Chapter 10 – Python期权分析

Frank Ziwei Zhang
School of Finance

上海对外经贸大学
SHANGHAI UNIVERSITY OF INTERNATIONAL BUSINESS AND ECONOMICS

Contents

10.1 A股股票期权市场

10.1.1 权证市场

1、权证的试点

A股市场的第一个权证是于1992年6月推出的，当时的权证名称是“大飞乐配股权证”。由于当时证券市场刚刚起步，市场主体自我约束能力和市场创新能力严重不足，法律法规不健全，而监管能力和效率又不高，导致权证市场过度投机，损害了普通投资者的权益，监管层在1996年6月底暂停了权证交易。

2、权证的重启并终结

2005年7月18日，沪深证券交易所同时颁布了《权证管理暂行办法》，为权证产品复出奠定了制度基础。随后，宝钢权证——宝钢JTBI580000于2005年8月22日在上海证券交易所正式挂牌交易，标志着权证重新回归。2011年8月18日，伴随着最后一只权证长虹CWBI的到期日，权证从此退出了历史舞台。

这次权证的重启和最后终结有许多复杂的原因。一方面是监管层为了顺利地推进股权分置改革，重新推出了权证，将权证作为非流通股股东支付给流通股股东的对价，以减轻对二级市场的冲击；另一方面是权证以其高杠杆性和T+0的交易特性吸引了众多个人投资者参加，中小投资者成为权证投资的主力军。但是，在缺乏对权证正确认识的前提下，大量中小投资者盲目开展权证交易，不少投资者将权证当成股票进行炒作，最终损失惨重。

10.1.2 股指期权合约

2015年2月9日，上海证券交易所正式挂牌交易上证50ETF期权，这意味着中国金融市场在权证谢幕的三年半迎来了全新的股指期权产品，2015年也被称为“股指期权元年”。表10-1就梳理了上证50ETF期权合约的主要要素。

合约标的（基础资产）	上证50交易型开放式指数证券投资基金（简称“50ETF”）
合约类型	认购期权（看涨期权）和认沽期权（看跌期权）
合约单位	10000份
合约到期月份	当月、下月和随后两个季月
行权价格	9个（1个平值合约、4个虚值合约、4个实值合约）
行权价格间距	3元或以下为0.05元； 3元（不含）至5元（含）为0.1元； 5元（不含）至10元（含）为0.25元； 10元（不含）至20元（含）为0.5元； 20元（不含）至100元（含）为2.5元； 100元以上为5元
行权方式	到期日行权（欧式期权）
交割方式	实物交割（业务规则另有规定的除外）
到期日	到期月份的第4个星期三（遇法定节假日顺延）
行权日	同合约到期日，行权指令提交时间为9:15-9:25，9:30-11:30，13:00-15:30

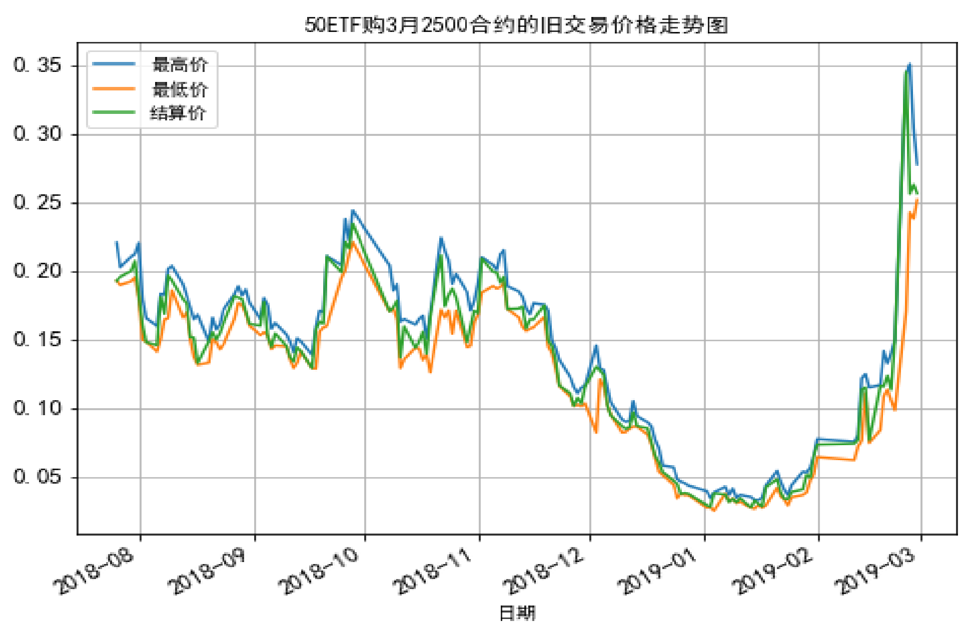
10.1.2 股指期权合约

交收日	行权日次一交易日
交易时间	上午9:15-9:25，9:30-11:30(9:15-9:25为开盘集合竞价时间) 下午13:00-15:00(14:57-15:00为收盘集合竞价时间）
买卖类型	买入开仓、买入平仓、卖出开仓、卖出平仓、备兑开仓、备兑平仓以及业务规则规定的其他买卖类型
最小报价单位	0.0001元
申报单位	1张或其整数倍
涨跌幅限制	•认购期权最大涨幅=max{合约标的前收盘价x0.5%，min[(2x合约标的前收盘价一行权价格)，合约标的前收盘价]x10%} •认购期权最大跌幅=合约标的前收盘价x10% •认沽期权最大涨幅=max{行权价格x0.5%，min[(2x行权价格一合约标的前收盘价)，合约标的前收盘价]x10% •认沽期权最大跌幅=合约标的前收盘价x10%
熔断机制	连续竞价期间，期权合约盘中交易价格较最近参考价格涨跌幅度达到或者超过50%且价格涨跌绝对值达到或者超过5个最小报价单位时，期权合约进入3分钟的集合竞价交易阶段
开仓保证金最低标准	•认购期权义务仓开仓保证金=[合约前结算价+max(12%x合约标的前收盘价-认购期权虚值，7%x合约标的前收盘价)]x合约单位 •认沽期权义务仓开仓保证金=min[合约前结算价+max(12%x合约标的前收盘价-认沽期权虚值，7%x行权价格)，行权价格]x合约单位
维持保证金最低标准	•认购期权义务仓维持保证金=[合约结算价+max(12%x合约标的收盘价-认购期权虚值，7%x合约标的收盘价)]x合约单位 •认沽期权义务仓维持保证金=min[合约结算价+max(12%x合标的收盘价-认沽期权虚值，7%x行权价格)，行权价格]x合约单位

10.1.2 股指期权合约

上海证券交易所提供了上证50ETF期权日交易数据的下载服务。这里就以2019年3月到期、执行价格为2.5元的“50ETF购3月2500合约”为例，运用Python演示该合约从上市首日（2018年7月26日）至2019年2月的日交易价格走势图（见图10-1），具体的代码如下：

