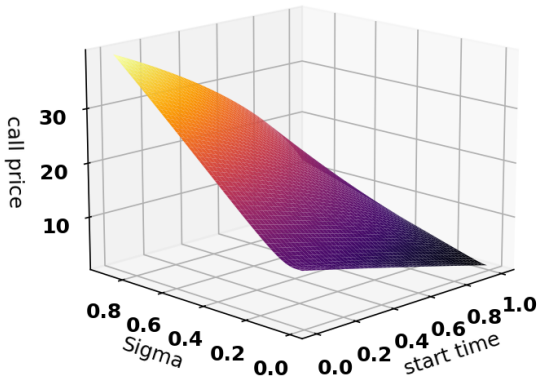


Put Price

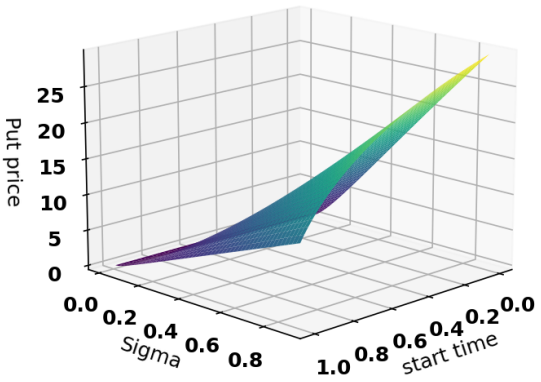


Call and Put price with varying start time and Sigma

Call Price

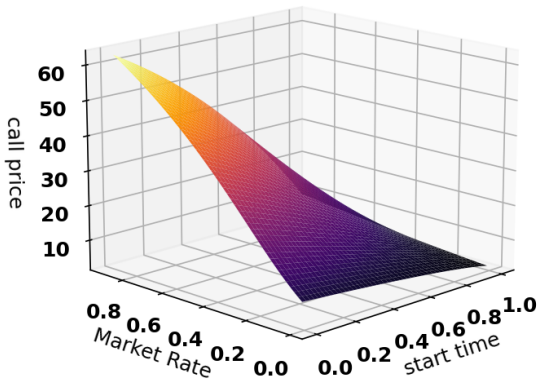


Put Price

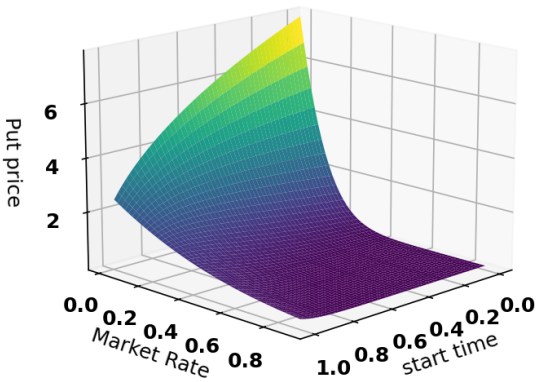


Call and Put price with varying start time and Market Rate

Call Price

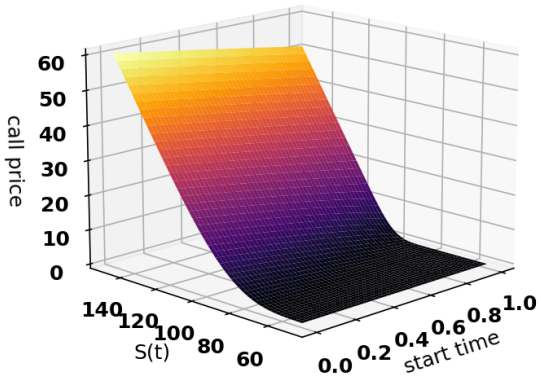


Put Price

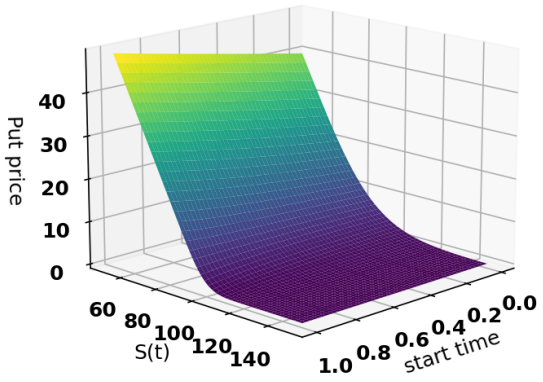


Call and Put price with varying start time and S(t)

