


Chapter 5 – Matplotlib 金融场景

Frank Ziwei Zhang
School of Finance



上海對外經貿大學
SHANGHAI UNIVERSITY OF INTERNATIONAL BUSINESS AND ECONOMICS

Contents



Q1	基本函数
Q2	线图
Q3	直方图
Q4	条形图
Q5	散点图
Q6	饼图
Q7	股价蜡烛图

5.1 基本函数

5.1 基本函数

Matplotlib有一套完全仿照 MATLAB函数形式的绘图接口。此外，在金融领域针对 Matplotlib运用最多的是子模块pyplot。同样，由于是第三方模块，因此需要导入并且查看模块的版本号信息，具体代码如下：

```
import matplotlib
```

```
matplotlib.__version__
```

```
import matplotlib.pyplot as plt #导入matplotlib子模块pyplot
```

5.1 基本函数

函数	函数功能	主要参数的说明
figure	定义画面大小	<p>figsize: 指定宽和高（单位英寸），figsize=（9,6）代表宽是9英寸、高是6英寸</p> <p>dpi: 代表分辨率（每英寸多少个像素）。默认值是80</p> <p>facecolor: 设定图形的背景颜色</p> <p>edgecolor: 设定图形的边框颜色</p> <p>frameon: 表示是否显示边框，frameon=True代表显示边框，frameon=False代表不显示边框</p>
plot	曲线图	<p>x: 代表对应于x轴的数据</p> <p>y: 代表对应于y轴的数据</p> <p>label: 表示曲线的标签</p> <p>format_string: 表示控制曲线的格式字串，包括设定颜色（具体见表5-2）、样式（具体见表5-3）、宽度（lw，比如lw=2.0表示曲线的宽是2磅）等</p>
subplot	子图	<p>nrows: 表示子图的行数</p> <p>ncols:表示子图的列数；</p> <p>index:表示子图的序号，最大序号是等于行数乘以列数</p>

5.1 基本函数

hist	直方图	<p>x: 表示每个矩形（bin）分布所应对的数据，对应图中的X轴；</p> <p>bins:表示图中的矩形数量，比如bin=20表示有20个矩形</p> <p>facecolor: 设定矩形的背景颜色</p> <p>edgecolor: 设定矩形的边框颜色</p>
bar	垂直条形图	<p>x: 表示条形图x坐标对应的相关数据；</p> <p>height: 代表每个条形图案的高度</p> <p>width: 代表每个条形图案的宽度（可选）</p>
barh	水平条形图	<p>y: 表示条形图y坐标对应的相关数据</p> <p>width: 代表每个条形图案的宽度；</p> <p>height: 代表每个条形图案的高度（可选）</p>
scatter	散点图	<p>x: 代表对应于x变量的数据</p> <p>y: 代表对应于y变量的数据；</p> <p>c: 代表散点的颜色，默认为蓝色</p> <p>marker: 代表散点的样式（具体见表5-3）</p>

5.1 基本函数

pie	饼图	x : 代表饼图中每块饼的占比; labels : 代表饼图中每块饼的标签文字; colors : 代表饼图中每块饼的颜色
axis	坐标轴	xmin : 设置x轴刻度的最小值; xmax : 设置x轴刻度的最大值; ymin : 设置y轴刻度的最小值; ymax : 设置y轴刻度的最大值; 此外, 通过输入以下的字符串对坐标轴进行控制: 'off' : 关闭坐标的轴线和标签; 'equal' 使用等刻度的坐标轴 'Scaled' : 通过尺寸变化平衡坐标轴的刻度; 'tight' : 设置限值使所有数据可见; 'image' : 使刻度的限值等于数据的限值; 'square' : 与 'scaled' 相似, 但强制要求 $xmax-xmin=ymax-ymin$
xticks	x轴的刻度	ticks : 表示x轴刻度的列表, 如果放置一个空列表就表示禁用 xticks ; labels : 在给定x轴刻度位置的标签。
xlabel	x轴的坐标标签	输入字符串的方式, 输出x轴的坐标标签; 可通过输入 fontsize 数字, 控制标签字体的大小; 可通过输入 rotatio 数字, 控制标签的角度

5.1 基本函数

xlim	x轴刻度范围	xmin: 设置x轴刻度的最小值 xmax: 设置x轴刻度的最大值
yticks	y轴的刻度	与xticks的用法相同
ylabel	y轴的坐标标签	与xlabel的用法相同
ylim	y轴刻度范围	与xlim的用法相同
title	图例的标题	输入字符串的方式，输出图例的标题； 可通过输入 fontsize=数字 ，控制标题字体的大小
legend	显示图例	通过输入 loc=数字 ，来控制图例的位置，具体的数字可选范围及含义如下： 0 表示最佳， 1 表示右上， 2 表示左上， 3 表示左下， 4 表示右下， 5 表示右， 6 表示中左， 7 表示中右， 8 表示中下， 9 表示中上， 10 表示中，空白表示自动
grid	网格	通常不用输入参数，如果确有需要，则有以下的主要参数可以选择： axis: 绘制哪一组网格线， axis='x' 表示仅绘制x轴的网格线， axis='y' 表示仅绘制y轴的网格线， axis="both" 表示绘制x、y轴的网格线 color: 设置网格线的颜色； linestyle: 设置网格线的样式 linewidth: 设置网格线的宽度