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pylab import mpl
mpl.rcParams['font.sans-serif']=['SimHei']
mpl.rcParams['axes.unicode_minus']=False
option_data=pd.read_excel('D:\Zhangzw\Python\Python金融数据分析\RawData\第10章\SZ50ETF购3月2500合约每日价格数据.xlsx',sheet_name="Sheet1",header=0,index_col=0)   #导入外部数据
option_data.plot(figsize=(8,6),title=u'50ETF购3月2500合约的旧交易价格走势图',grid=True,fontsize=13)
```



10.1.2 股指期权合约

交易所	上海证券交易所	深圳证券交易所	中国金融期货交易所
合约标的	华夏上证50ETF 华泰柏瑞沪深300ETF	嘉实沪深300ETF	沪深300股票指数
合约类型	认购期权和认沽期权		
行权方式	欧式		
合约单位	10000份		每点人民币100元
合约到期月份	4个（当月、下月及随后两个季月）		6个（当月、下两个月及随后三个季月）
交割方式	实物交割		现金交割
行权价格间距	3元或以下为0.05元，3元至5元（含）为0.1元，5元至10元（含）为0.25元，10元至20元（含）为0.5元，20元至50元（含）为1元，50元至100元（含）为2.5元，100元以上为5元		对当月与下2个月合约，行权价格小于2500点（含）为25点，2500点至5000点（含）为50点，5000点至10000点（含）为100点，大于10000点为200点；对于随后3个季月合约行权价格小于2500点（含）为50点，2500点至5000点（含）为100点，5000点至10000点（含）为200点，大于10000点为400点
到期日	到期月份的第四个星期三（遇法定节假日顺延）		到期月份的第三个星期五（遇法定节假日顺延）
交易时间	上午9:15-9:25，9:30-11:30（9:15-9:25为开盘集合竞价时间） 下午13:00-15:00（14:57-15:00为收盘集合竞价时间）		上午9:25-11:30（9:25-9:30为开盘集合竞价时间） 下午13:00-15:00（14:57-15:00为收盘集合竞价时间）
买卖类型	买入开仓、买入平仓、卖出开仓、卖出平仓、备兑开仓、备兑平仓		买入开仓、买入平仓、卖出开仓、卖出平仓
最小报价单位	0.0001元		0.2点

10.2 期权类型和期权交易

10.2.1 期权的类型和要素

在期权市场上，期权合约可以分成看涨期权和看跌期权这两种基本类型。看涨期权（call option，也称“认购期权”）是指给期权持有人在未来某一时刻以约定价格有权利买入基础资产的金融合约；相反，看跌期权（put option，也称“认沽期权”）则是指给期权持有人在将来某一时刻以约定价格有权利卖出基础资产的金融合约。

期权还可以分为美式期权和欧式期权。美式期权（American option）可以在合约到期日之前的任何时刻行使权利，欧式期权（European option）则只能在到期日才能行使权利，A股市场的股指期权就是欧式期权。在理论上，欧式期权比美式期权更容易分析，当然美式期权的一些性质也常常可以从相应欧式期权的性质中推导出来。

期权的买入方被称为期权的多头（long position）或持有人，期权的卖出方被称为期权的空头（short position）或沽出方。因此，期权市场中有4类参与者，一是看涨期权的买入方（持有人或多头），二是看涨期权的卖出方（沽出方或空头），三是看跌期权的买入方，四是看跌期权的卖出方。

为了便于阅读，本书统一使用多头表示期权合约的买入方或持有人，空头表示期权合约的卖出方或沽出方。

10.2.1 期权的类型和要素

需要强调的是，期权的多头只有权利而无义务，具体而言就是看涨期权赋予多头买入某个基础资产的权利，但是多头可以有权选择不行使买入该基础资产的权利；同样，看跌期权赋予多头卖出某个基础资产的权利，但是多头也可以有权选择不行使卖出该基础资产的权利。

在期权合约中会明确合约到期日（expiration date或maturity date），合约中约定的买入价格或者卖出价格则称为执行价格（exercise price或strike price，又称“行权价格”）。

当然，期权多头拥有的这项权利是有代价的，必须付出一定金额的期权费（premium，也称“权利金”）作为对价给空头才能获得该项权利，并且期权费是在合约达成时就需要支付。

10.2.1 期权的类型和要素

序号	代码	名称	现价	涨跌	涨跌幅	今开	最高	最低	成交量	成交金额	期权成交量	期权持仓量	成交量PCR
1	510050	上证50ETF	2.539	-0.026	-1.01%	2.563	2.565	2.537	2.41亿	6.13亿	99.68万	232.32万	99.35%
2	510300	沪深300ETF	3.814	-0.031	-0.81%	3.843	3.845	3.813	1.74亿	6.66亿	53.18万	143.23万	115.84%
3	510500	中证500ETF	5.996	-0.024	-0.40%	6.022	6.032	5.986	7574万	4.55亿	23.82万	66.98万	121.16%

指数类 主力期货 上交所 深交所 中金所 上期所 上期能源 大商所 广期所 郑商所 港交所

认购								全部	认沽							
最新价	涨跌	涨跌幅	成交量	持仓量	IV	MIV	IV涨跌幅	行权价	最新价	涨跌	涨跌幅	成交量	持仓量	IV	MIV	IV涨跌幅
2023年06月(到期日2023-06-28;剩余34个自然日、22个交易日;合约乘数10000)														F 2.548	ATM IV 16.86%	
0.2017	-0.0265	-11.61%	260	236	17.25%	17.41%	-	2.350	0.0033	0.0007	26.92%	1.3万	9528	17.43%	17.47%	-
0.1559	-0.0246	-13.63%	2011	4692	17.59%	17.33%	0.13%	2.400	0.0080	0.0032	66.67%	1.8万	6.3万	17.35%	17.32%	2.72%
0.1141	-0.0214	-15.79%	4791	8412	17.18%	17.05%	3.54%	2.450	0.0168	0.0056	50.00%	2.9万	6.1万	17.27%	17.24%	1.82%
0.0785	-0.0184	-18.99%	2.6万	2.2万	17.02%	16.89%	2.10%	2.500	0.0308	0.0092	42.59%	6.3万	7.3万	16.98%	17.00%	2.41%
0.0501	-0.0140	-21.84%	7.0万	6.7万	16.82%	16.82%	2.44%	2.550	0.0526	0.0136	34.87%	12万	9.3万	16.92%	16.94%	2.93%
0.0304	-0.0090	-22.84%	10万	15万	17.05%	17.05%	4.21%	2.600	0.0827	0.0186	29.02%	7.0万	8.2万	17.09%	17.17%	4.62%
0.0176	-0.0057	-24.46%	7.8万	19万	17.41%	17.37%	4.28%	2.650	0.1196	0.0206	20.81%	3.1万	9.1万	17.35%	17.51%	1.54%
0.0102	-0.0028	-21.54%	4.9万	20万	18.01%	18.07%	5.91%	2.700	0.1621	0.0231	16.62%	1.3万	7.0万	17.85%	18.20%	0.91%
0.0058	-0.0011	-15.94%	2.1万	13万	18.64%	18.58%	7.60%	2.750	0.2081	0.0231	12.49%	2136	2.8万	18.92%	18.87%	10.94%
0.0032	-0.0005	-13.51%	2.7万	9.9万	19.16%	19.09%	7.48%	2.800	0.2546	0.0196	8.34%	1077	1.4万	18.71%	19.44%	7.41%
0.0018	-0.0003	-14.29%	4935	5.8万	19.77%	19.67%	6.39%	2.850	0.2981	0.0131	4.60%	334	6051	16.99%	20.29%	-2.99%
0.0013	-0.0001	-7.14%	3921	4.8万	21.00%	21.27%	6.29%	2.900	0.3500	0.0150	4.48%	154	5914	23.33%	21.33%	9.91%
0.0009	-0.0002	-18.18%	1332	2.8万	22.12%	22.43%	3.74%	2.950	0.4020	0.0170	4.42%	265	2382	19.53%	21.18%	-3.87%
0.0007	-0.0001	-12.50%	290	2.9万	23.52%	23.45%	5.26%	3.000	0.4464	0.0114	2.62%	45	2243	27.43%	21.39%	26.33%
0.0005	0.0000	0.00%	1113	3.8万	26.57%	26.27%	9.40%	3.100	0.5460	0.0110	2.06%	46	2154	27.49%	20.61%	38.22%
2023年06月(到期日2023-06-28;剩余34个自然日、22个交易日;合约乘数10142)														F 2.548	ATM IV 16.86%	
0.4270	-0.0279	-6.13%	129	987	25.36%	18.67%	-4.37%	2.120	0.0005	0.0000	0.00%	61	1.3万	25.84%	25.30%	-5.35%
0.3950	-0.0109	-2.69%	2	364	17.00%	20.23%	-37.45%	2.169	0.0006	0.0001	20.00%	50	8771	23.62%	23.05%	-1.46%

10.2.2 看涨期权到期时的盈亏

【例10-1】假定A投资者买入基础资产为100股W股票、执行价格为50元股的欧式看涨期权。假定W股票的当前市场价格为46元/股，期权到期日为4个月以后，购买1股W股票的期权价格(期权费)是6元，投资首最初投资为600元(100×6)，也就是一份看涨期权的期权费是600元。由于期权是欧式期权，因此A投资者只能生合约到期日才能行使期权。下面，考虑两种典型的情形：

情形1：如果在期权到期日，股票价格低于50股(比如下跌至43元股)，A投资者不会行使期权，因为没有必要以50元股的价格买入该股票，而是可以在市场上以低于50元股的价格购买股票。因此，A投资者将损失全部600元的初始投资，这也是A投资者的最大亏损。

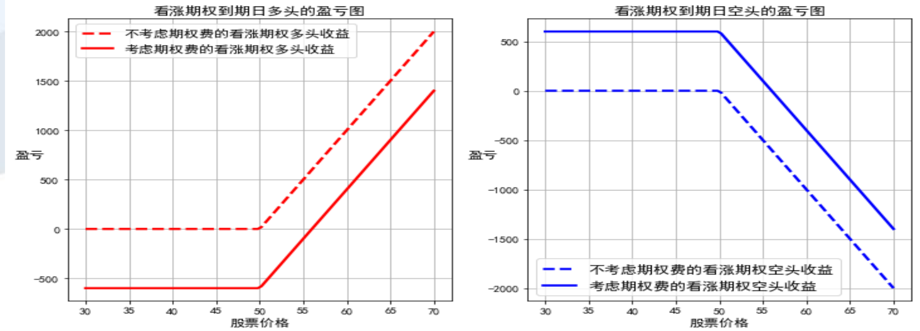
情形2：如果在期权到期日，股票价格大于50元股，期权将会被行使。比如，在期权到期日，股价上涨至60元/股，通过行使期权，A投资者可以按照50元股的执行价格买入100股股票，同时立刻将股票在市场上出售，每股可以获利10元，共计1000元。将最初的期权费考虑在内，A投资者的净盈利为1000-600=400元，这里假定不考虑股票买卖本身的交易费用。

不失一般性，假设K代表期权的执行价格，ST是基础资产在期权合约到期时的价格，在期权到期时，欧式看涨期权多头的盈亏是max(ST-K，0)，空头的盈亏则是-max(ST-K，0)。

如果用C表示看涨期权的期权费，在考虑了期权费以后，在期权到期时，欧式看涨期权多头的盈亏就是max(ST-K-C，-C)，空头的盈亏则是-max(ST-K-C，-C)

10.2.2 看涨期权到期时的盈亏

```
S=np.linspace(30,70,100)            #模拟看涨期权到期时的股价
K=50                                #看涨期权的执行价格
C=6                                 #看涨期权的期权费
call1=100*np.maximum(S-K,0)         #看涨期权到期时不考虑期权费的收益
call2=100*np.maximum(S-K-C,-C)      #看涨期权到期时考虑期权费以后的收益

plt.figure(figsize=(12,6))
p1=plt.subplot(1,2,1)
p1.plot(S,call1,'r--',label=u'不考虑期权费的看涨期权多头收益',lw=2.5)
p1.plot(S,call2,'r-',label=u'考虑期权费的看涨期权多头收益',lw=2.5)
p1.set_xlabel(u'股票价格',fontsize=12)
p1.set_ylabel(u'盈亏',fontsize=12,rotation=0)
p1.set_title(u'看涨期权到期日多头的盈亏图',fontsize=13)
p1.legend(fontsize=12)
p1.grid('True')
p2=plt.subplot(1,2,2)
p2.plot(S,-call1,'b--',label=u'不考虑期权费的看涨期权空头收益',lw=2.5)
p2.plot(S,-call2,'b-',label=u'考虑期权费的看涨期权空头收益',lw=2.5)
p2.set_xlabel(u'股票价格',fontsize=12)
p2.set_ylabel(u'盈亏',fontsize=12,rotation=0)
p2.set_title(u'看涨期权到期日空头的盈亏图',fontsize=13)
p2.legend(fontsize=13)
p2.grid('True')
```



10.2.3 看跌期权到期时的盈亏

【例10-2】假定B投资者买入基础资产为100股Z股票、执行价格为70元/股的欧式看跌期权。股票的当前价格是75元/股，期权到期日是3个月以后，1股股票的看跌期权价格为7元(期权费)，B投资者的最初投资为700元(100×7)，也就是一份看跌期权的期权费700元。同样是分两种情形进行讨论。

情形1：假定在期权到期日，Z股票价格下跌至60元股，B投资者就能以70元股的价格卖出100股股票，因此在不考虑期又费的情况下，B投资者每股盈利为10元，即总收益为1000元;将最初的期权费用700元考虑在内，投资者的净盈利为300元。

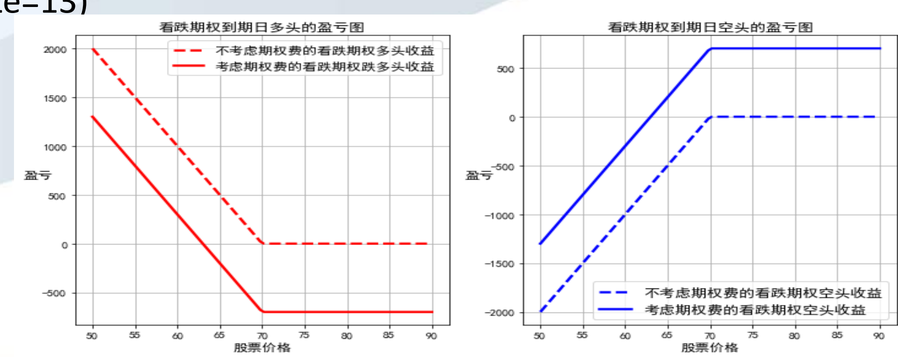
情形2：如果在到期日股票价格高于70元/股，此时看跌期权变得一文不值，B投资者当然也就不会行使期权，损失就是最初的期权费700元。

不失一般性，可以得到在不考虑初始期权费的情况下，欧式看跌期权多头的盈亏max(K-ST，0)，欧式看跌期权空头的盈亏则是-max(K-ST，0)。

如果用P来表示看跌期权的期权费，在考虑了期权费以后，在期权到期时，欧式看跌期权多头的盈亏是max(K-ST-P，-P)，空头的盈亏则是-max(K-ST-P，-P)。式子中的其他符号含义与前面的看涨期权保持一致。

10.2.3 看跌期权到期时的盈亏