Call Price

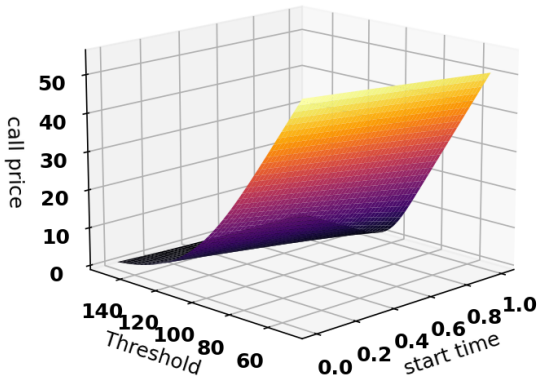


Put Price

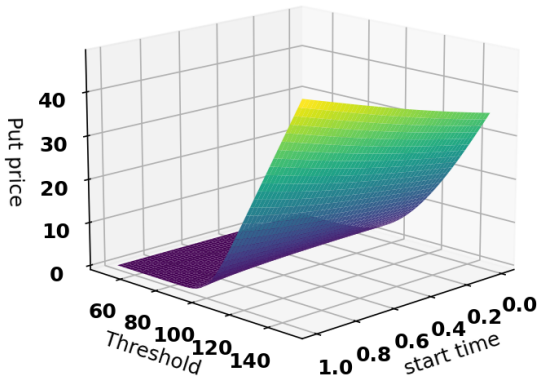


Call and Put price with varying start time and Threshold

Call Price

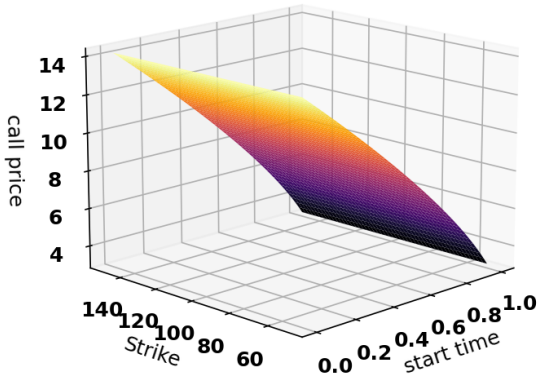


Put Price

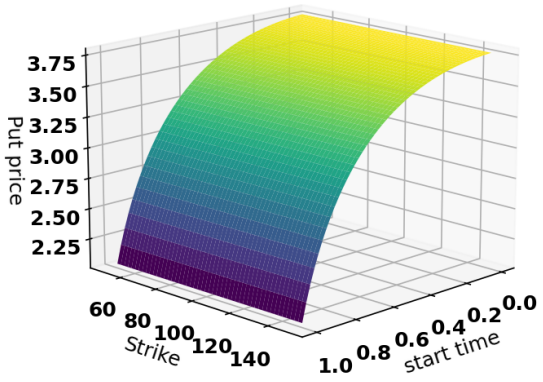


Call and Put price with varying start time and Strike

Call Price

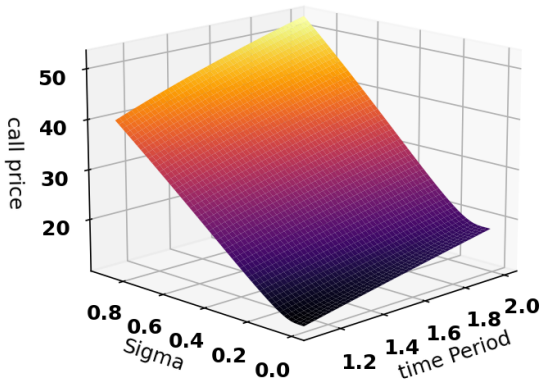


Put Price

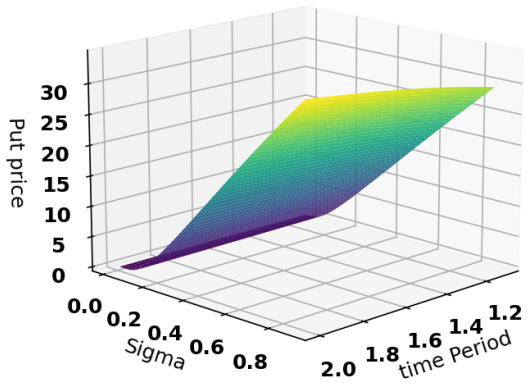


Call and Put price with varying time Period and Sigma

Call Price

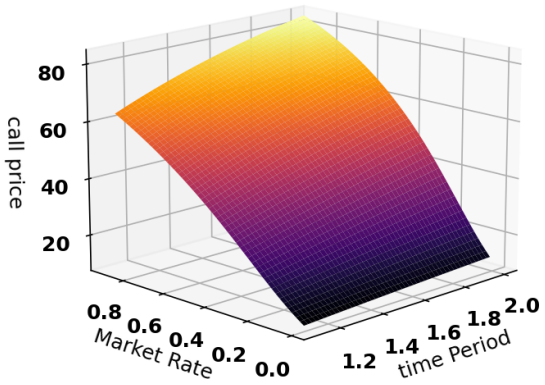


Put Price

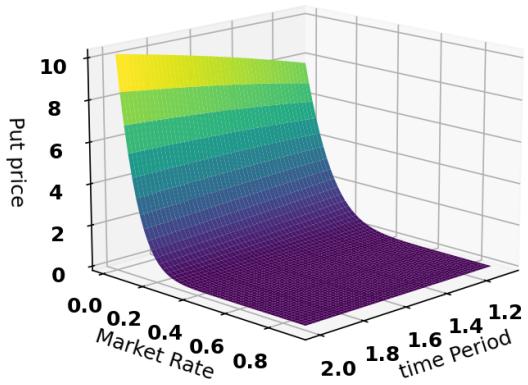


Call and Put price with varying time Period and Market Rate

Call Price

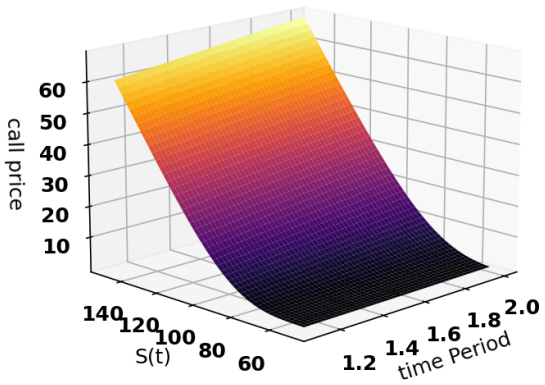


Put Price

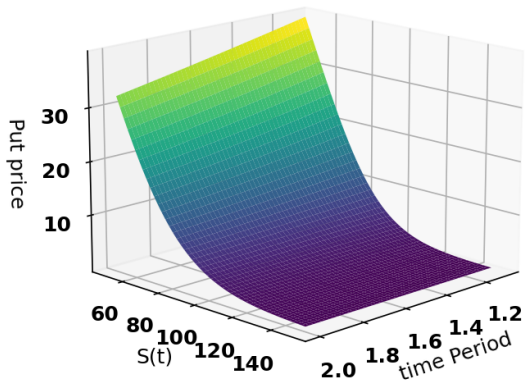


Call and Put price with varying time Period and S(t)

Call Price

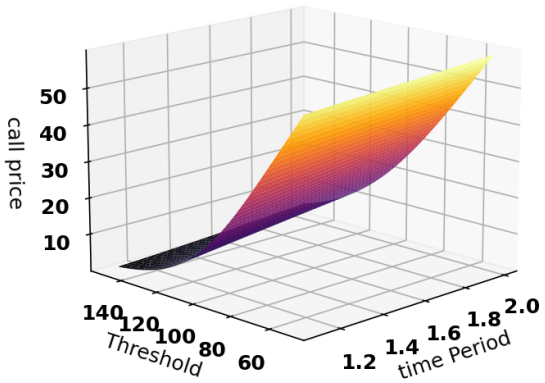


Put Price

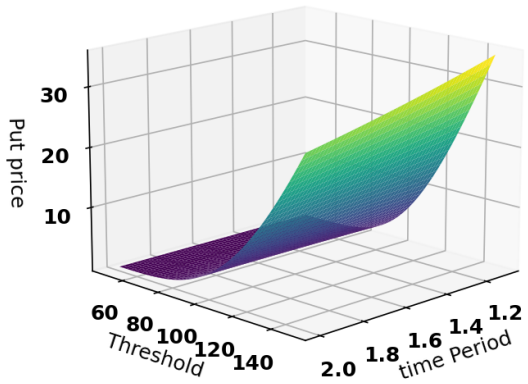


Call and Put price with varying time Period and Threshold

Call Price

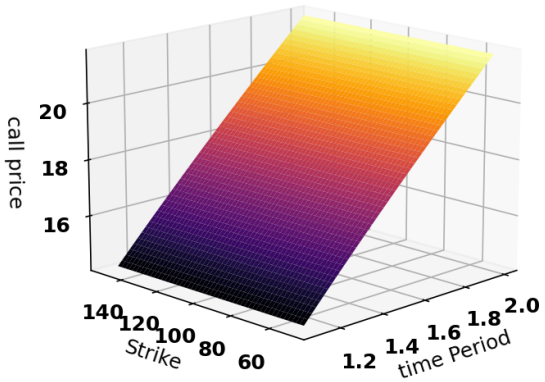


Put Price

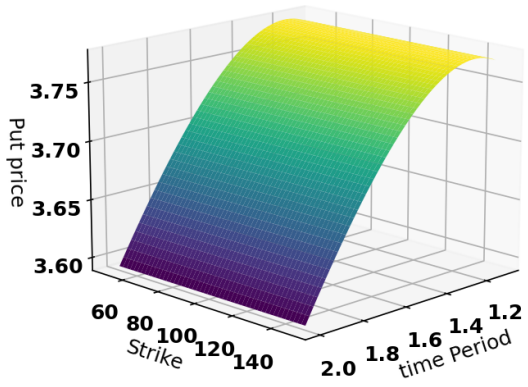


Call and Put price with varying time Period and Strike

Call Price

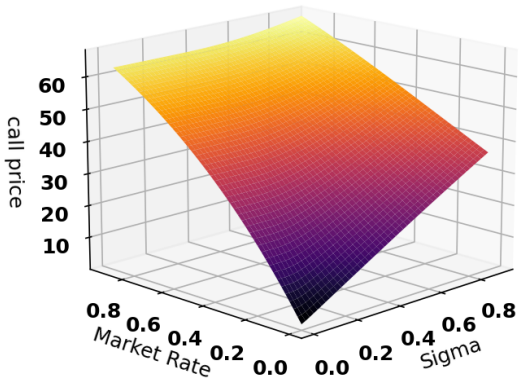


Put Price

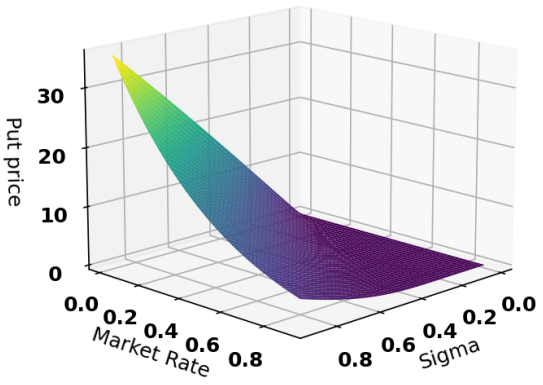


Call and Put price with varying Sigma and Market Rate

Call Price

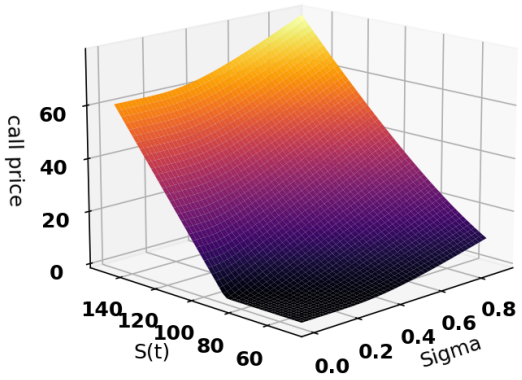


Put Price

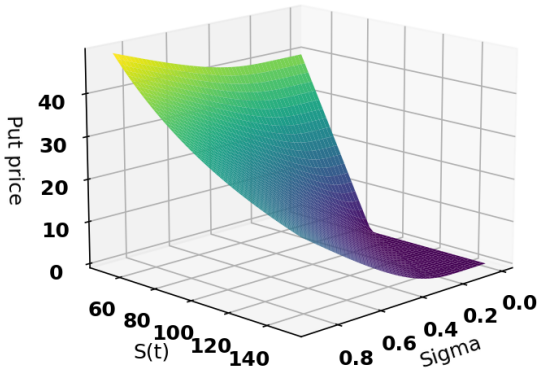


Call and Put price with varying Sigma and S(t)

