Black-Scholes Model

Formula

$$C(S,t) = N(d_1)S - N(d_2)Ke^{-rT}$$

$$d_1 = \frac{ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

$C(S,t)$	(call option price)
$N(\,)$	(cumulative distribution function)
$T = (T_1 - t)$	(time left til maturity (in years))
S	(stock price)
K	(strike price)
r	(risk free rate)
σ	(volatility)

WallStreetMojo

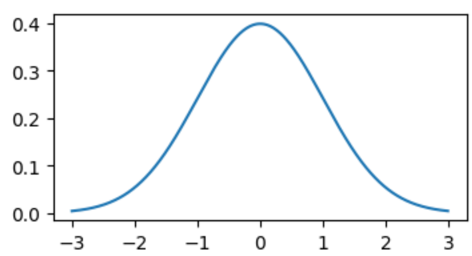
BLACK SCHOLE MERTON

Double-click (or enter) to edit

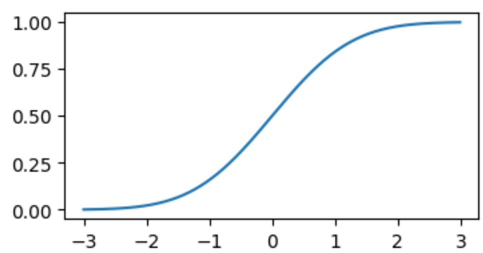
```
import math
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"]=(4, 3)
```

plot normal distribution

```
x = np.arange(-3,3,0.01)
y = stats.norm.pdf(x)
plt.plot(x,y)
plt.show()
```




```
x = np.arange(-3,3,0.01)
y = stats.norm.cdf(x)
plt.plot(x, y)
plt.show()
```



```
def BSM(S0 = 100, K=100, T=1, r=0.05, sigma= 0.2, option_type='call'):
  d1= (math.log(S0/K) + (r + sigma**2/2)*T)/(sigma*math.sqrt(T))
  d2 = d1 - sigma*math.sqrt(T)
  N = stats.norm.cdf
  V = 0
  if (option_type == 'call'):
    V = S0* N(d1) - K*math.exp(-r*T) * N(d2)
  elif(option_type== 'put'):
    V = K*np.exp(-r*T) * N(-d2) - S0* N(-d1)
  del N
  return V

C = BSM(S0=100, K=100, T=1, r=0.05, sigma= 0.2, option_type='call')
P = BSM(S0=100, K=100, T=1, r=0.05, sigma = 0.2, option_type='put')
print('ATM call %.2f, ATM PUT %.2f' % (C,P))
```

```
ATM call 10.45, ATM PUT 5.57
```