```
S=np.linspace(50,90,100)                 #设定看跌期权到期时的股价
K=70                                     #看跌期权的执行价格
P=7                                      #看跌期权的期权费
put1=100*np.maximum(K-S,0)               #看跌期权到期时不考虑期权费的收益
put2=100*np.maximum(K-S-P,-P)      #看跌期权到期时考虑期权费以后的收益

plt.figure(figsize=(12,6))
p3=plt.subplot(1,2,1)
p3.plot(S,put1,'r--',label=u'不考虑期权费的看跌期权多头收益',lw=2.5)
p3.plot(S,put2,'r-',label=u'考虑期权费的看跌期权跌多头收益',lw=2.5)
p3.set_xlabel(u'股票价格',fontsize=12)
p3.set_ylabel(u'盈亏',fontsize=12,rotation=0)
p3.set_title(u'看跌期权到期日多头的盈亏图',fontsize=13)
p3.legend(fontsize=12)
p3.grid('True')
p4=plt.subplot(1,2,2)
p4.plot(S,-put1,'b--',label=u'不考虑期权费的看跌期权空头收益',lw=2.5)
p4.plot(S,-put2,'b-',label=u'考虑期权费的看跌期权空头收益',lw=2.5)
p4.set_xlabel(u'股票价格',fontsize=12)
p4.set_ylabel(u'盈亏',fontsize=12,rotation=0)
p4.set_title(u'看跌期权到期日空头的盈亏图',fontsize=13)
p4.legend(fontsize=13)
p4.grid('True')
```



10.2.3 看跌期权到期时的盈亏

按照基础资产价格与期权执行价格的大小关系，期权可以划分为实值期权（in-the-money option）和虚值期权（out-of-the money option）。表10-2梳理了这3类期权的数学特征，表中的S代表基础资产价格，K代表期权的执行价格。

表10-2

	实值期权	平价期权	虚值期权
看涨期权	S>K	S=K	S<K
看跌期权	S<K	S=K	S>K

10.2.4 看跌-看涨平价关系式

1、两个投资组合

首先，考虑以下两个投资组合在期权合约到期时的盈亏情况。

A投资组合：一份欧式看涨期权和一份在T时刻到期的本金为K的零息债券；

B投资组合：一份欧式看跌期权和一份基础资产。

这里需要假设看涨期权与看跌期权具有相同的执行价格K与相同的合约期限T。

对于A投资组合而言，零息债券在期权合约到期日(T时刻)的价值显然是等于K，而对于看涨期权则分两种情况讨论。

情形1：如果在T时刻，基础资产价格ST>K，A投资组合中的欧式看涨期权将被执行，此时，A投资组合的价值是(ST-K)+K=ST；

情形2：如果在T时刻，基础资产价格ST<K，A投资组合中的欧式看涨期权就没有价值，此时A投资组合的价值为K。

对于B投资组合而言，也分两种情形讨论。

情形1：如果在T时刻，基础资产价格ST>K，此时B投资组合中的欧式看跌期权没有价值，此时B投资组合价值为ST，也就是仅剩下基础资产的价值;

情形2：如果在T时刻，基础资产价格ST<K，此时B投资组合中的欧式看跌期权会被行使，此时B投资组合价值为(K-ST)+ST=K。

综合以上的分析，当ST>K时，在T时刻两个投资组合的价值均为ST；当ST<K时，在T时刻两个投资组合的价值均为K。换言之，在T时刻(期权合约到期时)，两个投资组合的价值均为

10.2.4 看跌-看涨平价关系式

在期权初始日，A投资组合中的看涨期权和零息债券的价值分别表示为c和Ke^{-rT}，B投资组合中的看跌期权和基础资产的价值分别表示为p和S_0，因此

$$c+Ke^{-rT}=p+S_0 \tag{10-1}$$

注意，式子中的r是连续复利的无风险收益率。等式(10-1)就是看跌–看涨平价关系（put-call parity)。

将等式(10-1)略做变换，就可以得到

$$c=p+S_0-Ke^{-rT} \tag{10-2}$$

$$p=c+Ke^{-rT}-S_0 \tag{10-3}$$

下面，运用Python自定义通过看跌-看涨平价关系式计算欧式看涨期权价格、看跌期权价格的函数，具体代码如下：

```
def call_parity(p,S,K,r,T):
    '''通过看跌-看涨平价关系式计算欧式看涨期权的价格
    P，代表欧式看跌期权的价格;
    S，代表期权基础资产的价格;
    K，代表期权的执行价格;
    r，代表无风险收益率;
    T，代表期权合约的剩余期限。'''
    import numpy as np
    return p+S-K*np.exp (-r*T)
```

10.2.4 看跌-看涨平价关系式

```
def put_parity(c, S, K, r, T):
    '''通过看跌-看涨平价关系式计算欧式看跌期权的价格
    c,代表欧式看涨期权的价格;
    S,代表期权基础资产的价格;
    K,代表期权的执行价格;
    r,代表无风险收益率;
    T,代表期权合约的剩余期限。'''
    import numpy as np
    return c+K*np.exp (-r*T)-S
```

10.2.4 看跌-看涨平价关系式

【例10-3】 假设当前股票价格为20元/股,期权的执行价格为18元/股，无风险收益为每年5%，3个月的欧式看涨期权价格对外报价是2.3元，3个月的欧式看跌期权对外报价0.3元，通过看跌-看涨平价关系式判断期权价格的合理性，如果价格不合理则如何实施套利？

下面直接运用前面通过Python自定义的函数call_parity和put_parity，分别计算满足看涨-平价关系式的看涨期权、看跌期权价格，具体代码如下：

```
call=call_parity(p=0.3, S=20,K=18, r=0.05, T=0.25) #计算看涨期权价格
put=put_parity (c=2.3, S=20, K=18, r=0.05, T=0.25) #计算看跌期权价格
print('运用平价关系式得到的看涨期权价格:', round (call, 3))
print('运用平价关系式得到的看跌期权价格:', round (put, 3))
```

运用平价关系式得到的看涨期权价格：2.524
运用平价关系式得到的看跌期权价格：0.076

显然，通过以上的计算，不难发现看涨期权被低估，看跌期权则被高估，因此可以通过持有看涨期权的多头头寸并买入零息债券(相当于买入A投资组合),同时持有看跌期权的空头头寸，并卖空基础资产（相当于卖空B组合），从而实现无风险套利。

10.3 布莱克-斯科尔斯-默顿模型

在20世纪70年代初，费希尔·布莱克(Fisher Black)、迈伦·斯科尔斯(Myron Scholes)和罗伯特·默顿(Robert Merton)在对欧式股票期权定价研究方面取得了重大的理论突破，提出了针对欧式期权定价的模型，该模型被称为布莱克-斯科尔斯-默顿模型(简称BSM模型)。

在风险中性世界中，标的资产的预期收益率为无风险利率 r，股票价格服从的对数正态分布满足:

$$\ln S_T \sim N\left[\ln S_0 + (r - \frac{1}{2}\sigma^2)T, \sigma\sqrt{T}\right]$$

其中 S_T 一到期日股票的价格，S_0 一股票的初始价格，T 一到期日，σ 一股票价格的波动率，r 一无风险收益率，记 $y = \ln S_T, m = \ln S_0 + (r - \frac{1}{2}\sigma^2)T, v = \sigma^2 T$。欧式看涨期权的损益为：

$$C_T = \max(S_T - X, 0)$$

其中 X 为执行价格。则期权到期时的期望损益就为：$E[C_T] = E[\max(S_T - X, 0)]$。进一步计算期望损益 $E[C_T]$，我们有：

$$E[C_T]=E\left[\max(e^{\ln S_T}-X,0)\right]$$

$$=E\left[\max(e^{y}-X,0)\right]$$

$$=\int_{-\infty}^{\infty}\max(e^{y}-X,0)\frac{1}{\sqrt{2\pi\nu}}e^{-\frac{(y-m)^2}{2\nu}}dy$$

$$=\int_{-\infty}^{\ln X}0\frac{1}{\sqrt{2\pi\nu}}e^{-\frac{(y-m)^2}{2\nu}}dy+\int_{\ln X}^{\infty}(e^{y}-X)\frac{1}{\sqrt{2\pi\nu}}e^{-\frac{(y-m)^2}{2\nu}}dy$$

$$=\int_{\ln X}^{\infty}e^{y}\frac{1}{\sqrt{2\pi\nu}}e^{-\frac{(y-m)^2}{2\nu}}dy+\int_{\ln X}^{\infty}(-X)\frac{1}{\sqrt{2\pi\nu}}e^{-\frac{(y-m)^2}{2\nu}}dy$$

继续简化 $E[C_T]$，就有：

$$E[C_T]=\int_{\ln X}^{\infty}e^{y}\frac{1}{\sqrt{2\pi\nu}}e^{-\frac{(y-m)^2}{2\nu}}dy+\int_{\ln X}^{\infty}(-X)\frac{1}{\sqrt{2\pi\nu}}e^{-\frac{(y-m)^2}{2\nu}}dy$$

$$=e^{m+0.5\nu}\int_{\ln X}^{\infty}\frac{1}{\sqrt{2\pi\nu}}e^{-\frac{(y-m-\nu)^2}{2\nu}}dy-X\int_{\ln X}^{\infty}\frac{1}{\sqrt{2\pi\nu}}e^{-\frac{(y-m)^2}{2\nu}}dy$$

$$=e^{m+0.5\nu}\int_{\frac{\ln X-m-\nu}{\sqrt{\nu}}}^{\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}}dz-X\int_{\frac{\ln X-m}{\sqrt{\nu}}}^{\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}dx$$

$$=S_0e^{rT}N\left(-\frac{\ln X-m-\nu}{\sqrt{\nu}}\right)-XN\left(-\frac{\ln X-m}{\sqrt{\nu}}\right)$$

注意上式中的 $N(-\frac{\ln X-m-\nu}{\sqrt{\nu}})$ 和 $N(-\frac{\ln X-m}{\sqrt{\nu}})$，假设记 $d_1=-\frac{\ln X-m-\nu}{\sqrt{\nu}}$，

$d_2=-\frac{\ln X-m}{\sqrt{\nu}}$，将 m 和 ν 代入 d_1 和 d_2 中得到：

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

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

则 $E[C_T]$ 的最终结果就为：

$$E[C_T]=S_0e^{rT}N(d_1)-XN(d_2) \tag{1}$$

最后将（1）式用无风险利率 r 贴现后得到：

$$C_0=S_0N(d_1)-Xe^{-rT}N(d_2) \tag{2}$$

式子（2）就为不支付红利的股票的欧式看涨期权 0 时刻的价格。有根据欧式看涨期权价格和看跌期权价格之间的平价公式：$c+Xe^{-rT}=p+S_0$，就可以求出欧式看跌期权的价格，即：

$$P_0=Xe^{-rT}N(-d_2)-S_0N(-d_1) \tag{3}$$

下面就直接给出欧式看涨期权的定价数学表达式：

欧式看涨期权的定价公式：

$$c=S_0N(d_1)-Ke^{-rT}N(d_2) \qquad (10\text{-}5)$$

通过上节讨论的看跌-看涨评价关系式，可得到欧式看跌期权的定价公式：

$$p=ke^{-rT}N(-d_2)-S_0N(-d) \qquad (10\text{-}6)$$

其中：

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

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

10.3.1 BSM模型

通过Python自定义基于布莱克-斯科尔斯-默顿模型计算欧式看涨期权、看跌期权定价的函数，具体如下:

```
def call_BS (S, K,sigma, r,T):
    '''运用布莱克-斯科尔斯-默顿定价模型计算欧式看涨期权价格
    S:代表期权基础资产的价格;
    K:代表期权的执行价格;
    sigma:代表基础资产价格百分比变化的年化波动率
    r:代表无风险收益率;
    T:代表期权合约的剩余期限。'''

    import numpy as np
    from scipy.stats import norm #从SciPy的子模块stats中导入norm函数
    d1=(np.log(S/K)+ (r+pow(sigma,2)/2)*T)/(sigma*np.sqrt (T))
    d2=d1-sigma*np.sqrt (T)
    return S*norm.cdf(d1)-K*np.exp(-r*T) *norm.cdf (d2)
```

```
def put_BS (S, K, sigma, r, T):
    '''11运用布莱克-斯科尔斯-默顿定价模型计算欧式看跌期权价格
    S:代表期权基础资产的价格;
    K:代表期权的执行价格;
    sigma:代表基础资产价格百分比变化的年化波动率
    r:代表无风险收益率;
    T:代表期权合约的剩余期限。'''

    import numpy as np
    from scipy.stats import norm #从Scipy的子模块stats中导入norm函数
    d1=(np.log(S/K)+(r+pow(sigma, 2)/2)*T)/(sigma*np.sqrt(T))
    d2=d1-sigma*np.sqrt(T)
    return K*np.exp(-r*T)*norm.cdf(-d2)-S*norm.cdf(-d1)
```

【例10-4】考虑一份期限为6个月的股票期权，期权的基础资产是工商银行的A股股票， 2018年12月28日股票收盘价是5.29元股,期权的执行价格为6元股，无风险利年化4%，股票收益率的年化波动率是24%，运用布莱克-斯科尔斯-默顿模型计算看涨期看跌期权的价格。

下面，运用前面Python定义的计算欧式看涨期权价格、看跌期权价格的函数call_BS和put_BS对例10-4进行快速求解,具体的代码如下：