Call Price

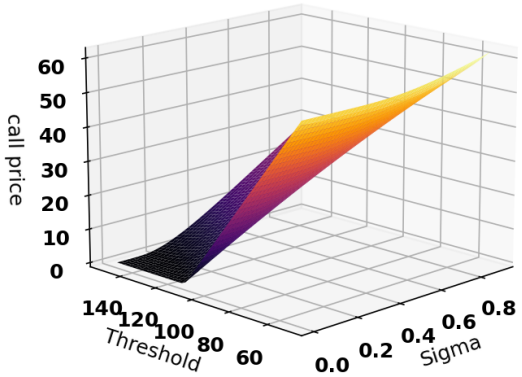


Put Price

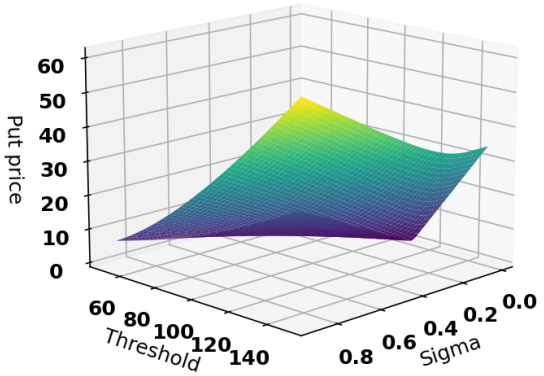


Call and Put price with varying Sigma and Threshold

Call Price

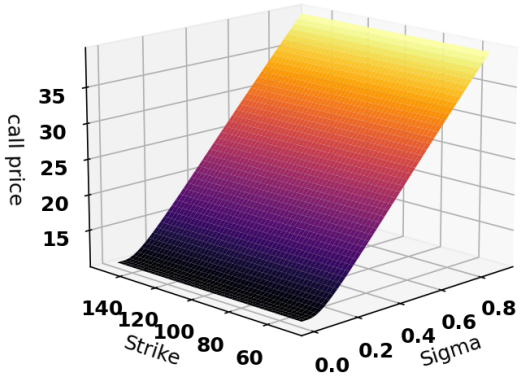


Put Price

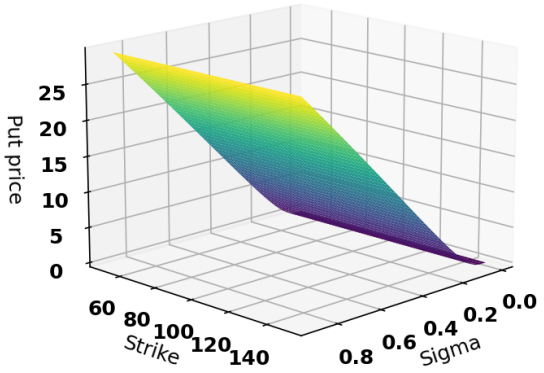


Call and Put price with varying Sigma and Strike

Call Price

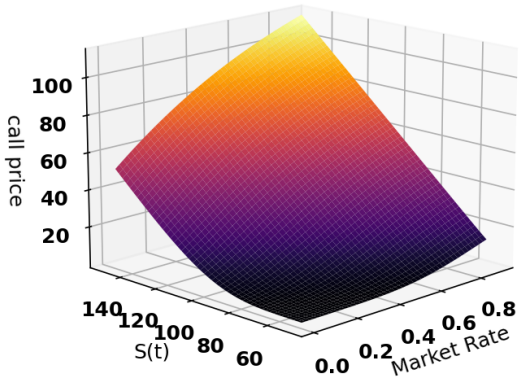


Put Price

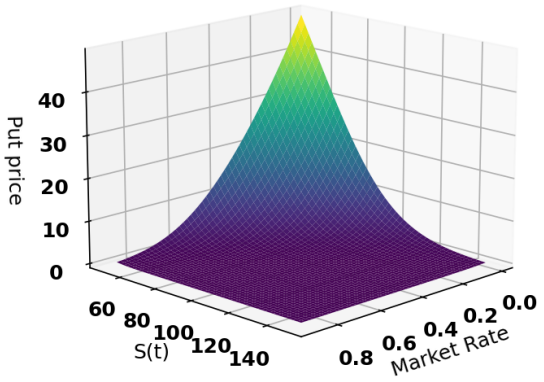


Call and Put price with varying Market Rate and S(t)

Call Price

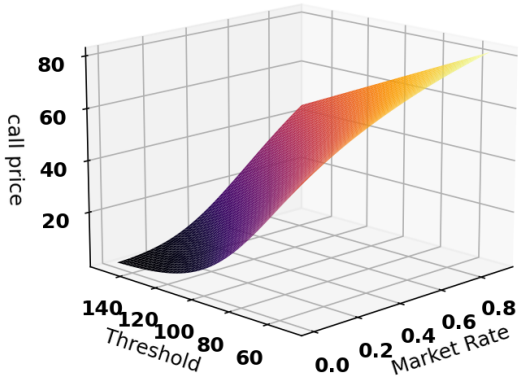


Put Price

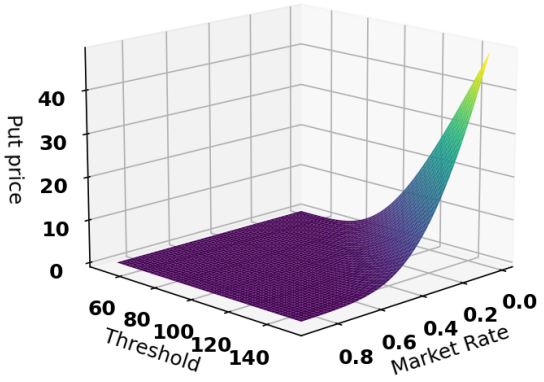


Call and Put price with varying Market Rate and Threshold

Call Price

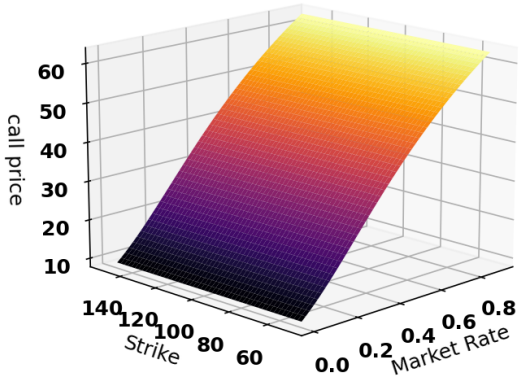


Put Price

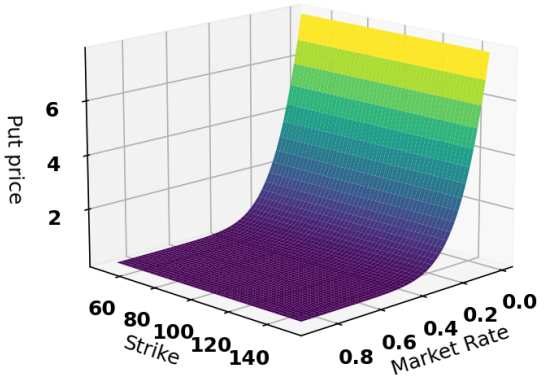


Call and Put price with varying Market Rate and Strike

Call Price

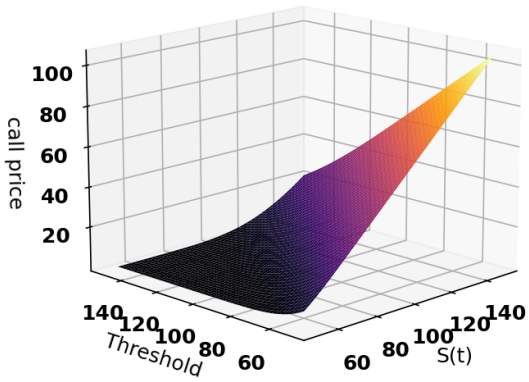


Put Price

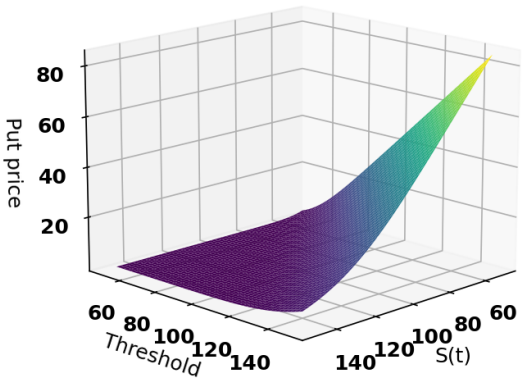


Call and Put price with varying S(t) and Threshold

Call Price

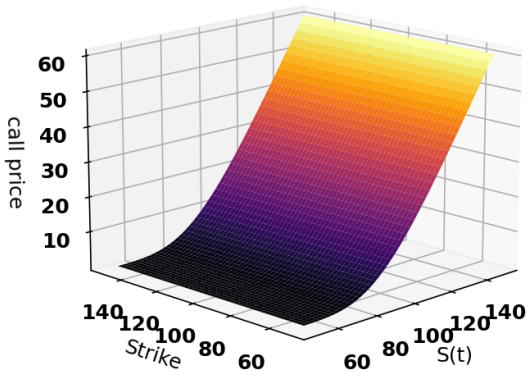


Put Price

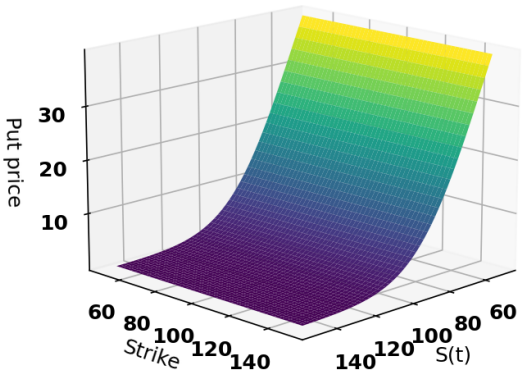


Call and Put price with varying S(t) and Strike

Call Price

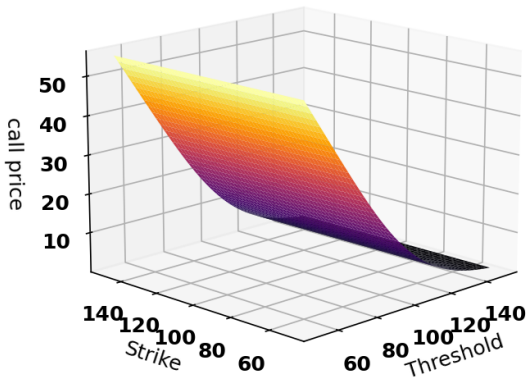


Put Price

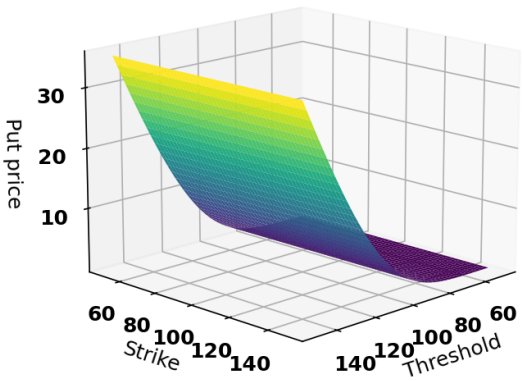


Call and Put price with varying Threshold and Strike

Call Price



Put Price



Shout Option

A shout option is a type of financial derivative where the holder has the right, but not the obligation, to "shout" to the seller and lock in the option's payoff at any time before expiration. This allows the option holder to capture profits if the price of the underlying asset moves in their favor without waiting until expiration.

Shout options are often used to hedge against sudden price movements in volatile markets, as they allow the holder to lock in profits without having to wait until expiration. However, this feature comes at a cost, as the option holder typically pays a higher premium for the option due to the added flexibility it provides.

The BSM pricing formula for a shout option is similar to that of a standard European option, but with the added flexibility of the shout feature. If the option is not shouted, the payout is calculated as the difference between the strike price and the price of the underlying asset at expiration for a call option, or the difference between the price of the underlying asset at expiration and the strike price for a put option. If the option is shouted, the payout is calculated as the difference between the strike price and the price of the underlying asset at the time of shouting for a call option, or the difference between the price of the underlying asset at the time of shouting and the strike price for a put option.