Sensitivity to CALL and PUT Options

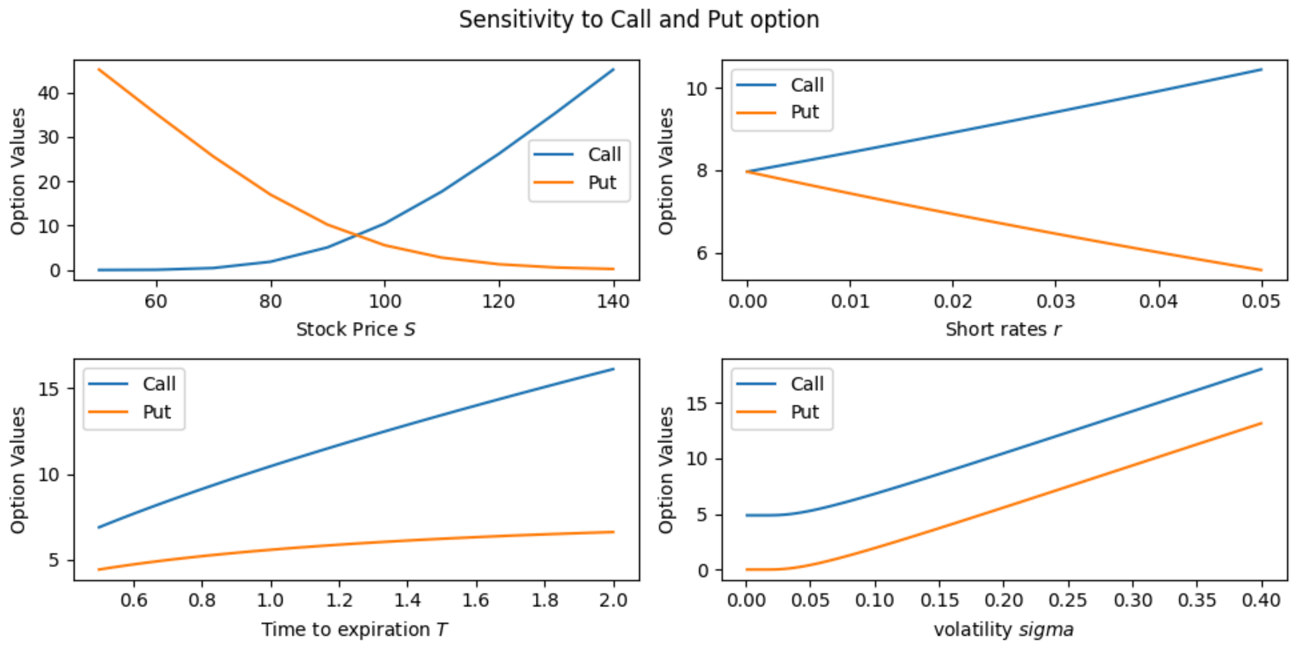
```
#call and put values vs stock prices
plt.figure(figsize=(10,5))
S_values = np.arange(50,150,10)
C = [BSM(S0= i, K=100, T=1, r=0.05, sigma= 0.2, option_type='call') for i in S_values]
P = [BSM(S0= i, K=100, T=1, r=0.05, sigma = 0.2, option_type='put') for i in S_values]
plt.subplot(2, 2, 1)
plt.plot(S_values, C, label ='Call')
plt.plot(S_values, P, label ='Put')
plt.xlabel('Stock Price $S$')
plt.ylabel('Option Values')
plt.legend()
plt.tight_layout()

#Call and Put values vs short-term interest rates
plt.subplot(2,2,2)
r_values = np.linspace(0.0, 0.05, 100)
C = [BSM(S0= 100, K=100, T=1, r= i, sigma= 0.2, option_type='call') for i in r_values]
P = [BSM(S0= 100, K=100, T=1, r= i, sigma = 0.2, option_type='put') for i in r_values]
plt.plot(r_values, C, label ='Call')
plt.plot(r_values, P, label ='Put')
plt.xlabel('Short rates $r$')
plt.ylabel('Option Values')
plt.legend()
plt.tight_layout()

#Call and Put values vs time-to-expiration
plt.subplot(2,2,3)
T_values = np.linspace(0.5, 2,100)
C = [BSM(S0= 100, K=100, T= i, r= 0.05, sigma= 0.2, option_type='call') for i in T_values]
P = [BSM(S0= 100, K=100, T= i, r= 0.05, sigma = 0.2, option_type='put') for i in T_values]
plt.plot(T_values, C, label ='Call')
plt.plot(T_values, P, label ='Put')
plt.xlabel('Time to expiration $T$')
plt.ylabel('Option Values')
plt.legend()
plt.tight_layout()

#Call and put values vs Volatility
plt.subplot(2,2,4)
sigma_values = np.linspace(0.001, 0.4, 100)
C = [BSM(S0= 100, K=100, T= 1, r= 0.05, sigma= i, option_type='call') for i in sigma_values]
P = [BSM(S0= 100, K=100, T= 1, r= 0.05, sigma = i, option_type='put') for i in sigma_values]
plt.plot(sigma_values, C, label ='Call')
plt.plot(sigma_values, P, label ='Put')
plt.xlabel('volatility $sigma$')
plt.ylabel('Option Values')
plt.legend()
plt.tight_layout()
plt.axis('tight') #for displaying properly

#adding title
plt.suptitle('Sensitivity to Call and Put option')
plt.tight_layout()
plt.show()
```