```
call=call_BS(S=5.29, K=6, sigma=0.24,r=0.04,T=0.5)
put=put_BS(S=5.29,K=6,sigma=0.24, r=0.04,T=0.5)
print ('根据布莱克-斯科外斯-默顿模型计算的看涨期权价格:', round(call, 4))
print ('根据布莱克-斯科尔斯-默顿模型计算的看跌期权价格:', round (put,4))
```

根据布莱克-斯科外斯-默顿模型计算的看涨期权价格: 0.1532
根据布莱克-斯科尔斯-默顿模型计算的看跌期权价格: 0.7443

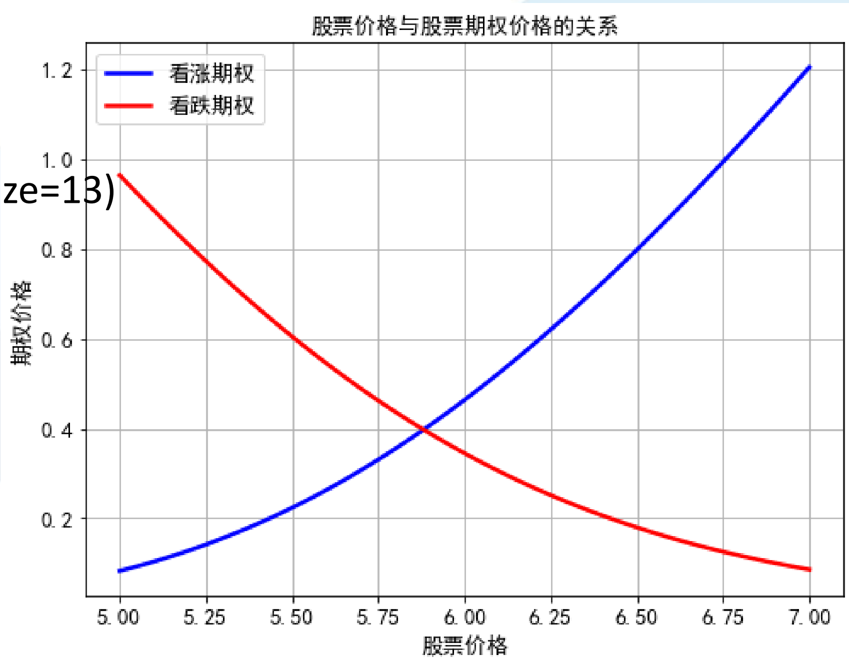
10.4 期权价格决定因素

通过布莱克-斯科尔斯-默顿模型，不难发现有5个变量会影响期权的价格：
一是当前基础资产价格S_0，
二是期权的执行价格K，
三是期权期限T，
四是基础资产的波动率δ，
五是无风险收益率r。

10.4.1 期权价格与基础资产价格

【例10-5】沿用前面例10-4的工商银行股票期权信息，对股票价格设定一个取值是在区间[5,7]的等差数列，其他的变量取值保持不变，运用BSM模型对期 权进行定价，从而模拟期权价格与基础资产价格变动之间的关系（**非线性+**），具体的代码如下：

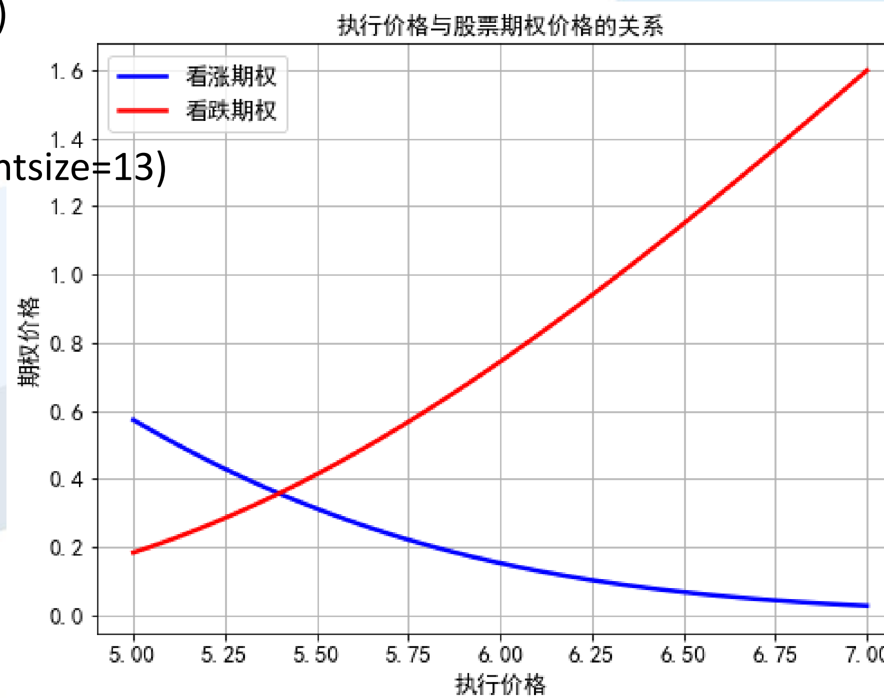
```
S_list=np.linspace (5.0,7.0,100)  #生成基础资产价格的数组
call_list1 = call_BS(S=S_list,K=6, sigma=0.24, r=0.04, T=0.5) #计算看涨期权的价格
put_list1 = put_BS(S=S_list,K=6, sigma=0.24, r=0.04,T=0.5) #计算看跌期权的价格
plt.figure(figsize=(8,6) )
plt.plot(S_list,call_list1,'b-',label=u'看涨期权',lw=2.5)
plt.plot (S_list,put_list1,'r-',label=u'看跌期权',lw=2.5)
plt.xlabel (u'股票价格',fontsize=13)
plt.ylabel (u'期权价格',fontsize=13, rotation=90)
plt.xticks (fontsize=13)
plt.yticks (fontsize=13)
plt.title (u'股票价格与股票期权价格的关系', fontsize=13)
plt.legend (fontsize=13)
plt.grid ('True')
plt. show()
```



10.4.2 期权价格与执行价格

【例10-6】沿用前面例10-4的工商银行股票期权信息，对期权的执行价格设定一个取值是在区间[5，7]的等差数列，其他的变量取值保持不变，模拟期权价格与执行价格变动之间的关系(**非线性-**)，具体的代码如下：

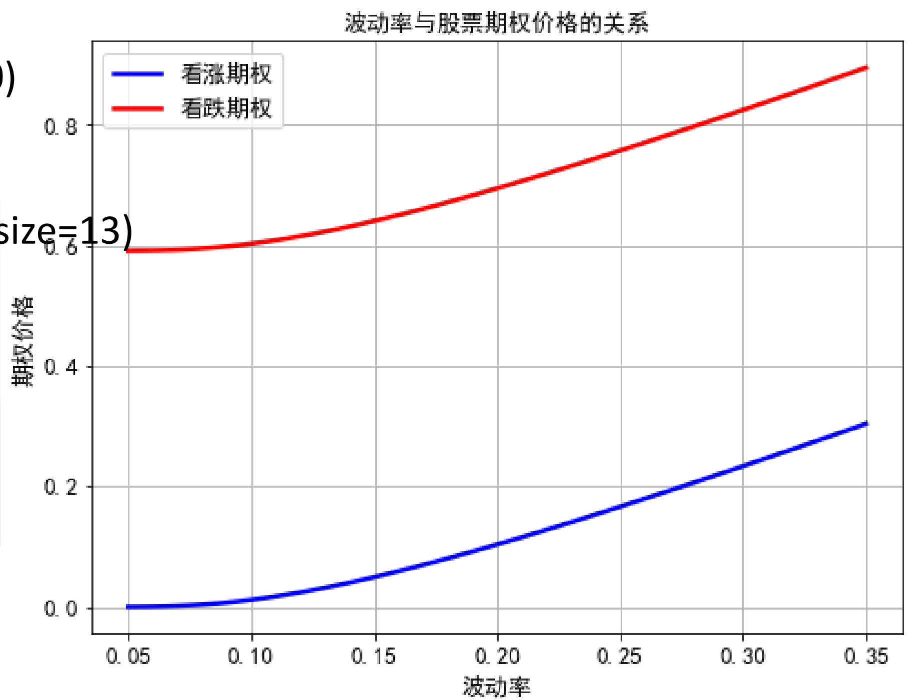
```
K_list=np.linspace(5.0,7.0,100)  #生成期权执行价格的数组
call_list2 = call_BS(S=5.29, K=K_list, sigma=0.24, r=0.04,T=0.5)
put_list2 = put_BS(S=5.29, K=K_list, sigma=0.24,r=0.04,T=0.5)
plt.figure(figsize=(8,6))
plt.plot (K_list,call_list2, 'b-', label=u'看涨期权',lw=2.5)
plt.plot (K_list,put_list2, 'r-',label=u'看跌期权',lw=2.5)
plt.xlabel (u'执行价格', fontsize=13)
plt.ylabel (u'期权价格' , fontsize=13, rotation=90)
plt.xticks (fontsize=13)
plt.yticks (fontsize=13)
plt.title (u'执行价格与股票期权价格的关系', fontsize=13)
plt.legend (fontsize=13)
plt.grid ('True')
plt.show ()
```



10.4.3 期权价格与波动率

【例10-7】沿用前面例10-4的工商银行股票期权信息，针对基础资产（股票）的波动率设定一个取值是在区间[0.05，0.35]的等差数列，其他的变量取值保持不变，模拟期权价格与波动率之间的关系(**非线性+**)，具体的代码如下：

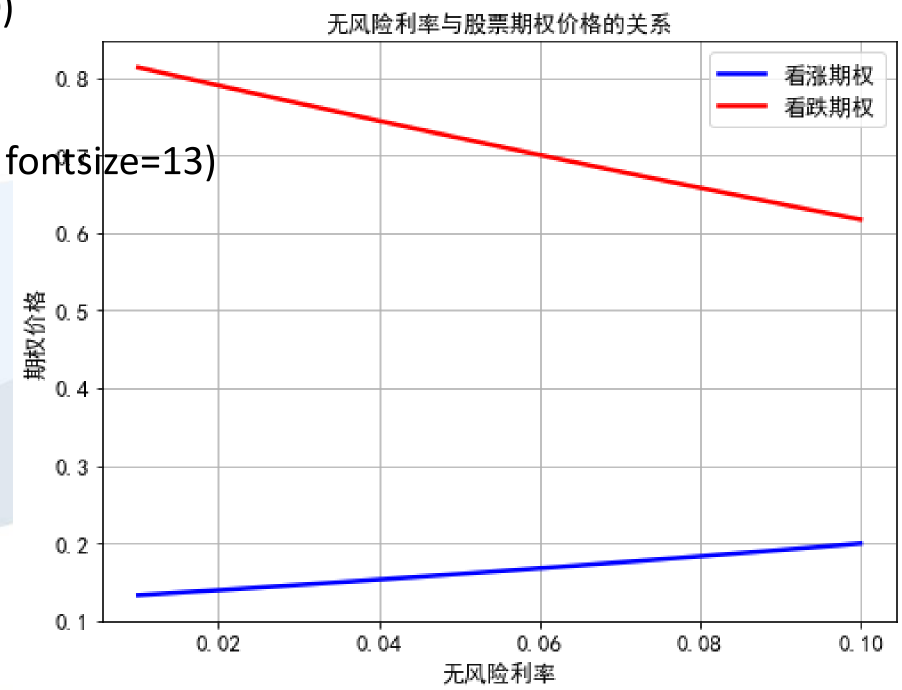
```
sigma_list=np.linspace (0.05,0.35, 100)  #生成波动率的数组
call_list3 = call_BS(S=5.29, K=6.0, sigma=sigma_list,r=0.04,T=0.5)
put_list3 = put_BS(S=5.29, K=6.0, sigma=sigma_list,r=0.04,T=0.5)