The BSM pricing formula for a shout call option is:

$$C = S_t \cdot N(d_1) - K \cdot e^{-r(T-t)} \cdot N(d_2) + e^{-r(T-t)} \cdot (S_{shout} - K) \cdot N(d_3)$$

where:

- C: the price of the shout call option
- S_t: the current price of the underlying asset
- K: the strike price
- r: the risk-free interest rate
- T: the time to expiration
- t: the current time
- σ: the volatility of the underlying asset's returns
- S_{shout}: the price of the underlying asset at the time of shouting
- $d_1 = \frac{\ln(S_t/K) + (r + (\sigma^2)/2) \cdot (T-t)}{\sigma\sqrt{T-t}}$
- $d_2 = d_1 - \sigma\sqrt{T-t}$
- $d_3 = \frac{\ln(S_{shout}/K) + (r + (\sigma^2)/2) \cdot (T-t)}{\sigma\sqrt{T-t}}$

The BSM pricing formula for a shout put option is:

$$P = K \cdot e^{-r(T-t)} \cdot N(-d_2) - S_t \cdot N(-d_1) + e^{-r(T-t)} \cdot (K - S_{shout}) \cdot N(-d_3)$$

where:

- P: the price of the shout put option

- P: the price of the shout put option
- S_t: the current price of the underlying asset
- K: the strike price
- r: the risk-free interest rate
- T: the time to expiration
- t: the current time
- σ: the volatility of the underlying asset's returns
- S_{shout}: the price of the underlying asset at the time of shouting
- $d_1 = \frac{\ln(S_t/K)+(r+(\sigma^2)/2)\cdot(T-t)}{\sigma\sqrt{T-t}}$

I will now derive this pricing formulae:-

$$P = e^{-rT}(S_0 e^{(b-r-\frac{1}{2}\sigma^2)T} N(d_1') - Ke^{-rT}N(d_1) - SN(d_2'))$$

To derive this formula, we first note that the shout option has a fixed payoff if the underlying asset price is above the shout level, and pays nothing if the underlying asset price is below the strike price. The payoff for a shout option can be represented as follows:

$$Q = \begin{cases} S_T - K, & \text{if } S_T \geq S \\ S - K, & \text{if } K < S_T < S \\ 0, & \text{otherwise} \end{cases}$$

where Q is the payoff, S_T is the price of the underlying asset at expiry, K is the strike price, and S is the shout level.

The price of a shout option can be calculated using the Black-Scholes model with an adjustment for the shout level.

Using the risk-neutral valuation principle, we can assume that the expected payoff of the option is equal to its present value. Therefore, we can write:

$$P = e^{-rT}E[Q]$$

To calculate $E[Q]$, we need to derive the distribution of S_T. Using the Black-Scholes model with an adjustment for the shout level, we can derive the following expressions for d_1' and d_2':

$$d_1' = \frac{\ln(S_0/S) + (b + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}$$

and

$$d_2' = d_1' - \sigma\sqrt{T}$$

Substituting these expressions into the equation for the expected payoff, we get:

$$P = e^{-rT}E[Q]$$

$$= e^{-rT}[(S_0 - K)N(d_1')e^{-(b-r)T} + (S - K)[N(d_2') - N(d_1)]]$$

$$= e^{-rT}(S_0 e^{(b-r-\frac{1}{2}\sigma^2)T} N(d_1') - K e^{-rT} N(d_1) - S N(d_2'))$$

which is the formula for pricing a shout option with strike K and shout level S.

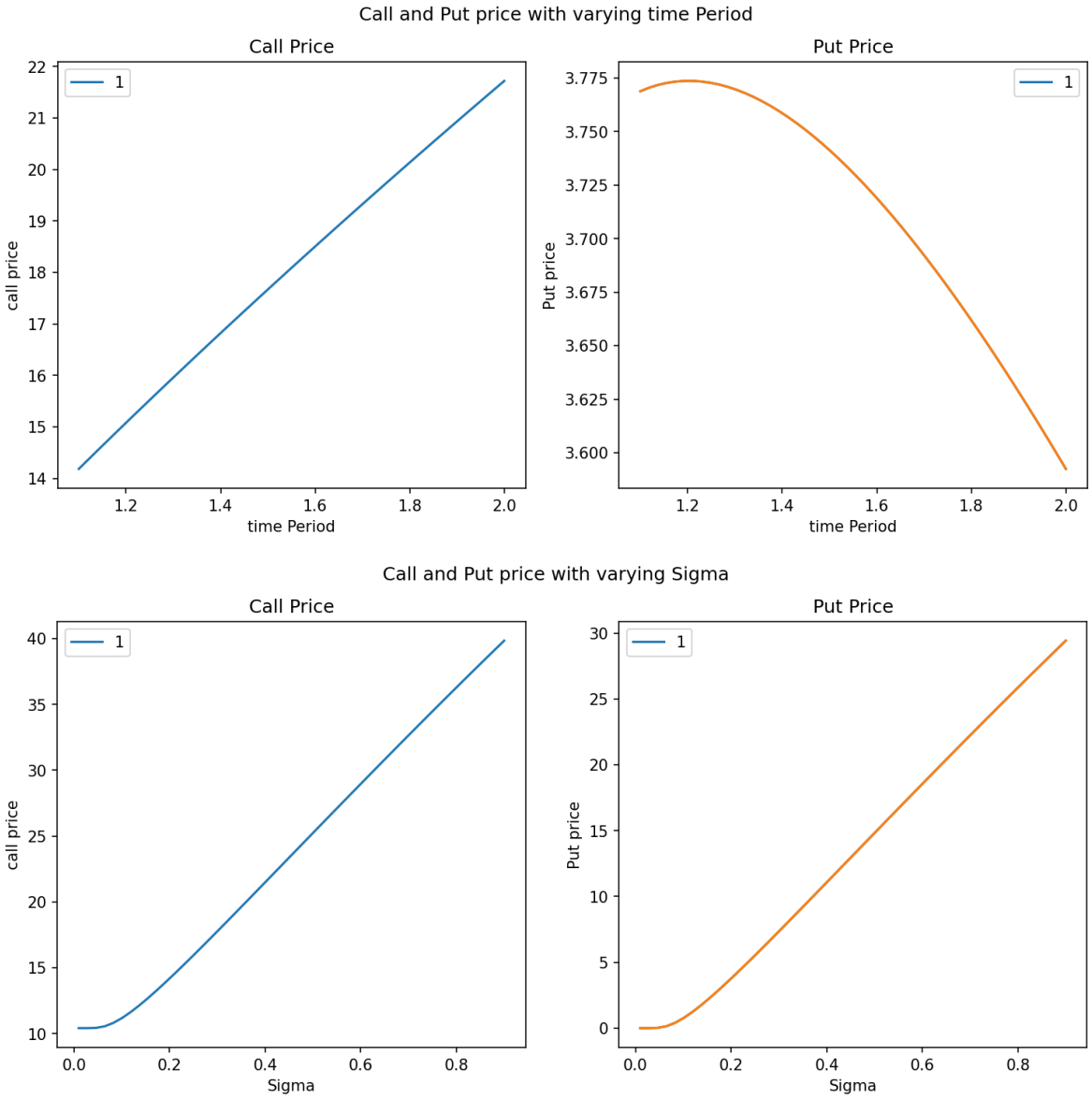
Below is the sensitivity analysis of a Gap Option along with all the relevant plots:-

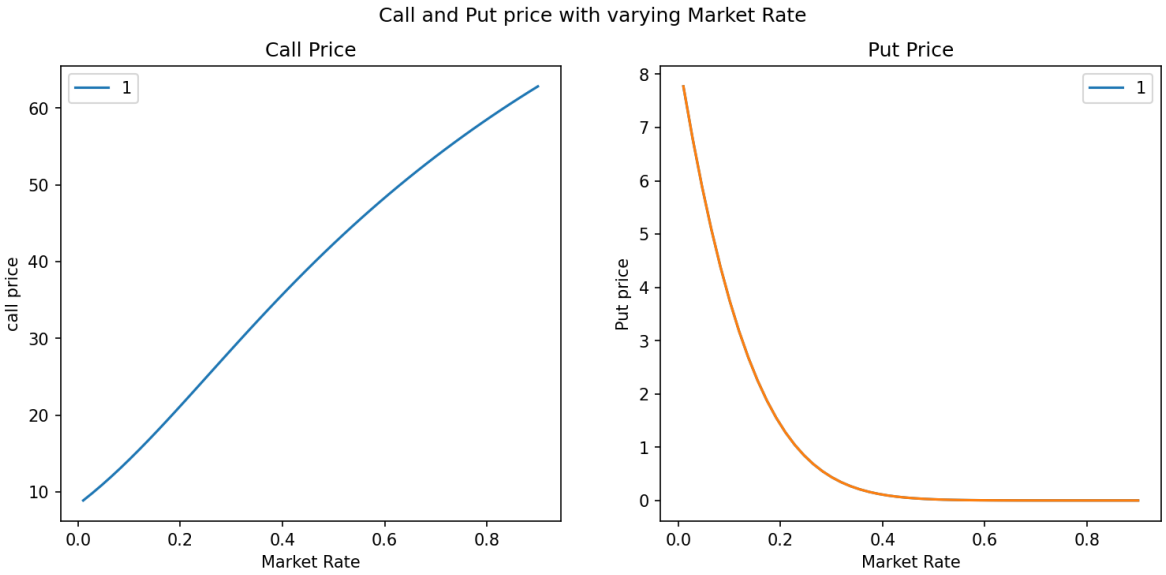
In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import norm
```

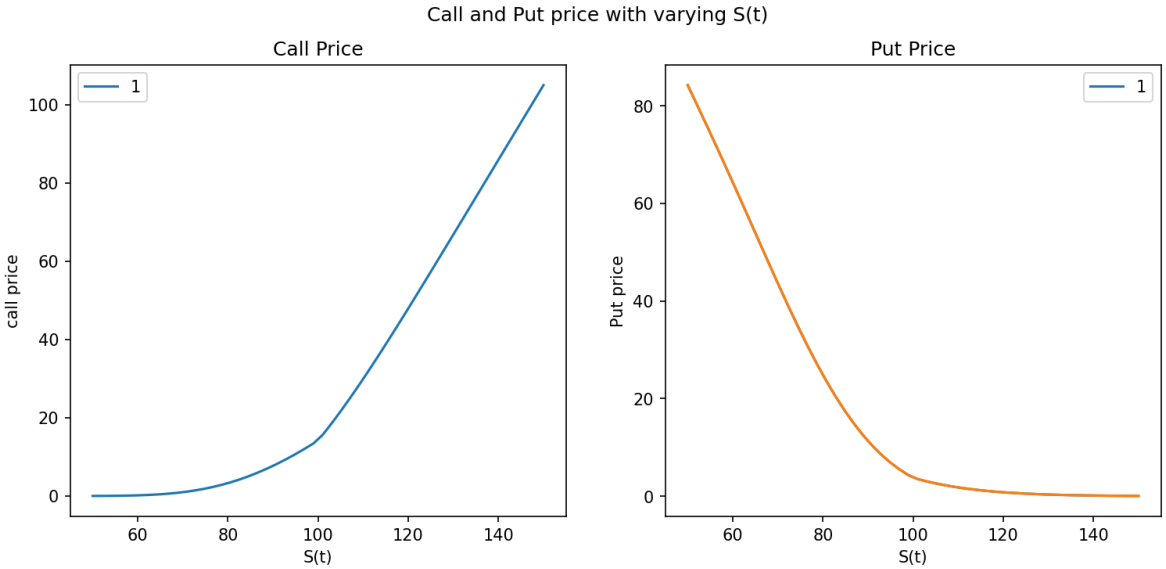
In [ ]:
```
def ShoutOptionCall(t, T, sigma, r, St, K, L):
    d1 = (np.log(St/K) + (r + sigma**2/2)*(T-t)) / (sigma*np.sqrt(T-t))
    d2 = d1 - sigma*np.sqrt(T-t)
    vanilla_price = St*norm.cdf(d1) - K*np.exp(-r*(T-t))*norm.cdf(d2)
    if St >= L:
        price = vanilla_price + (St - L)*np.exp(-r*(T-t))*norm.cdf((np.lo
    else:
        price = vanilla_price
    return price