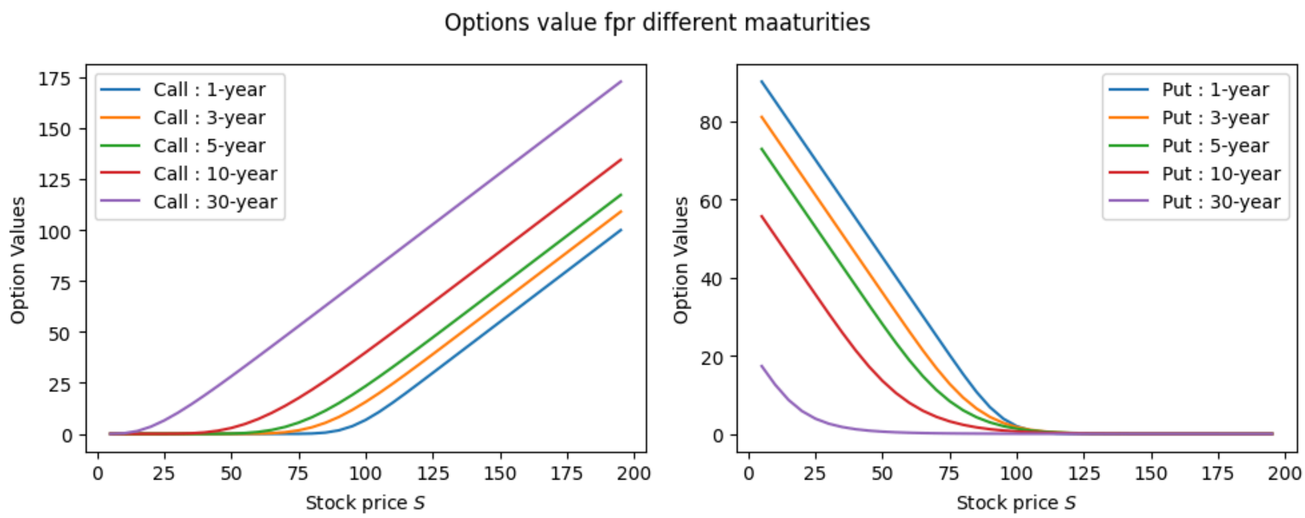
Options value for different maturities

```
#options value for different maturities
plt.figure(figsize=(10,4))
S_values = np.arange(5,200,5)

plt.subplot(1,2,1)
for t in [1,3,5,10,30]:
 C = [BSM(S0= i, K=100, T= t, r= 0.05, sigma= 0.1, option_type='call') for i in S_values]
 plt.plot(S_values, C, label ='Call : %d-year' % t)

plt.xlabel('Stock price $S$')
plt.ylabel('Option Values')
plt.legend()

plt.subplot(1,2,2)
for t in [1,3,5,10,30]:
 P = [BSM(S0= i, K=100, T= t, r= 0.05, sigma = 0.1, option_type='put') for i in S_values]
 plt.plot(S_values, P, label ='Put : %d-year' % t)
plt.xlabel('Stock price $S$')
plt.ylabel('Option Values')
plt.legend()
plt.suptitle('Options value fpr different maaturities')
plt.tight_layout()
plt.show()
```



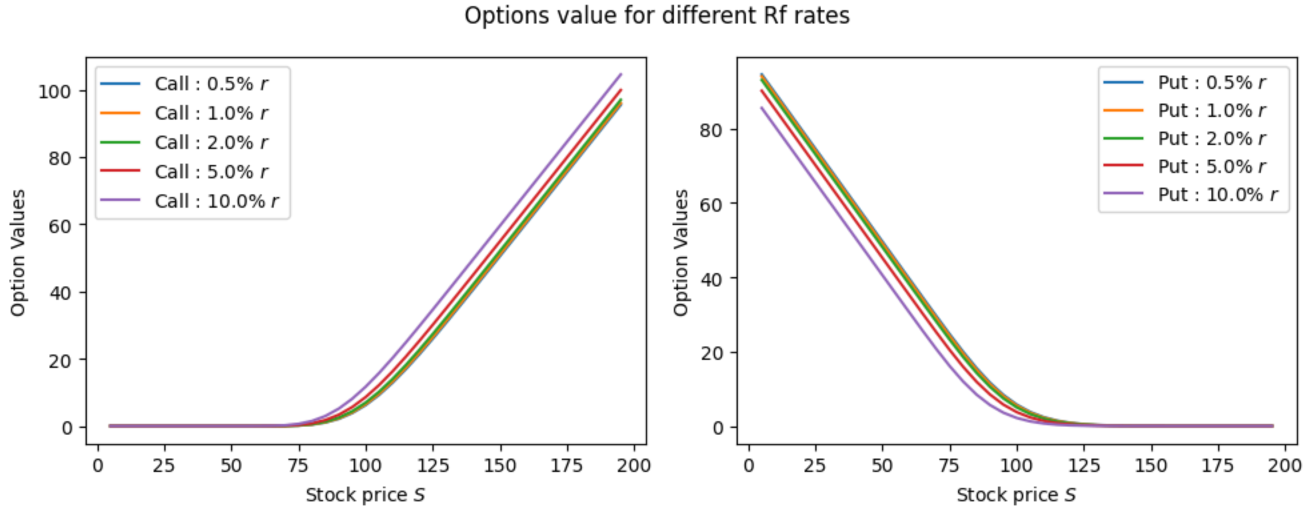
Options value for different risk free rates

```
#options value for different risk free rates
plt.figure(figsize=(10,4))
S_values = np.arange(5,200,5)

plt.subplot(1,2,1)
for r in [0.005,0.01,0.02,0.05,0.10]:
 C = [BSM(S0= i, K=100, T= 1, r= r, sigma= 0.15, option_type='call') for i in S_values]
 plt.plot(S_values, C, label = f'Call : {r:.1%} $r$')

plt.xlabel('Stock price $S$')
plt.ylabel('Option Values')
plt.legend()

plt.subplot(1,2,2)
for r in [0.005,0.01,0.02,0.05,0.10]:
 P = [BSM(S0= i, K=100, T= 1, r= r, sigma = 0.15, option_type='put') for i in S_values]
 plt.plot(S_values, P, label = f'Put : {r:.1%} $r$')
plt.xlabel('Stock price $S$')
plt.ylabel('Option Values')
plt.legend()
plt.suptitle('Options value for different Rf rates')
plt.tight_layout()
plt.show()
```