plt.figure(figsize=(8,6))
plt.plot (sigma_list,call_list3, 'b-', label=u'看涨期权',lw=2.5)
plt.plot (sigma_list,put_list3, 'r-',label=u'看跌期权',lw=2.5)
plt.xlabel (u'波动率', fontsize=13)
plt.ylabel (u'期权价格' , fontsize=13, rotation=90)
plt.xticks (fontsize=13)
plt.yticks (fontsize=13)
plt.title (u'波动率与股票期权价格的关系', fontsize=13)
plt.legend (fontsize=13)
plt.grid ('True')
plt.show ()
```



10.4.4 期权价格与无风险收益率

【例10-8】沿用前面例10-4的工商银行股票期权信息，对无风险收益率设定一个取值是在区间[0.01，0.1]的等差数列，其他的变量取值保持不变，模拟无风险收益率与期权价格之间的关系(**线性+ 执行价折现&杠杆作用**)，具体的代码如下：

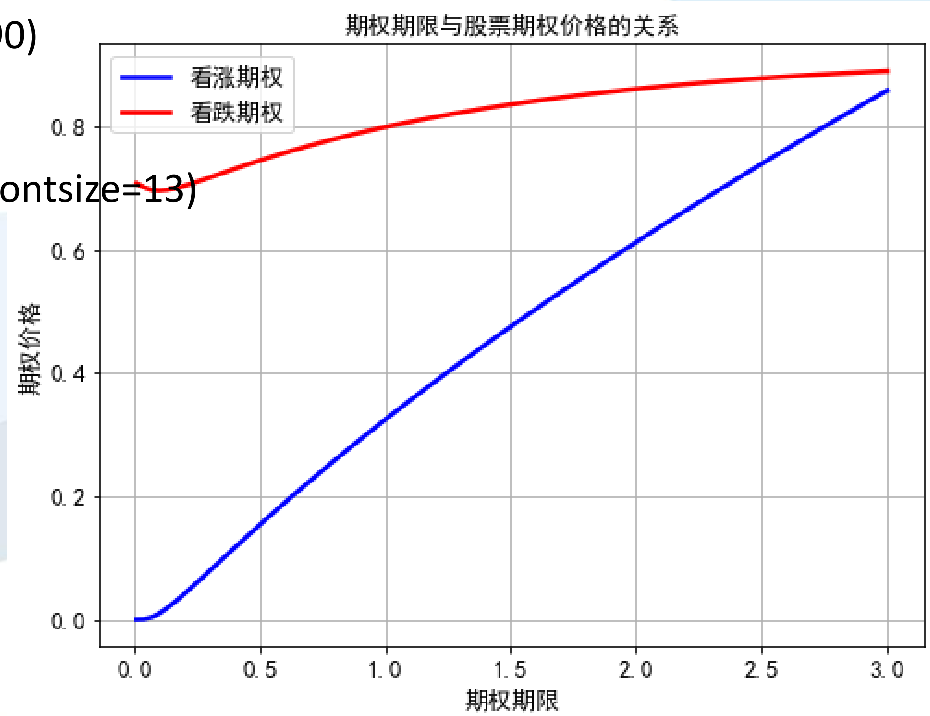
```
r_list=np.linspace(0.01, 0.10, 100)  #生成无风险收益率的数组
call_list4=call_BS(S=5.29, K=6.0,sigma=0.24,r=r_list,T=0.5)
put_list4=put_BS(S=5.29, K=6.0,sigma=0.24,r=r_list,T=0.5)
plt.figure(figsize=(8,6))
plt.plot(r_list,call_list4, 'b-', label=u'看涨期权',lw=2.5)
plt.plot(r_list,put_list4, 'r-',label=u'看跌期权',lw=2.5)
plt.xlabel (u'无风险利率', fontsize=13)
plt.ylabel (u'期权价格' , fontsize=13, rotation=90)
plt.xticks (fontsize=13)
plt.yticks (fontsize=13)
plt.title (u'无风险利率与股票期权价格的关系', fontsize=13)
plt.legend (fontsize=13)
plt.grid ('True')
plt.show ()
```



10.4.5 期权价格与期权期限

【例10-9】沿用前面例10-4的工商银行股票期权信息，对期权的期限设定一个取值是在区间[0.01，3]的等差数列，其他的参数保持不变，模拟期权期限与期权价格之间的关系(**非线性+**)，具体的代码如下：

```
T_list=np.linspace(0.01, 3.0, 100)  #生成无风险收益率的数组
call_list5=call_BS(S=5.29, K=6.0,sigma=0.24,r=0.04,T=T_list)
put_list5=put_BS(S=5.29, K=6.0,sigma=0.24,r=0.04,T=T_list)
plt.figure(figsize=(8,6))
plt.plot(T_list,call_list5, 'b-', label=u'看涨期权',lw=2.5)
plt.plot(T_list,put_list5, 'r-',label=u'看跌期权',lw=2.5)
plt.xlabel (u'期权期限', fontsize=13)
plt.ylabel (u'期权价格' , fontsize=13, rotation=90)
plt.xticks (fontsize=13)
plt.yticks (fontsize=13)
plt.title (u'期权期限与股票期权价格的关系', fontsize=13)
plt.legend (fontsize=13)
plt.grid ('True')
plt.show ()
```



10.5 期权的风险度量：希腊字母

10.5.1 期权的Delta

期权的Delta(Δ)被定义为期权价格变动与基础资产价格变动的比率，也就是期权价格与基础资产价格之间关系曲线的切线斜率，即

$$\varDelta = \frac{\partial \Pi}{\partial S} \qquad (10\text{-}7)$$

其中，∏表示期权的价格，S表示基础资产的价格。

比如，期权Delta值等于0.6就意味着当基础资产价格变化一个很小的金额时，相应的期权价格变化约等于基础资产价格变化的60%。

根据BS模型，对于欧式期权的Delta值，具体的数学表达式如表：

期权类型	方向	Delta的表达式
欧式看涨期权	多头	$\varDelta = N(d_1)$
	空头	$\varDelta = -N(d_1)$
欧式看跌期权	多头	$\varDelta = N(d_1) - 1$
	空头	$\varDelta = 1 - N(d_1)$

10.5.1 期权的Delta

下面，通过Python自定义计算欧式期权Delta的函数，具体代码如下：

```
def delta_option(S, K, sigma, r,T, optype, positype):
    '''计算欧式期权的Delta值
    S:代表期权基础资产的价格;
    K:代表期权的执行价格;
    sigma:代表基础资产价格百分比变化的波动率;
    r:代表无风险收益率;
    T：代表期权合约的剩余期限;
    optype:代表期权类型,输入'call'表示看涨期权,输入'put'表示看跌期权;
    positype:代表期权头寸方向,输入'long'表示多头,输入'short'表示空头。'''
    import numpy as np
    from scipy.stats import norm #从sciPy的子模块stats中导入norm函数
    dl = (np.log(S/K)+(r+pow(sigma,2)/2)*T)/( sigma*np.sqrt(T) )
    if optype=='call':
        if positype=='long':
            delta=norm.cdf (dl)
        else:
            delta=-norm.cdf(dl)
    else:
        if positype=='long':
            delta=norm.cdf(dl)-1
        else:
            delta=1-norm.cdf(dl)
    return delta
```

10.5.1 期权的Delta

【例10-10】沿用前面例10-4的工商银行股票期权，假定股票的当前价格是5元/股，其他的参数均不变，运用前面Python定义的计算期权Delta值的函数delta_option，分别计算看涨、看跌期权的多头与空头的Delta值。具体的代码如下：

```
delta1 = delta_option(S=5, K=6, sigma=0.24,r=0.04, T=0.5, optype='call', positype='long')
delta2 = delta_option(S=5,K=6,sigma=0.24, r=0.04,T=0.5, optype='call',positype='short')
delta3=delta_option(S=5, K=6, sigma=0.24, r=0.04,T=0.5, optype='put',positype='long')
delta4=delta_option(S=5, K=6,sigma=0.24, r=0.04,T=0.5, optype='put',positype='short')
print ('看涨期权多头的Delta值: ',round (delta1, 4) )
print ('看涨期权空头的Delta值: ',round(delta2,4) )
print ('看跌期权多头的Delta值: ',round (delta3, 4) )
print ('看跌期权空头的Delta值: ',round (delta4, 4) )
```

看涨期权多头的Delta值：0.1917
看涨期权空头的Delta值：-0.1917
看跌期权多头的Delta值：-0.8083
看跌期权空头的Delta值：0.8083

10.5.2 期权的Gamma

期权的Gamma(Γ)是指期权Delta直的变化与基础资产价格变化的比率。Gamma是期权价值关于基础资产价格的二阶偏导数：

$$\Gamma=\frac{\partial^2 \Pi}{\partial S^2} \quad (10\text{-}8)$$

其中，Π依然表示期权的价值，S依然表示期权基础资产价格。

比如，期权Gamma等于0.3，这就意味着当基础资产价格变化时，相应期权Delta值的变化约等于基础资产价格变化的30%。

根据BS模型，对于欧式看涨和欧式看跌期权的Gamma值的表达式均是：

$$\Gamma=\frac{N'(d_1)}{S_0\sigma\sqrt{T}} \quad (10\text{-}9)$$

其中，

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

$$N'(x)=\frac{1}{2\sqrt{\pi}}e^{-x^2/2}$$

为了便于建模，以上式子（10-9）经整理后得到Gamma的表达式

$$\Gamma=\frac{N'(d_1)}{S_0\sigma\sqrt{T}}=\frac{1}{S_0\sigma\sqrt{2\pi T}}e^{-{d_1}^2/2} \quad (10\text{-}10)$$

10.5.2 期权的Gamma

接着通过Python自定义计算欧式期权Gamma的函数，具体的代码如下:

```
def gamma_option (S, K, sigma,r,T):
    '''计算欧式期权的gamma 值
    S:代表期权基础资产的价格；
    K：代表期权的执行价格；
    sigma:代表基础资产价格百分比变化的波动率；
    r：代表无风险收益率；
    T:代表期权合约的剩余期限。'''
    import numpy as np
    from scipy.stats import norm #从Scipy的子模块stats中导入norm函数
    d1=(np.log(S/K)+(r+pow(sigma,2)/2)*T)/(sigma*np.sqrt(T))
    return np.exp(-pow(d1,2)/2)/(S*sigma*np.sqrt(2*np.pi*T))
```

【例10-13】沿用前面例10-4的工商银行股票期权信息，同时假定股票的当前价格是5元/股，其他的参数均不变，运用前面Python定义的计算期权Gamma值的函数：