def ShoutOptionPut(t, T, sigma, r, St, K, L):
    d1 = (np.log(St/K) + (r + sigma**2/2)*(T-t)) / (sigma*np.sqrt(T-t))
    d2 = d1 - sigma*np.sqrt(T-t)
    vanilla_price = K*np.exp(-r*(T-t))*norm.cdf(-d2) - St*norm.cdf(-d1)
    if St <= L:
            price = vanilla_price + (L - St)*np.exp(-r*(T-t))*norm.cdf(-(
    else:
        price = vanilla_price
    return price
```

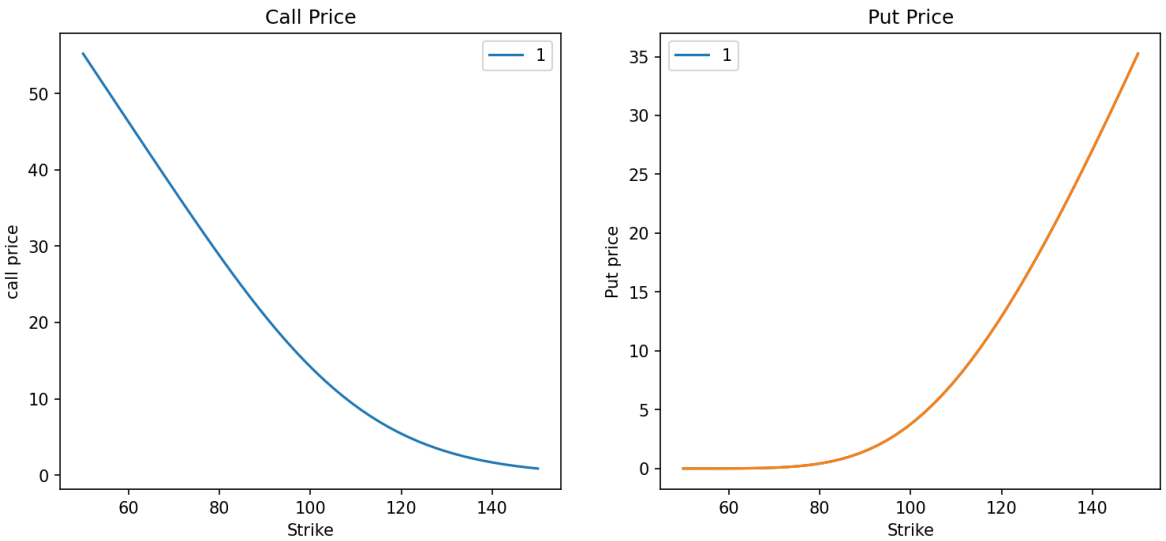
In [ ]:
```
# def optionPrice(t, T, sigma, r, St, K, L):
varying = []
varying.append(np.linspace(0,1,50))
varying.append(np.linspace(1.1,2,50))
varying.append(np.linspace(0.01,0.90,50))
varying.append(np.linspace(0.01,0.90,50))
varying.append(np.linspace(50,150,50))
varying.append(np.linspace(50,150,50))
varying.append(np.linspace(50,150,50))

names = ["start time", "time Period", "Sigma", "Market Rate", "S(t)", "St
for i in range(1,6):
    fig, ax = plt.subplots(nrows = 1,ncols = 2)
    for st in [1]:
        call_p = []
        put_p = []
        params = [0, 1.1, 0.2, 0.1, 100, 100, 100]
        for ii in range(len(varying[i])):
            params[i]=varying[i][ii]
            call_p.append(ShoutOptionCall(params[0],params[1],params[2],p
            put_p.append(ShoutOptionPut(params[0],params[1],params[2],par
            fig.suptitle("Call and Put price with varying "+names[i])
        ax[0].plot(varying[i],call_p)
        ax[1].plot(varying[i],put_p)
    fig.set_size_inches(12, 5)
    fig.set_dpi(150)
    ax[0].set_title("Call Price")
    ax[0].set_xlabel(names[i])
    ax[0].set_ylabel("call price")
    ax[1].plot(varying[i],put_p)
    ax[1].set_title("Put Price")
    ax[1].set_xlabel(names[i])
    ax[1].set_ylabel("Put price")
    ax[0].legend([1])
    ax[1].legend([1])
    plt.show()

for i in range(7):
    for j in range(i+1,7):
        fig, ax = plt.subplots(nrows = 1,ncols = 2,subplot_kw={"projectio
        call_p = np.zeros((50,50))
        put_p = np.zeros((50,50))
        p_axis, q_axis = np.meshgrid(varying[i],varying[j])
        params = [0, 1.1, 0.2, 0.1, 100, 100, 100]
        for ii in range(len(p_axis)):
            for jj in range(len(q_axis)):
                params[i]=varying[i][ii]
                params[j]=varying[j][jj]
                call_p[ii][jj]=ShoutOptionCall(params[0],params[1],params
                put_p[ii][jj]=ShoutOptionPut(params[0],params[1],params[2
        call_p=call_p.T
        put_p=put_p.T
        fig.suptitle("Call and Put price with varying "+names[i]+" and "+
        fig.set_size_inches(12, 5)
        fig.set_dpi(150)
        ax[0].plot_surface(p_axis,q_axis,call_p,cmap='inferno')
        ax[0].set_title("Call Price")
        ax[0].set_xlabel(names[i])
        ax[0].set_ylabel(names[j])
        ax[0].set_zlabel("call price")
```