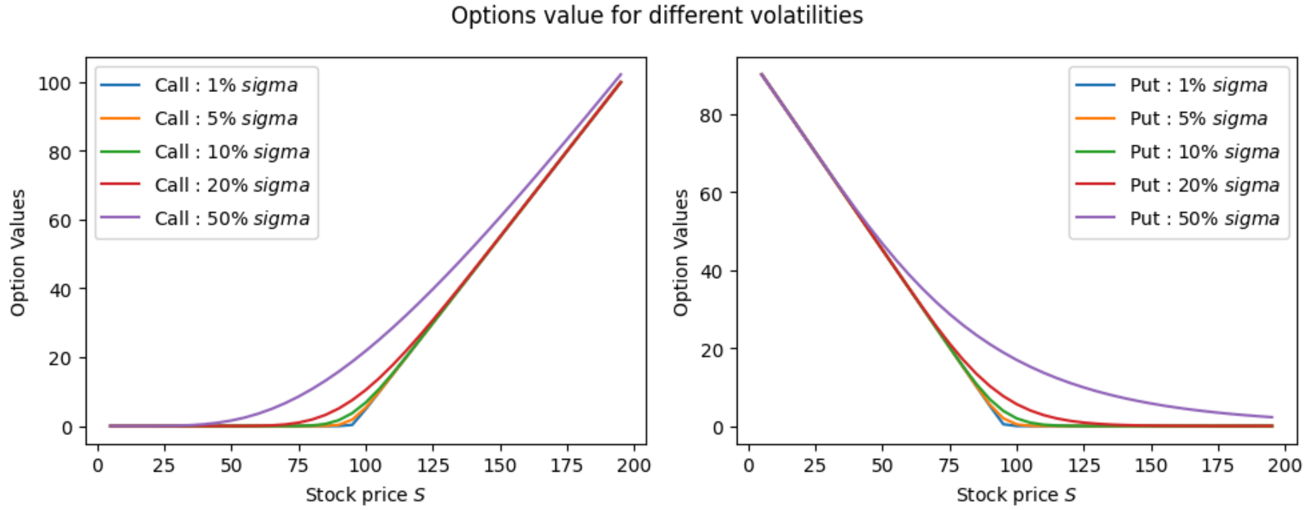
options values for different volatilities

```
#options value for different risk free rates
plt.figure(figsize=(10,4))
S_values = np.arange(5,200,5)

plt.subplot(1,2,1)
for v in [0.01,0.05,0.1,0.2,0.5]:
 C = [BSM(S0= i, K=100, T= 1, r= 0.05, sigma= v, option_type='call') for i in S_values]
 plt.plot(S_values, C, label = f'Call : {v:.0%} $sigma$')

plt.xlabel('Stock price $S$')
plt.ylabel('Option Values')
plt.legend()

plt.subplot(1,2,2)
for v in [0.01,0.05,0.1,0.2,0.5]:
 P = [BSM(S0= i, K=100, T= 1, r= 0.05, sigma = v, option_type='put') for i in S_values]
 plt.plot(S_values, P, label = f'Put : {v:.0%} $sigma$')
plt.xlabel('Stock price $S$')
plt.ylabel('Option Values')
plt.legend()
plt.suptitle('Options value for different volatilities')
plt.tight_layout()
plt.show()
```


Options value for different volatilities
Call : 1% sigma
Call : 5% sigma
Call : 10% sigma
Call : 20% sigma
Call : 50% sigma
Option Values
Stock price S
Put : 1% sigma
Put : 5% sigma
Put : 10% sigma
Put : 20% sigma
Put : 50% sigma
Option Values
Stock price S