```
gamma=gamma_option (S=5, K=6, sigma=0.24, r=0.04, T=0.5)
print ('计算得到的欧式期权Gamma值: ', round (gamma, 4))
```

计算得到的欧式期权Gamma值：0.3216

10.5.3 期权的Theta

期权的Theta(Θ)定义为在其他条件不变时，期权价值变化与时间变化的比率，即

$$\Theta=\frac{\partial \Pi}{\partial T} \qquad (10\text{-}11)$$

其中，Π依然表示期权的价格，T表示期权的期限。

Theta有时也被称为期权的时间损耗(time decay)。在其他条件不变的情况下，不论是看涨期权还是看跌期权，通常距离期权到期日越远，期权价值是越高的;距离期权到期日越近，期权价值则是越低的。

对于一个欧式看涨期权，计算Theta的公式可以通过BSM模型得出，并且看涨期权与看跌期权是存在差异的，对于看涨期权的Theta，计算公式如下:

$$\Theta_{\text{call}}=\frac{S_0 N'(d_1)\sigma}{2\sqrt{T}}-rKe^{-rT}N(d_2) \qquad (10\text{-}12)$$

其中，

$$d_1=\frac{ln(S_0/K)+\left(r+\sigma^2/2\right)T}{\sigma\sqrt{T}}$$

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

$$N'(x)=\frac{1}{2\sqrt{\pi}}e^{-x^2/2}$$

对于一个欧式股票看跌期权，Theta的计算公式则是

$$\Theta_{\text{put}}=-\frac{S_0 N'(d_1)\sigma}{2\sqrt{T}}+rK^{-rT}N(-d_2)=\Theta_{call}+Ke^{-rT} \qquad （10\text{-}14）$$

10.5.3 期权的Theta

下面，通过Python构建计算欧式期权Theta值的函数，具体如下：

```
def theta_option (S, K, sigma, r,T,optype):
   '''计算欧式期权的Theta值
   S:代表期权基础资产的价格;
   K:代表期权的执行价格;
   sigma:代表基础资产价格百分比变化的波动率;
   r:代表无风险收益率;
   T:代表期权合约的剩余期限;
   optype:代表期权的类型,输入'call 表示看涨期权,输入'put'表示看跌期权。'''
   import numpy as np
   from scipy.stats import norm #从SciPy的子模块stats中导入norm函数
   d1=(np.log(S/K)+ (r+pow(sigma,2)/2)*T)/(sigma*np.sqrt(T))
   d2=d1-sigma*np.sqrt(T)
   theta_call=-(S*sigma*np.exp(-pow(d1,2)/2))/(2*np.sqrt(2*np.pi*T))-r*K*np.exp(-
r*T)*norm.cdf(d2)
   if optype=='call':
      theta=theta_call
   else:
      theta=theta_call+r*K*np.exp(-r*T)
    return theta
```

10.5.3 期权的Theta

【例10-16】沿用前面例10-4的工商银行股票期权信息，假定股票的当前价格是5元/股，其他的参数均不变，运用前面Python定义的计算期权Theta的函数theta_option，分别计算看涨、看跌期权的Theta值，具体的代码：

```
theta_call=theta_option (S=5, K=6, sigma=0.24, r=0.04, T=0.5, optype='call')
theta_put=theta_option (S=5, K=6, sigma=0.24, r=0.04,T=0.5, optype='put')
print ('计算得到的欧式看涨期权Theta值', round(theta_call, 6) )
print ('计算得到的欧式看涨期权每日历天Theta值', round(theta_call/365, 6))
print('计算得到的欧式看涨期权每交易日Theta值', round(theta_call/252, 6))
print('计算得到的欧式看跌期权Theta值', round(theta_put, 6) )
print('计算得到的欧式看跌期权每日历天Theta值', round (theta_put/365, 6))
print('计算得到的欧式看跌期权每交易日Theta值', round (theta_put/252,6) )
```

计算得到的欧式看涨期权Theta值 -0.266544
计算得到的欧式看涨期权每日历天Theta值 -0.00073
计算得到的欧式看涨期权每交易日Theta值 -0.001058
计算得到的欧式看跌期权Theta值 -0.031296
计算得到的欧式看跌期权每日历天Theta值 -8.6e-05
计算得到的欧式看跌期权每交易日Theta值 -0.000124

10.5.4 期权的Vega

期权的Vega(V)是指期权价值变化与基础资产波动率变化的比率，即

$$V=\frac{\partial \Pi}{\partial \sigma} \tag{10-15}$$

其中，Π依然表示期权的价格，σ表示基础资产的波动率。

如果一个期权的Vega绝对值很大，该期权的价值会对基础资产波动率的变化非常敏感。相反，当一个期权的Vega接近零时，基础资产波动率的变化对期权价值的影响则会很小。

欧式看涨期权或看跌期权的Vega均由以下公式给出

$$v=S_0\sqrt{T}N'(d_1) \tag{10-16}$$

其中，

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

$$N'(x)=\frac{1}{d\sqrt{2\pi}}e^{-x^2/2}$$

为了便于建模，将式子(10-16)进行整理以后得到:

$$v=\frac{S_0\sqrt{T}e^{-d_1^2/2}}{\sqrt{2\pi}} \quad （10\text{-}17）$$

10.5.4 期权的Vega

通过Python自定义计算期权Vega值的函数，具体代码如下：

```
def vega_option(S, K,sigma, r,T):
   '''计算欧式期权的Vega值
   s:代表期权基础资产的价格
   K:代表期权的执行价格,
   sigma:代表基础资产价格百分比变化的波动率;
   r:代表无风险收益率
   T:代表期权合约的剩余期限。'''
   import numpy as np
   dl=(np.log (S/K)+(r+pow(sigma,2)/2)*T)/(sigma*np.sqrt(T))
   return S*np.sqrt (T)*np.exp(-pow(dl,2)/2)/np.sqrt (2*np.pi)
```

【例10-19】沿用前面例10-4的工商银于股票期权信息，假定股票的当前价格是58元/股，其他的参数均不变，运用前面Python定义的计算期权Vega值的函数vega_option，求解该股票期权的Vega值以及当波动率增1%时期权价格的变动情况，具体的代码如下：

```
vega=vega_option (S=5.8, K=6, sigma=0.24, r=0.04, T=0.5)
print ('计算得到期权的Vega值:',round (vega, 4))
print ('波动率增加1%时期权价格的变动:', round (vega*0.01,4))
```

计算得到期权的Vega值: 1.6361
波动率增加1%时期权价格的变动: 0.0164

10.5.5 期权的Rho

期权的Rho表示期权价值变化与无风险收益率变化的比率，即

$$Rho = \frac{\partial \Pi}{\partial r} \qquad (10\text{-}18)$$

其中，Π依然表示期权的价格，r表示无风险收益率。

希腊字母Rho用于衡量当其他变量保持不时，期权价值对于无风险收益率变化的敏感性，具体是指当无风险收益率变化1%(比如从3%上升至4%)导致期权价值变化0.01Rho。

对于一个欧式看涨期权，期权Rho由以下的公式给出:

$$\text{Rho}=KTe^{-rT}N(d_2) \geq 0 \qquad (10\text{-}19)$$

其中，

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

对于一个欧式看跌期权，Rho 则由以下的公式所给出

$$\text{Rho}=-KTe^{-rT}N(-d_2) \leqslant 0 \qquad (10\text{-}20)$$

10.5.5 期权的Rho

下面，通过Python构建计算欧式期权rho值的函数，具体的代码如下:

```
def rho_option (S,K, sigma, r,T , optype) :
  '''计算欧式期权的Rho值
  S:代表期权基础资产的价格；
  K:代表期权的执行价格;
  sigma:代表基础资产价格百分比变化的波动率;
  r:代表无风险收益率；
  T:代表期权合约的剩余期限;
  optype:代表期权的类型,输入'call'表示看涨期权,输入'put'表示看跌期权。'''
  import numpy as np
  from scipy.stats import norm #从SciPy的子模块stats 中导入norm
  d1=(np.log (S/K)+(r+pow(sigma,2)/2) *T)/(sigma*np.sqrt (T) )
  d2=d1-sigma*np.sqrt (T)
  if optype=='call':
    rho=K*T*np.exp (-r*T) *norm.cdf (d2)
  else:
    rho=-K*T*np.exp(-r*T)*norm.cdf (-d2)
  return rho
```

10.5.5 期权的Rho

【例10-22】沿用前面例10-4的工商银行股票期权信息，假定工商银行股票的当前价格是5元/股，其他的参数均不变，运用前面Python定义的计算期权Rho的函数rho_option，分别求出看涨、看跌期权的Rho值，具体的代码如下：

```
rho_call=rho_option (S=5, K=6, sigma=0.24, r=0.04, T=0.5, optype='call')
rho_put=rho_option (S=5,K=6, sigma=0.24,r=0.04, T=0.5, optype='put')
print ('计算得到的看涨期权的Rho值: ', round (rho_call,4))
print ('计算得到的看跌期权的Rho值: ',round (rho_put, 4))
print('当无风险利率增加1%时着涨期权价值的变化: ',round(rho_call*0.01,4))
print('当无风险利率增加1%时看跌期权价值的变化: ',round(rho_put*0.01,4))
```

```
计算得到的看涨期权的Rho值：0.4377
计算得到的看跌期权的Rho值：-2.5029
当无风险利率增加1%时着涨期权价值的变化：0.0044
当无风险利率增加1%时看跌期权价值的变化：-0.025
```

从以上的计算结果可以得到，当无风险收益率增加1%时(即由4%增长到5%)，看涨期权价格会相应增加0.0044元，而看跌期权的价格则下降0.025元。

10.6 期权的隐含波动率

10.6.1 运用牛顿迭代法计算隐含波动率

在BSM模型中，可以直接观察到基础资产的当前价格S_0、期权的执行价格K、期权合约期限T以及无风险收益率r，唯一不能直接观察到的变量就是基础资产的波动率σ。（当然，可以通过基础资产的历史价格来估计波动率，可参见6.3节）

但在实践中，通常会使用所谓的隐含波动率(implied volatility)，该波动率是指通过期权的市场价格、运用BSM模型计算得到的波动率。

但比较棘手的问题是，无法直接通过反解10.3节的看涨期权定价式子(10-5)或看跌期权定价式子(10-6)将σ表示为变量c(或p)、S_0、K、r、T的函数，只能运用迭代方法求解出隐含的σ值。

常用的迭代方法包括牛顿迭代法和二分查找法。

10.6.1 运用牛顿迭代法计算隐含波动率

牛顿迭代法(Newton's Method)，也称为牛顿-拉弗森方法(Newton-Raphson method).

在利用该方法计算期权的隐含波动率时，需要做好以下3个方面的工作：
1、需要输入一个初始的隐含波动率；
2、建立一种迭代关系式，如果由初始的隐含波动率得到的期权价格高于市场价格，则需要减去一个标量(比如0.0001)，相反则加上一个标量；
3、需要对迭代过程进行控制，也就是针对隐含波动率得到的期权价格与期权的市场价格之间的差额设置一个可接受的临界值。

下面，就利用牛顿迭代法并运用Python自定义分别计算欧式看涨、看跌期权隐含波动率的函数，具体代码如下：

10.6.1 运用牛顿迭代法计算隐含波动率

```
def impvol_call_Newton( C, S, K, r,T):
  '''运用布莱克-斯科尔斯-默顿定价模型计算看涨期权的隐含波动率,并且使用的方法是牛顿迭代法
  C:代表看涨期权的市场价格
  S:代表期权基础资产的价格
  K:代表期权的执行价格,
  r:代表无风险收益率
  T:代表期权合约的剩余期限。'''
  def call_BS(S,K, sigma, r, T):
    import numpy as np
    from scipy.stats import norm #从SciPy子模块stats中导入norm
    d1=(np.log (S/K) + (r+pow(sigma, 2)/2)*T)/(sigma*np.sqrt(T))
    d2=d1-sigma*np.sqrt(T)
    return S*norm.cdf(d1)-K*np.exp(-r*T)*norm.cdf (d2)
  sigma0=0.2
  diff=C-call_BS(S, K, sigma0, r, T)
  i=0.0001  #设置一个标量
  while abs(diff)>0.0001:  #用while语句
    diff=C-call_BS(S,K, sigma0, r, T)
    if diff>0:
      sigma0 +=i
    else:
      sigma0 -=i
  return sigma0
```

10.6.1 运用牛顿迭代法计算隐含波动率