```
        ax[1].plot_surface(p_axis,q_axis,put_p,cmap='viridis')
        ax[1].set_title("Put Price")
        ax[1].set_xlabel(names[i])
        ax[1].set_ylabel(names[j])
        ax[1].set_zlabel("Put price")
        ax[0].view_init(15,-135)
        ax[1].view_init(15,45)
        plt.show()
```



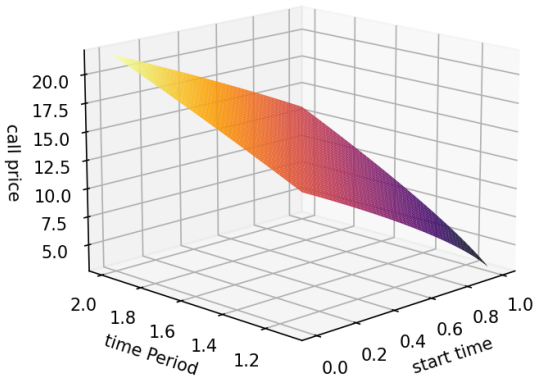

Call and Put price with varying Market Rate
Call Price
Put Price
call price
Put price
Market Rate
Market Rate
1
1


Call and Put price with varying S(t)
Call Price
Put Price
call price
Put price
S(t)
S(t)
1
1

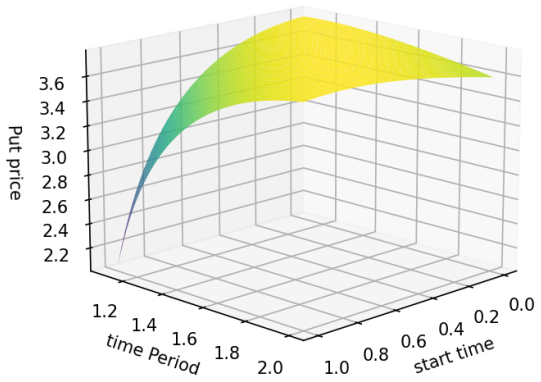

Call and Put price with varying Strike
Call Price
Put Price
call price
Put price
Strike
Strike
1
1

Call and Put price with varying start time and time Period

Call Price

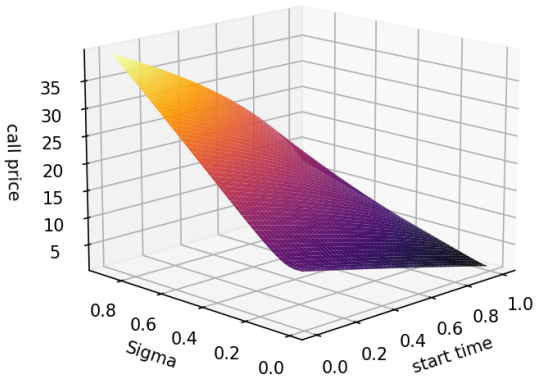


Put Price

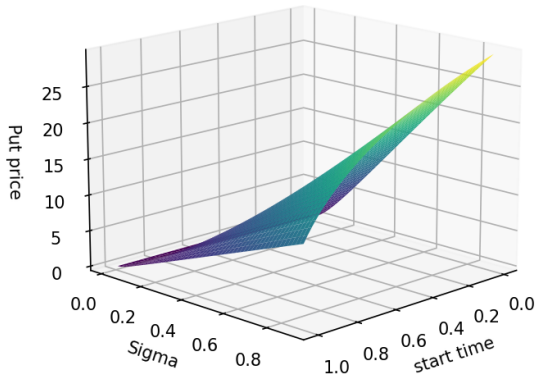


Call and Put price with varying start time and Sigma

Call Price

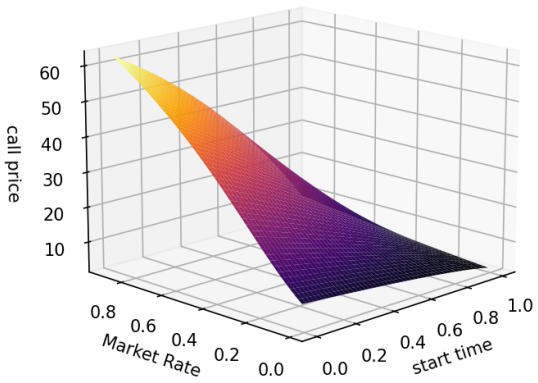


Put Price

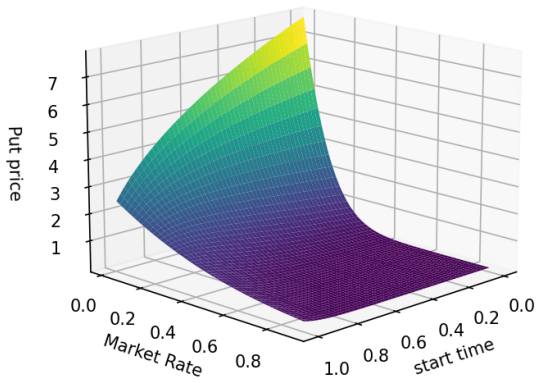


Call and Put price with varying start time and Market Rate

Call Price

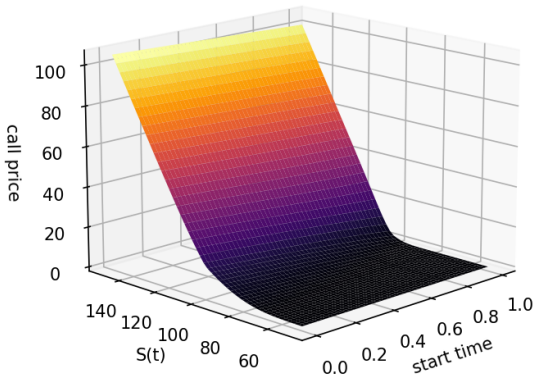


Put Price

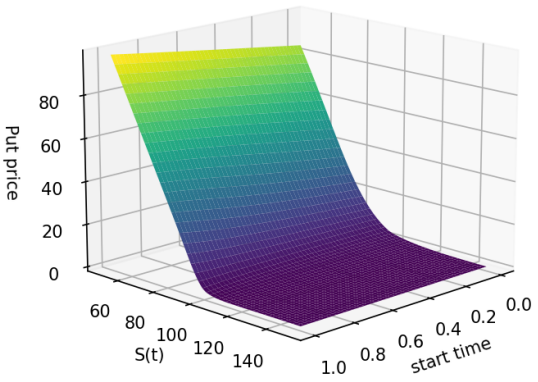


Call and Put price with varying start time and S(t)

Call Price

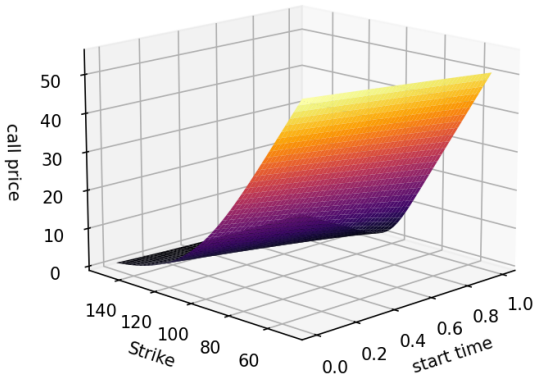


Put Price

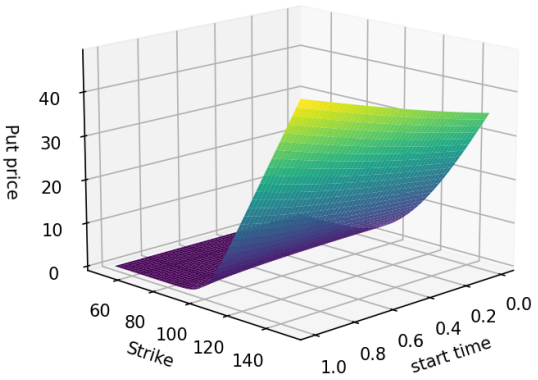


Call and Put price with varying start time and Strike

Call Price

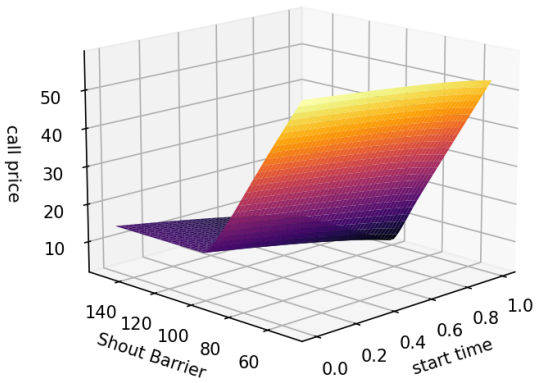


Put Price

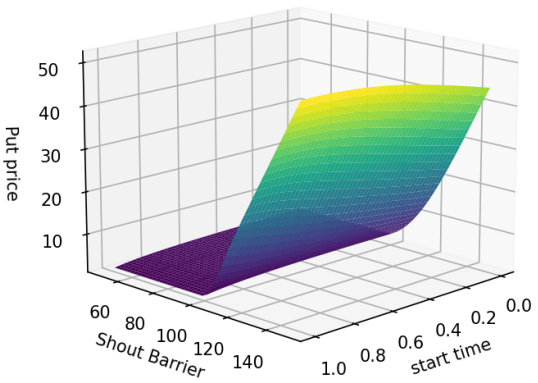


Call and Put price with varying start time and Shout Barrier

Call Price

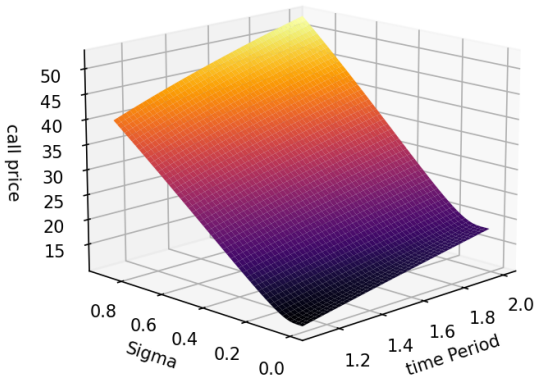


Put Price

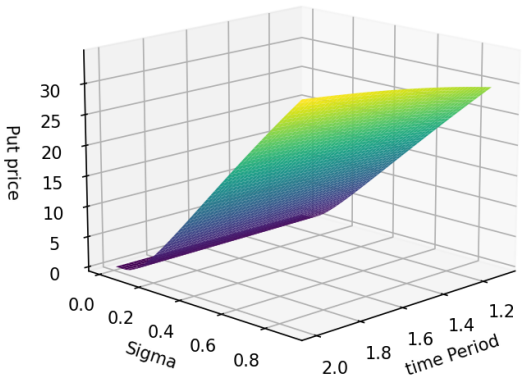


Call and Put price with varying time Period and Sigma

Call Price

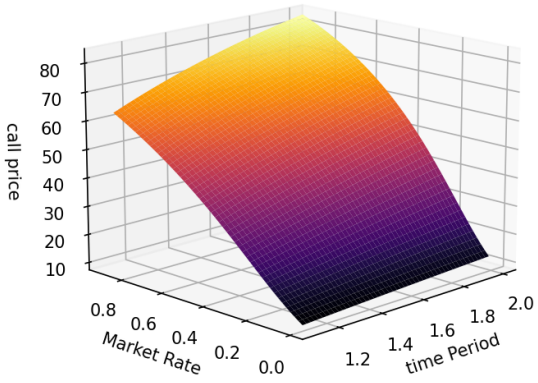


Put Price

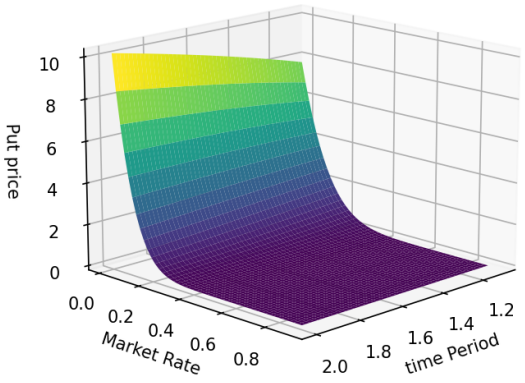


Call and Put price with varying time Period and Market Rate

Call Price

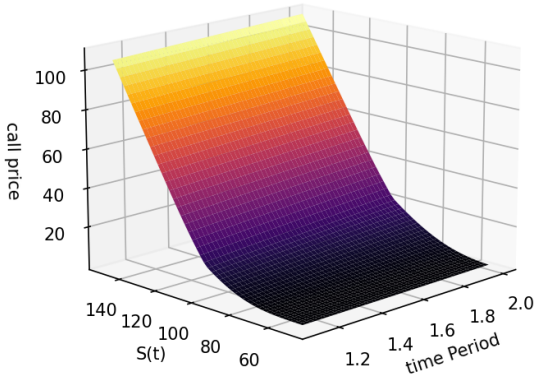


Put Price

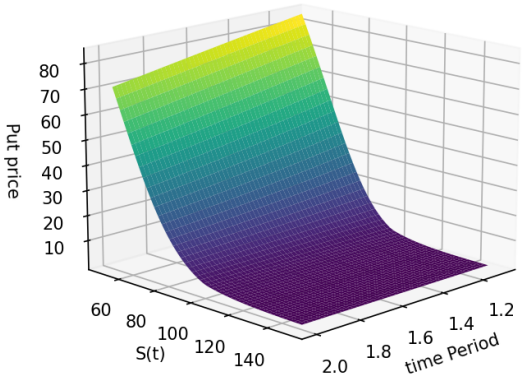


Call and Put price with varying time Period and S(t)

Call Price

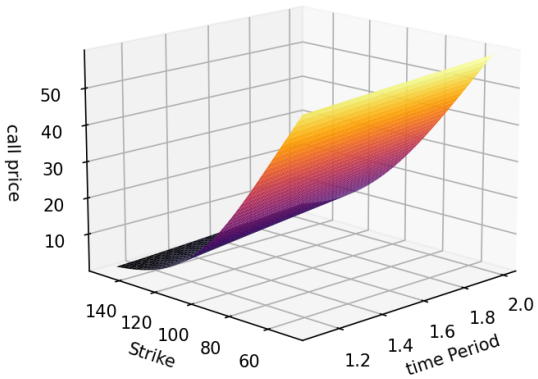


Put Price

