

# 内容

- 欧式看涨期权简介
- 欧式看涨期权定价公式
- 编程实现定价公式



# 欧式看涨期权简介

# 股票期权

- 股票期权是一个金融契约；
- 赋予持有者在特定的日期、以一个指定的价格交易一个指定的普通特定份额的权利；
- 契约中指定的价格称为敲定价格或执行价格；
- 契约规定的日期称为到期日或截止日；
- 美式期权可在到期日之前的任何时刻执行，欧式期权只能在到期日执行
- 两种类型：看涨期权和看跌期权
  - 看涨期权：赋予持有者买某种标的资产的权利；
  - 看跌期权：赋予持有者卖某种标的资产的权利；



# 欧式看涨期权定价公式

# Black-Scholes-Merton (1973) 定价公式 (在没有分红情况下)

$$C(S_t, K, t, T, r, \sigma) \\ = S_t \cdot N(d_1) - e^{-r(T-t)} \cdot K \cdot N(d_2)$$

其中有

$$N(d) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^d e^{-\frac{1}{2}x^2} dx$$

$$d_1 = \frac{\log \frac{S_t}{K} + (r + \frac{\sigma^2}{2})(T-t)}{\sigma \sqrt{T-t}}$$

$$d_2 = \frac{\log \frac{S_t}{K} + (r - \frac{\sigma^2}{2})(T-t)}{\sigma \sqrt{T-t}}$$

不同参数含义

- $S_t$ : t时标的物价格
- $\sigma$ : 标的物固定波动率
- $K$ : 期权执行价格
- $T$ : 期权到期日
- $r$ : 固定无风险短期利率
- 最简单情形:  $t=0$



# 编程实现定价公式

# 编写程序

```
def bsm_call_value(S0, K, T, r, sigma):
```

```
    """
```

注释部分

```
    """
```

```
    from math import log, sqrt, exp
```

```
    from scipy import stats
```

```
    S0 = float(S0)
```

```
    d1 = (log(S0 / K) + (r + 0.5 * sigma ** 2) * T) / (sigma * sqrt(T))
```

```
    d2 = (log(S0 / K) + (r - 0.5 * sigma ** 2) * T) / (sigma * sqrt(T))
```

```
    value = (S0 * stats.norm.cdf(d1, 0.0, 1.0)
```

```
            - K * exp(-r * T) * stats.norm.cdf(d2, 0.0, 1.0))
```

```
# stats.norm.cdf:cumulative distribution function for normal
```

```
    return value
```

注意：新的模块SciPy，将在第五、六章介绍

# 注释部分

"""

Valuation of European call option in BSM model.

Analytical formula.

Parameters

=====

S0 : float

initial stock/index level

K : float

strike price

T : float

maturity date (in year fractions)

r : float

constant risk-free short rate

sigma : float

volatility factor in diffusion term

Returns

=====

value : float

present value of the European call option

"""



# 编写的累积标准正态分布函数？

```
def CND(X):  
    (a1, a2, a3, a4, a5) = (0.31938153, -0.356563782, 1.781477937,  
                             -1.821255978, 1.330274429)  
    L = abs(X)  
    K = 1.0 / (1.0 + 0.2316419 * L)  
    w = 1.0 - 1.0 / sqrt(2 * pi) * exp(  
        -L * L / 2.) * (a1 * K + a2 * K * K + a3 * pow(K, 3) + a4 * pow(K, 4) + a5 * pow(K, 5))  
    if X < 0:  
        w = 1.0 - w  
    return w
```

# 小 结

- 期权简介
- 欧式看涨期权定价公式
- 编程(操作)