```
def impvol_put_Newton (P,S, K,r, T):
  '''运用布莱克-斯科尔斯一默顿定价模型计算看跌期权的隐含波动率,并且使用的方法是牛顿送代法
  P:看跌期权的市场价格;
  s:代表期权基础资产的价格;
  K:代表期权的执行价格;
  r:代表无风险收益率;
  T:代表期权合约的剩余期限。'''
  def put_BS(S,K , sigma, r,T):
    import numpy as np
    from scipy.stats import norm
    d1=(np.log (S/K)+(r+pow(sigma, 2)/2) *T)/(sigma*np.sqrt (T))
    d2=d1-sigma*np.sqrt(T)
    return K*np.exp(-r*T) *norm.cdf (-d2)-S*norm.cdf (-d1)
  sigma0=0.2
  diff=P-put_BS (S, K,sigma0,r,T)
  i=0.0001
  while abs(diff)>0.0001:
    diff=P-put_BS(S, K, sigma0, r, T)
    if diff>0:
      sigma0 +=i
    else:
      sigma0 -=i
return sigma0
```

10.6.1 运用牛顿迭代法计算隐含波动率

【例10-25】依然沿用前面例10-4的工商银行股票期权作为分析对象，同时假定看涨期权的市场价格为0.1566元、看跌期权的市场价格是0.7503元，其他的参数均不变，通过前面Python定义的运用牛顿迭代去计算隐含波动率的函数impvol_call_Newton和impvol_put_Newton，求解期权的隐含波动率，具体的代码如下：

```
imp_vol1=impvol_call_Newton (C=0.1566, S=5.29, K=6,r=0.04, T=0.5)
imp_vol2=impvol_put_Newton (P=0.7503, S=5.29, K=6, r=0.04, T=0.5)
print ("用牛顿送代法计算得到看涨期权隐含波动率: ", round (imp_vol1,4))
print('用牛顿选代法计算得到看跌期权隐含波动率: ', round (imp_vol2, 4))
```

用牛顿送代法计算得到看涨期权隐含波动率：0.2427
用牛顿选代法计算得到看跌期权隐含波动率：0.2445

通过以上的计算结果可以得到看涨期权的隐含波动率是24.27%，看跌期权的隐含波动率是24.45%。

需要注意的是，由于计算的步骤会比较多，因此牛顿迭代法的效率往往是比较低的，如果将结果的精确度进一步提高，则需要花费比较长的时间进行运算。

10.6.2 运用二分查找法计算隐含波动率

为了提高运算速度，可以采用二分查找法(binary search，也称“折半查找法”)作为迭代的方法，通过举一个简单的例子就能够好地理解这种方法。

假定沿用前面例10-25的信息，初始猜测的波动率是20%对应该波动率数值估计得到的欧式看涨期权价格是0.1035元，显然，比市场价格0.1566元更小。由于期权价格是波动率的增函数，因此合理地估计正确的波动率应该会比20%更大。然后假定皮动率是30%，对应的期权价格0.2336元，这个结果又比0.1566元高，则可以肯定波动率是介于20%~30%的区间中。接下来，取上两次波动率数值的均值，也就是波动率25%，对应的期权价格为0.1662元，这个值又比0.1566元高，但是合理的波动率所处的区间范围收窄至20%与25%之间，然后取均值22.5%继续计算，每次迭代都使波动率所处的区间减半，最终就可以计算出满足较高精确度的隐含波动率近似值。

下面，就利用二分查找法并运用Python分别计算欧式看涨、看跌期权隐含波动率的自定义函数，具体的代码如下：

10.6.2 运用二分查找法计算隐含波动率

```
def impvol_call_Binary (C,S, K, r,T):
  '''运用BSM定价模型计算看涨期权的隐含波动率,并且使用的迭代方法是二分查找法
  C:代表看涨期权的市场价格
  S:代表期权基础资产的价格;
  K:代表期权的执行价格;
  r:代表无风险收益率;
  T:代表期权合约的剩余期限。'''
  def call_BS(S, K, sigma, r,T):
    import numpy as np
    from scipy.stats import norm #从sciFy的子模块stats中导入norm函数
    d1=(np.log (S/K)+(r+pow (sigma, 2)/2)*T)/(sigma*np.sqrt (T) )
    d2=d1-sigma*np.sqrt (T)
    return S*norm.cdf(d1)-K*np.exp (-r*T) *norm.cdf (d2)
  sigma_min=0.001   #设定波动率的初始最小值
  sigma_max=1.000   #设定波动率的初始最大值
  sigma_mid=(sigma_min+sigma_max)/2
  call_min=call_BS(S,K, sigma_min,r, T)
  call_max=call_BS(S, K, sigma_max,r, T)
  call_mid=call_BS(S, K, sigma_mid,r, T)
  diff=C-call_mid
  if C<call_min or C>call_max:
    print ('Error')
  while abs (diff) >1e-6:
    diff=C-call_BS(S, K,sigma_mid,r, T)
    sigma_mid=(sigma_min+sigma_max)/2
    call_mid=call_BS(S, K, sigma_mid, r, T)
    if C>call_mid:
      sigma_min=sigma_mid
    else:
      sigma_max=sigma_mid
  return sigma_mid
```

10.6.2 运用二分查找法计算隐含波动率

```
def impvol_put_Binary(P, S, K,r,T):
  '''运用BSM定价模型计算看跌期权的隐含波动率,并且使用的迭代方法是二分查找法
  P:代表看跌期权的市场价格;
  S:代表期权基础资产的价格;
  K:代表期权的执行价格;
  r:代表无风险收益率;
  T:代表期权合约的剩余期限。'''
  def put_BS(S, K, sigma, r,T):
    import numpy as np
    from scipy.stats import norm #从sciFy的子模块stats中导入norm函数
    d1=(np.log (S/K)+(r+pow (sigma, 2)/2)*T)/(sigma*np.sqrt (T) )
    d2=d1-sigma*np.sqrt (T)
    return K*np.exp (-r*T) *norm.cdf(-d2) - S*norm.cdf(-d1)
  sigma_min=0.001   #设定波动率的初始最小值
  sigma_max=1.000   #设定波动率的初始最大值
  sigma_mid=(sigma_min+sigma_max)/2
  put_min=put_BS(S,K, sigma_min,r, T)
  put_max=put_BS(S, K, sigma_max,r, T)
  put_mid=put_BS(S, K, sigma_mid,r, T)
  diff=P-put_mid
  if P<put_min or P>put_max:
    print ('Error')
  while abs (diff) >1e-6:
    diff=P-put_BS(S, K,sigma_mid,r, T)
    sigma_mid=(sigma_min+sigma_max)/2
    put_mid=put_BS(S, K, sigma_mid, r, T)
    if P>put_mid:
      sigma_min=sigma_mid
    else:
      sigma_max=sigma_mid
  return sigma_mid
```

10.6.2 运用二分查找法计算隐含波动率

【例10-26】沿用前面例10-25的信息，依然是假定看涨期权的市场价格为0.1566元、看跌期权的市场价格是0.7503元，通过前面Python定义的运用二分查找法计算隐含波动率的函数impvol_call_Binary和impvol_put_Binary，分别求出看涨、看跌期权的隐含波动率， 具体的代码如下：

```
imp_vol3=impvol_call_Binary (C=0.1566, S=5.29, K=6, r=0.04, T=0.5)
imp_vol4=impvol_put_Binary (P=0.7503,S=5.29, K=6, r=0.04, T=0.5)
print ('用二分查找法计算得到看涨期权隐含波动率: ', round (imp_vol3,4))
print ('用二分查找法计算得到看跌期权隐含波动率: ', round (imp_vol4,4))
```

用二分查找法计算得到看涨期权隐含波动率：0.2427
用二分查找法计算得到看跌期权隐含波动率：0.2446

显然，通过用二分查找法计算得到的期权隐含波动率与运用牛顿迭代法得出的结果是一致的，但是运算效率会更高。

此外，隐含波动率也正在被制作成指数。比如，芝加哥期权交易所(CBOE)于1993年对外发布了隐含波动率的指数VIX(Volatity Index)，VIX也被称作“恐惧指数”(fear factor)。此后，德国、法国、英国、瑞士、韩国等国家及我国的香港、台湾地区也相继推出了波动率指数。国际经验表明，波动率指数能够前瞻性地反映市场情绪与风险。例如，在2008年国际金融危机中，波动率指数及时准确地为全球各家金融监管机构提供了掌握市场压力、监控市场情绪的指标，有效提升了监管能力与决策水平。

10.7 波动率微笑与斜偏

10.7.1 波动率微笑

有人或许会问，在现实的期权交易中交易员和分析师会采用BSM模型对期权进行定价吗?答案是会将该模型的定价结果作为参考，但真实运用的定价模型将会考虑到允许波动率依赖于期权执行价格的变化而变化这一因素。

波动率微笑(volatility smile)是一种描述期权隐含波动率与执行价格函数关系的图形，具体是指针对相同的到期日和基础资产、但不同执行价格的期权，当执行价格偏离基础资产现货价格越远时，期权的隐含波动率就越大，类似于微笑曲线。下面，就通过上证50ETF期权作为例子来刻画波动率微笑曲线。

【例10-27】以2018年6月27日到期的、不同执行价格的上证50ETF期权合约在2017年12月29日(2017年最后一个交易日)的收盘价数据作为分析对象(见表10-4)，一共有7只看涨期权和7只看跌期权；当天的上证50ETF基金净值等于2.859元，无风险利率是运用6个月期Shibor利率，当天的利率是4.8823%。采用10.6节通过Python定义的运用牛顿迭代法计算看涨、看跌期权隐含波动率的函数impvol_call_Newton、impvol_put_Newton求出期权的隐含波动率，并且进行可视化，具体的过程分为4个步骤。

第1步：在Python中输入相关的变量，具体的代码如下:

```
import datetime as dt   #导入datetime模块
T1=dt.datetime (2017, 12,29) #计算隐含波动率的日期
T2=dt.datetime (2018, 6,27) #期权到期
T_delta= (T2-T1).days/365 #期权的剩余期限
S0=2.859    #50ETF基金净值
Call_list=np.array  ([0.2841,0.2486,0.2139,0.1846,0.1586,0.1369,0.1177])  #输入50ETF认购期权收盘价格
Put_list=np.array  ([0.0464,0.0589,  0.0750,  0.0947,0.1183,0.1441,  0.1756])  #输入5OETE
K_list=np.array ([2.7, 2.75,2.8, 2.85, 2.9,2.95, 3.0]) #输入期权执行价格
shibor=0.048823   #6个月期的SHIBOR利率
```

第2步：计算上证50ETF认购期权(看涨期权)的隐含波动率，并且需要运用for语句，具体的代码如下:

```
sigma_clist=np.zeros_like (Call_list) #构建存放隐含波动率的初始数组
for i in np.arange (len(Call_list)) :
    sigma_clist[i]=impvol_call_Newton(C=Call_list    [i],    S=S0,    K=K_list[i], r=shibor,T=T_delta)
print ('通过看涨期权计算得到的隐含波动率: ',sigma_clist)
```

第3步：计算上证50ETF认沽期权(看跌)的隐含波动率，同样需要运用for语句，具体的代码如下:

```
sigma_plist=np.zeros_like (Put_list) #构建存放隐含波动率的初始数组
for i in np.arange (len(Put_list) ):
    sigma_plist[i]=impvol_put_Newton      (P=Put_list[i],      S=S0,K=K_list[i], r=shibor, T=T_delta)
print('通过看跌期权计算得到的隐含波动率: ',sigma_plist)
```

第4步：将期权执行价格与隐含波动率的关系可视化，也就是绘制出波动率微笑曲线(见图10-19)，具体的代码如下:

```
plt.figure(figsize=(8,6))
plt.plot (K_list, sigma_clist, 'b-',label=u'50EFT认购期权(看涨) ',lw=2.5)
plt.plot (K_list, sigma_plist,'r-',label=u'50EFr认沽期权(看跌)' ,lw=2.5)
plt.xlabel (u'期权的执行价格',fontsize=13)
plt.ylabel (u'隐含波动率',fontsize=13,rotation=90)
plt.xticks (fontsize=13)
plt.yticks (fontsize=13)
plt.title(u'期权执行价格与期权隐含波动率的关系', fontsize=13)
plt.legend (fontsize=13)
plt.grid ('True')
plt. show()
```

第3步：计算上证50ETF认沽期权(看跌)的隐含波动率，同样需要运用for语句，具体的代码如下:

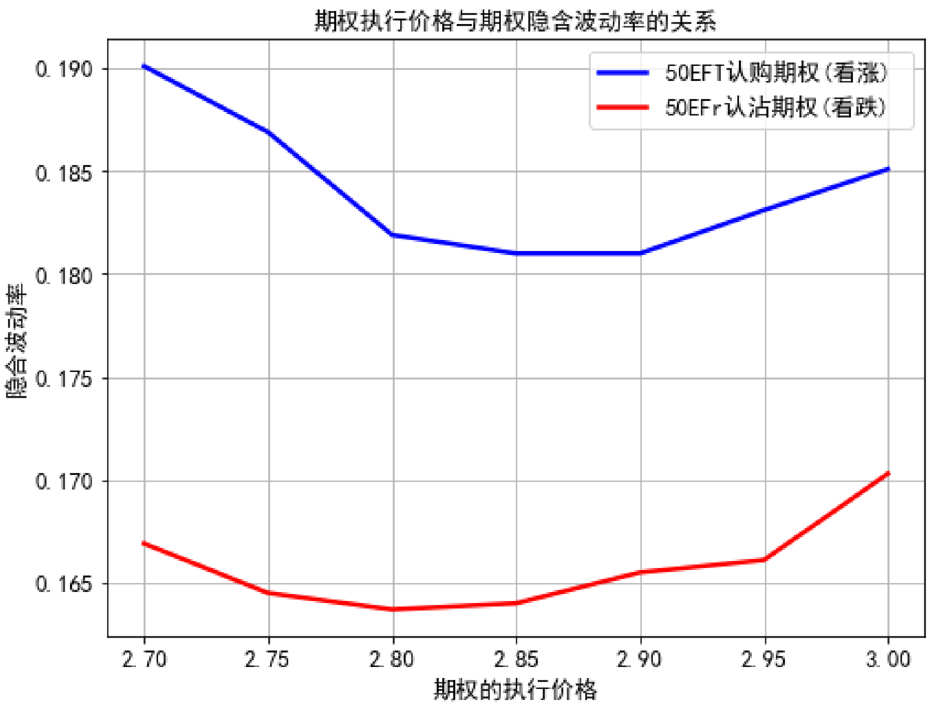
10.7.1 波动率微笑

图10-19刻画了2017年12月29日上证50ETF认购期权、认沽期权的隐含波动率与执行价格之间的关系，图中上方的曲线代表上证50ETF认购期权、下方的曲线代表上证50ETF认沽期权。从图10-19中不难发现，无论是看涨期权还是看跌期权，当期权执行价格越接近于基础资产价格(2.859元)时，期权的隐含波动率基本上就越低；越远离基础资产价格时，则隐含波动率越高，因此存在着比较明显的波动率微笑特征。

对于为什么会产生波动率微笑，存在着较多的理论研究，并且基于不同的研究视角给出了各种不同的解释，大体可以分为两类：

第1类是从BSM模型固有的缺陷进行解释；

第2类是从市场交易机制层面进行解释。



然而，在大多数交易中，股票期权的波动率曲线不是微笑的，而是表现为波动率偏斜(volatity skew)。

波动率偏斜可以分为广义和狭义两种，广义的波动率偏斜就泛指不满足微笑形状的各种波动率曲线；狭义波动率偏斜则是指当期权的执行价格由小变大时，期权的隐含波动率则是由大变小，即隐含波动率是执行价格的减函数。

下面，依然通过上证50ETF期权作为例子刻画狭义的波动率斜偏。

【例10-28】以2019年6月26日到期的、不同执行价格的上证50ETF期权合约在2018年12月28日(2018年最后一个交易日)的收盘价数据作为分析对象(见表10-6)，一共有13只看涨期权和13只看跌期权;当天的上证50ETF基金净值等于2.289元，无风险利率是运用6个月期的Shibor利率，当天该利率是3.297%。运用10.6节通过Python定义的运用二分查找法计算看涨、看跌期权隐含波动率的函数impvol_call_Binary和impvol_put_Binary，求出期权的隐含波动率并且进行可视化(见图10-20)，具体的过程分为4个步骤。

第1步：在Python中输入相关变量，具体的代码如下：

```
T1=dt.datetime (2018, 12,28)#计算隐含波动率的日期
T2=dt.datetime (2019, 6,26)#期权到期日
T_delta=(T2-T1).days/365#期权的剩余期限
S0=2.289#50ETF基金净值
Call_list=np.array ([0.2866,0.2525,0.2189,0.1912,0.1645,
0.1401,0.1191,0.0996,0.0834, 0.0690,0.0566,0.0464,0.0375])#输入50ETF认购期权的收盘价格
Put_list=np.array ( [0.0540,0.0689,0.0866,0.1061,0.1294,
0.1531,0.1814,0.2122,0.2447, 0.2759,0.3162,0.3562,0.3899])##50ETF认沽期权的收盘价格
K_list=np.array ([2.1,2.15,2,2,2.25,2.3,2.35,2.4,2.45, 2.5,2.55,2.6,2.65,2.7])#输入期权执行价格
shibor=0.03297#6个月期的SHIBOR利率
```

第2步：计算上证50ETF认购期权(看涨期权)的隐含波动率，并且需要运用for语句，具体的代码如下:

```
sigma_clist=np.zeros_like(Call_list) #构建一个初始的隐含波动率数组
for i in np.arange (len (Call_list)):
   sigma_clist[i]=impvol_call_Binary(C=Call_list[i], S=S0, K=K_list[i], r=shibor, T=T_delta)
print ('通过看涨期权计算得到的隐含波动率: ',sigma_clist)
```

第3步：计算上证50ETF认沽期权(看跌期权)的隐含波动率，同样需要运用for语句，具体的代码如下:

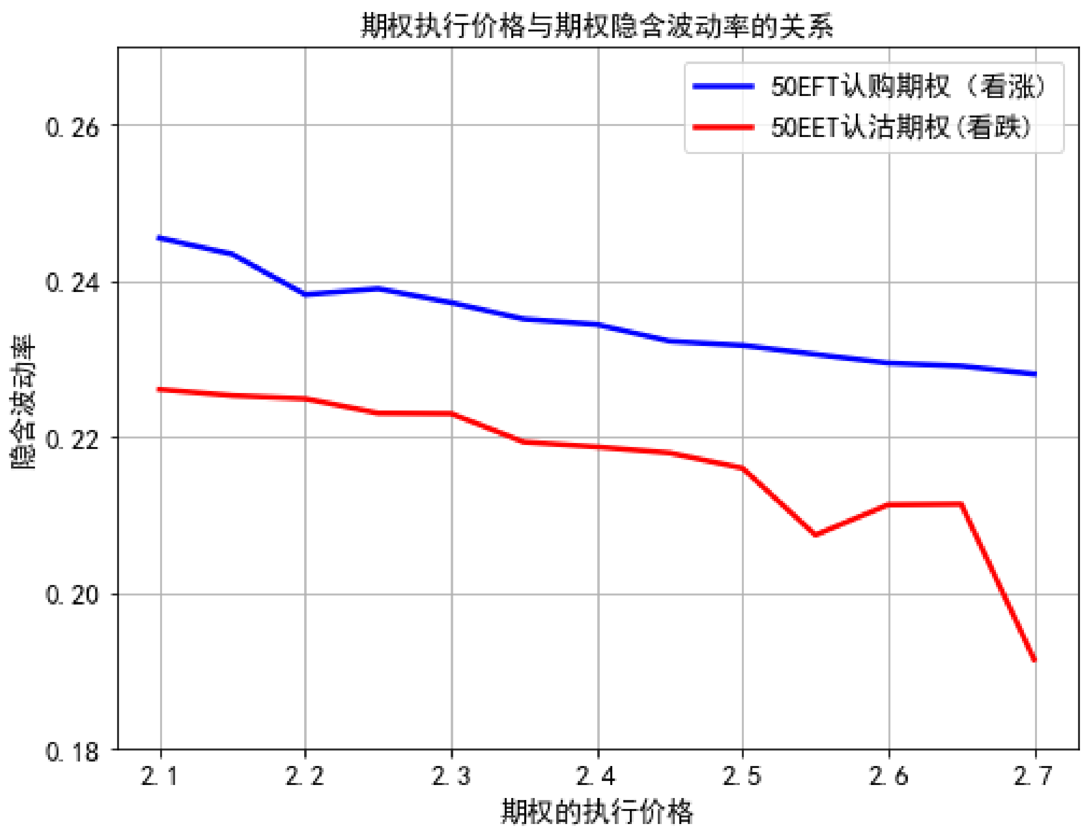
```
sigma_plist=np.zeros_like (Put_list) #构建一个初始的隐含波动率数组
for i in np.arange (len (Put_list)):
   sigma_plist[i]=impvol_put_Binary   (P=Put_list[i],   S=S0,   K=K_list[i],   r=shibor,
T=T_delta)
print ('通过看跌期权计算得到的隐含波动率: ',sigma_plist)
```

第4步：将期权执行价格与隐含波动率的关系可视化，也就是绘制出波动率斜偏曲线，具体的代码如下:

```
plt.figure (figsize= (8, 6) )
plt.plot (K_list, sigma_clist, 'b-', label=u'50EFT认购期权（看涨)',lw=2.5)
plt.plot (K_list,sigma_plist, 'r-',label=u'50EET认沽期权(看跌)' ,lw=2.5)
plt.xlabel (u'期权的执行价格', fontsize=13)
plt.ylabel (u'隐含波动率', fontsize=13, rotation=90)
plt.ylim (0.18,0.27)
plt.xticks (fontsize=13)
plt.yticks (fontsize=13)
plt.title (u'期权执行价格与期权隐含波动率的关系', fontsize=13)
plt.legend (fontsize=13)
plt.grid('True')
plt. show()
```

10.7.2 波动率斜偏

图10-20刻画了在2018年12月28日上证50ETF认购期权、认沽期权的隐含波动率与执行价格之间的关系，图中的上方曲线代表认购期权、下方曲线代表认沽期权，不难发现曲线存在着比较明显的狭义波动率斜偏特征。这张图还有另一个现实意义：对于执行价格较低的期权(通常是深度虚值看跌期权或深度实值看涨期权)，这类期权的隐含波动率较高;相比之下，对于执行价格较高的期权(通常是深度实值看跌期权或深度虚值看涨期权)，此类期权的隐含波动率较低。



Your appreciation makes me a miracle.
Thank you!

Frank Ziwei Zhang
18117228563
frank8027@163.com

上海对外经贸大学
SHANGHAI UNIVERSITY OF INTERNATIONAL BUSINESS AND ECONOMICS