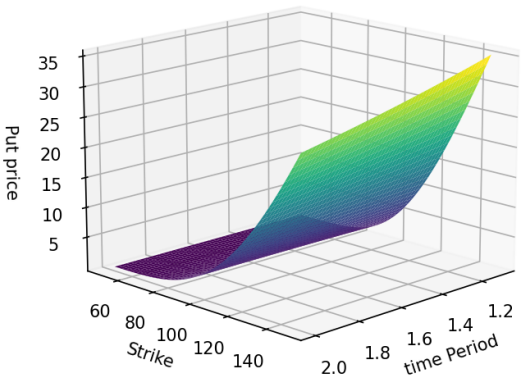


Call and Put price with varying time Period and Strike

Call Price

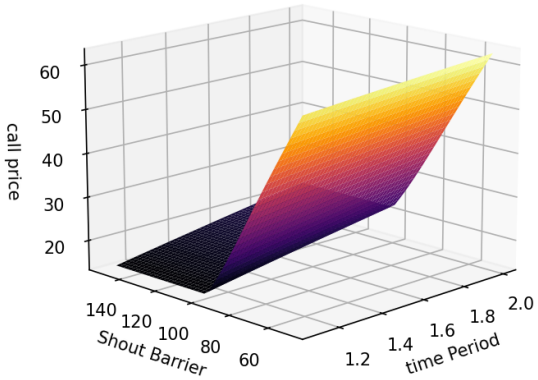


Put Price

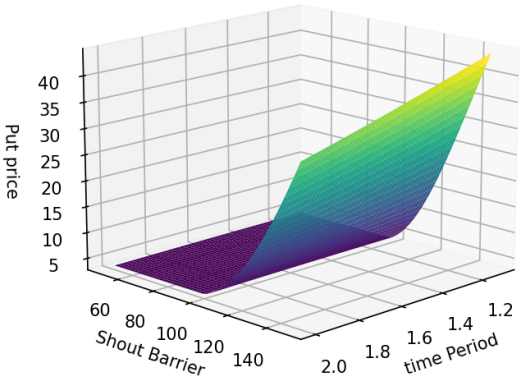


Call and Put price with varying time Period and Shout Barrier

Call Price

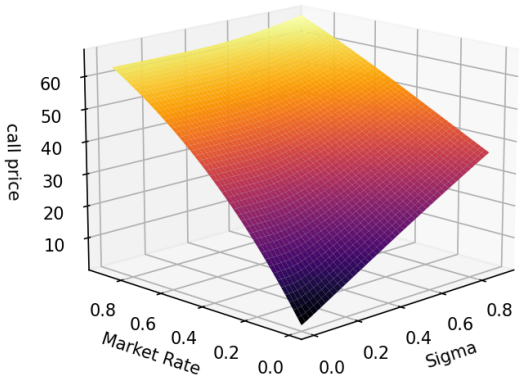


Put Price

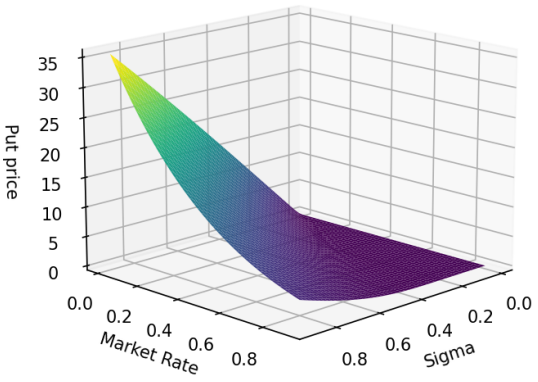


Call and Put price with varying Sigma and Market Rate

Call Price

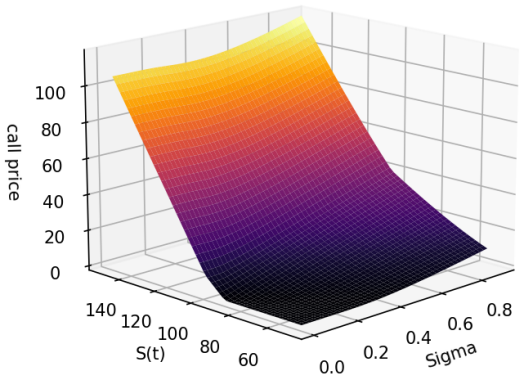


Put Price

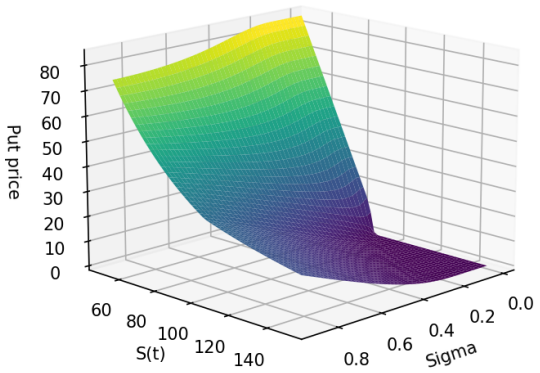


Call and Put price with varying Sigma and S(t)

Call Price

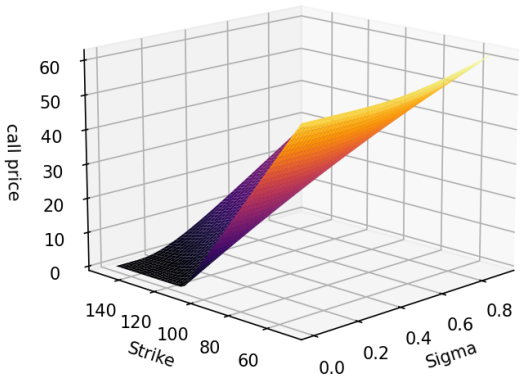


Put Price

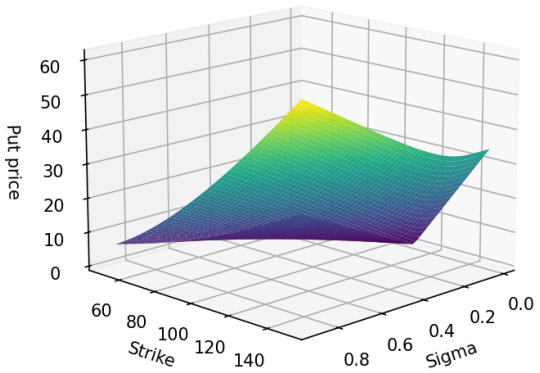


Call and Put price with varying Sigma and Strike

Call Price

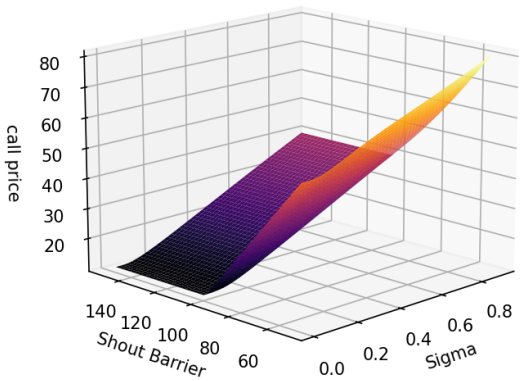


Put Price

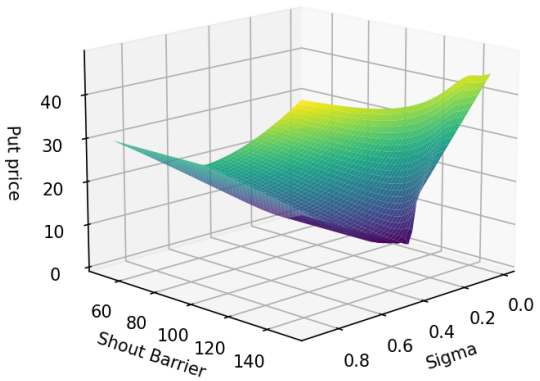


Call and Put price with varying Sigma and Shout Barrier

Call Price

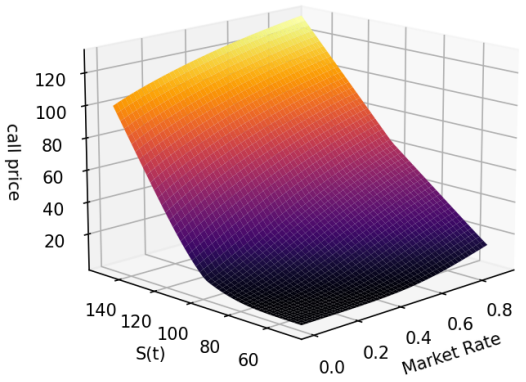


Put Price

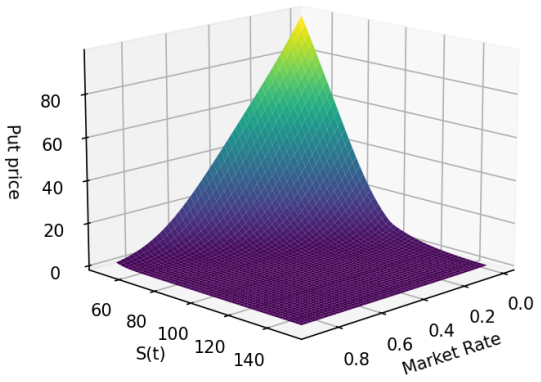


Call and Put price with varying Market Rate and S(t)

Call Price

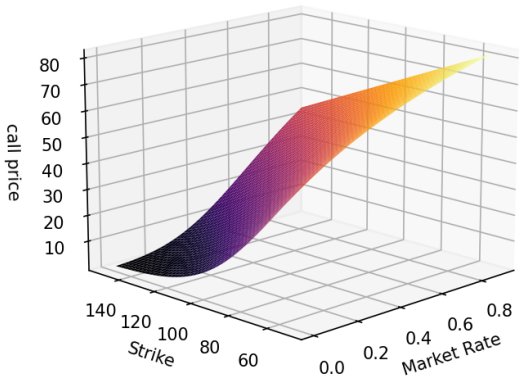


Put Price

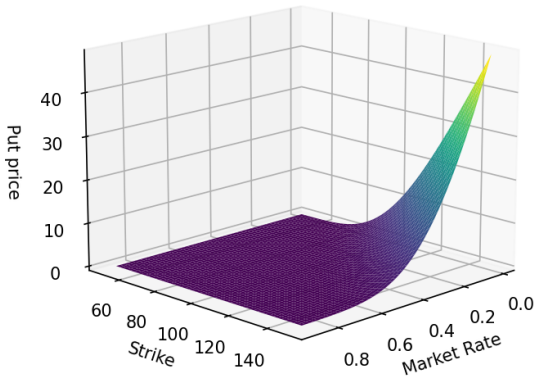


Call and Put price with varying Market Rate and Strike

Call Price

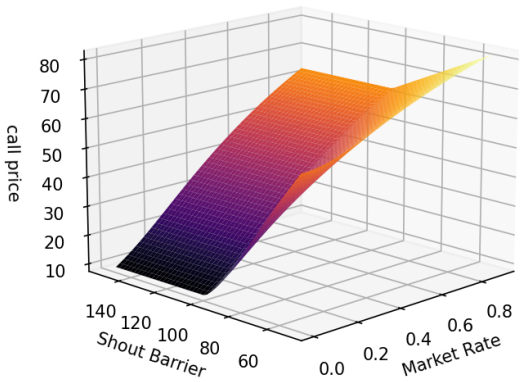


Put Price

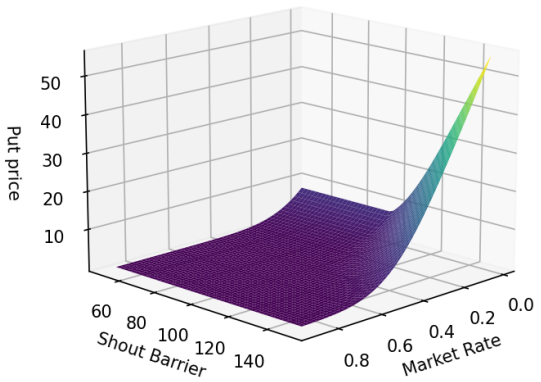


Call and Put price with varying Market Rate and Shout Barrier

Call Price

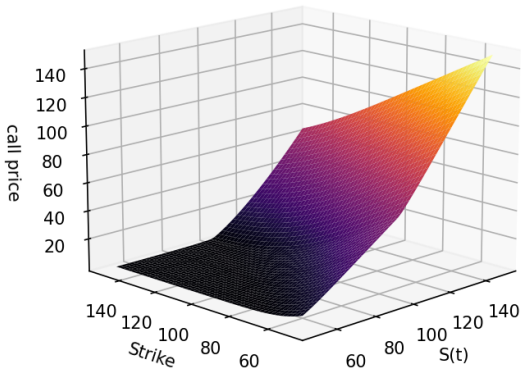


Put Price

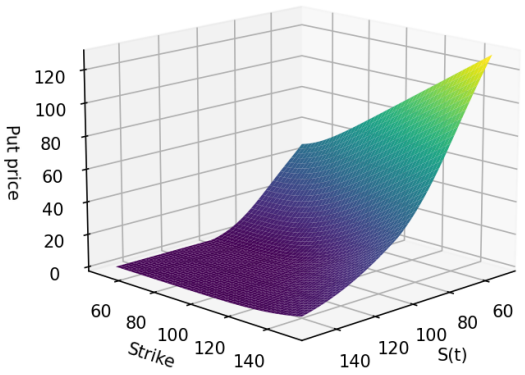


Call and Put price with varying S(t) and Strike

Call Price

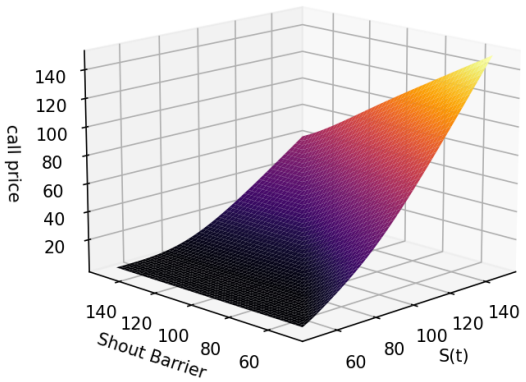


Put Price

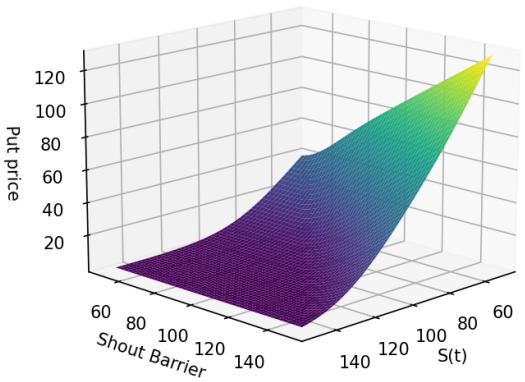


Call and Put price with varying S(t) and Shout Barrier

Call Price

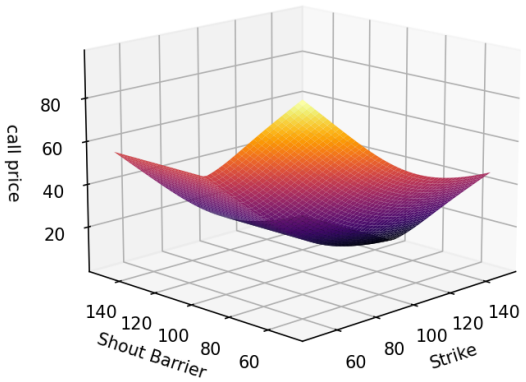


Put Price

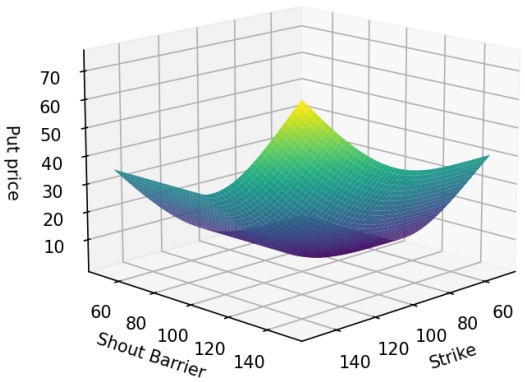


Call and Put price with varying Strike and Shout Barrier

Call Price



Put Price



Conclusions

Below is a brief overview of behaviour of option prices after observing the sensitivity analysis of the exotics . Note that these analysis are for call option and analogous results may be derived for the correspondng put option:-

Binary options:

Strike price (K): As the strike price of a binary option moves further away from the current price of the underlying asset, the probability of the option expiring in the money decreases, and the option becomes cheaper to purchase.
Time to expiration (T): As the time to expiration of a binary option decreases, the option becomes cheaper to purchase, since there is less time for the underlying asset to move in a favorable direction.
Volatility (σ): As the volatility of the underlying asset increases, the probability of the option expiring in the money also increases, and the option becomes more expensive to purchase.

Gap options:

Width of the gap (G): As the width of the gap between the strike price and the current price of the underlying asset increases, the option becomes cheaper to purchase, since there is less chance of the underlying asset moving into the gap before the option expires.
Time to expiration (T): As the time to expiration of a gap option decreases, the option becomes cheaper to purchase, since there is less time for the underlying asset to move into the gap before the option expires.
Volatility (σ): As the volatility of the underlying asset increases, the chance of the underlying asset moving into the gap before the option expires also increases, and the option becomes more expensive to purchase.

Shout options:

Strike price (K): As the strike price of a shout option moves further away from the current price of the underlying asset, the option becomes more expensive to purchase, since the option holder has the right to shout for the strike price at any time.
Time to expiration (T): As the time to expiration of a shout option increases, the option becomes more expensive to purchase, since the option holder has more time to shout for the strike price.
Volatility (σ): As the volatility of the underlying asset increases, the chance of the option holder shouting for the strike price before expiration also increases, and the option becomes more expensive to purchase.

References

- *Options, Futures and Other Derivatives by John C. Hull*