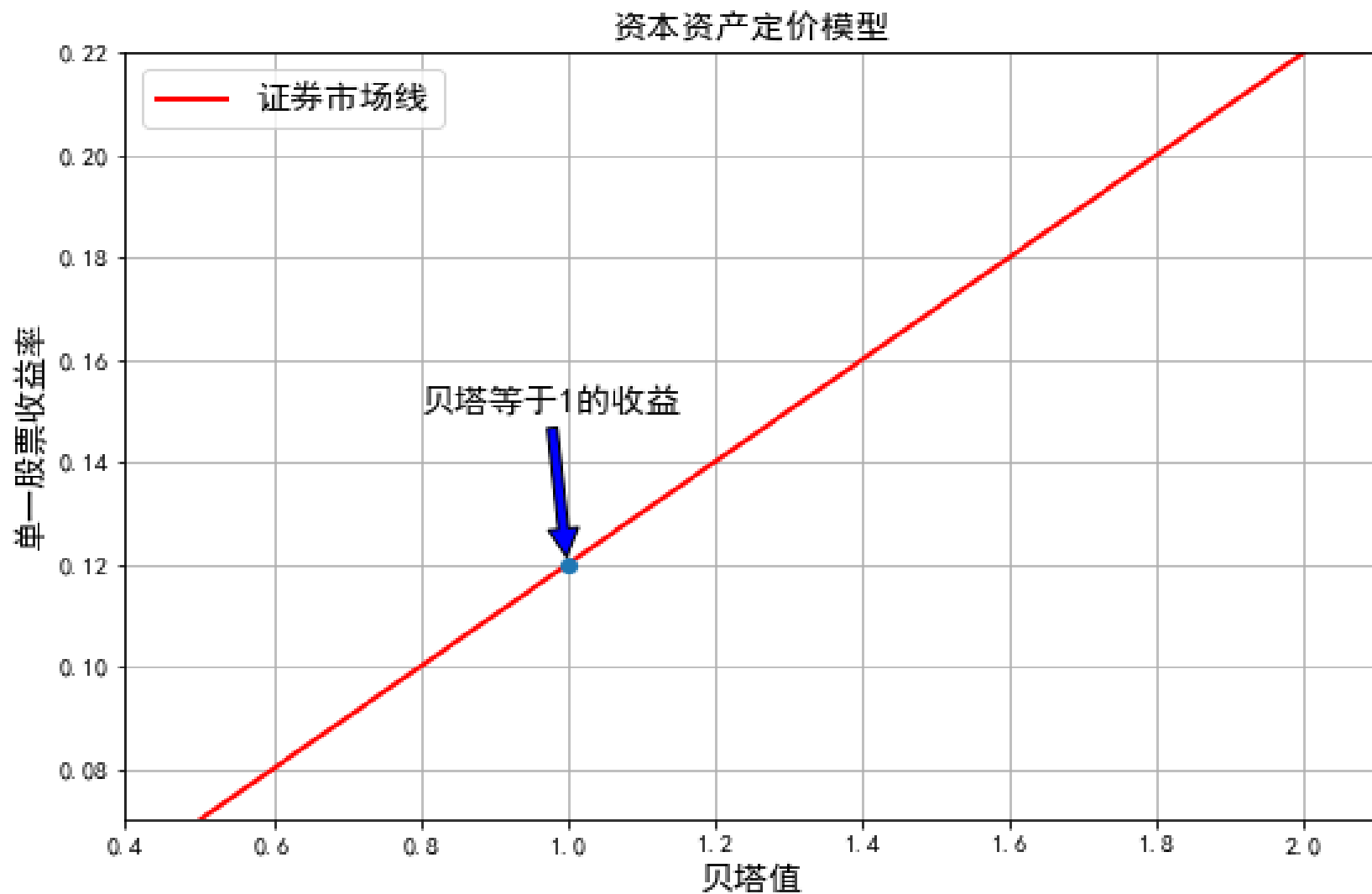
5.2 线图

5.2.1 单一曲线图

【例5-1】假定某一个股票的贝塔值处于[0.5,2.0]的区间之中，金融市场的无风险利率是3%，市场收益率是12%，运用子模块pyplot中的plot函数绘制该股票的证券市场线（图5-1）。具体的代码如下：

```
import numpy as np
beta = np.linspace(0.5,2.0,100)
Rf = 0.02
Rm = 0.12
Ri = Rf + beta * (Rm - Rf) #CAPM模型
plt.figure(figsize=(9,6))
plt.plot(beta,Ri,'r-',label='证券市场线',lw=2.0)
plt.plot(1.0,Rf+1.0*(Rm-Rf),'o',lw=2.5) #图中画一个贝塔值等于1的收益率点
plt.axis('tight')
plt.xlabel(u'贝塔值',fontsize=14)
plt.xlim(0.4,2.1)
plt.ylabel(u'单一股票收益率',fontsize=14,rotation=90)
plt.ylim(0.07,0.22)
plt.title(u'资本资产定价模型',fontsize=14)
plt.annotate(u'贝塔等于1的收益',
            fontsize=14,xy=(1.0,0.12),xytext=(0.8,0.15),arrowprops=dict(facecolor='b',shrink=0.05))
plt.legend(loc=0,fontsize=14)
plt.grid()
```

5.2.1 单一曲线图



5.2.2 数据框可视化的函数

【例5-2】以在第4章中例4-19中拼接生成的存放2018年期间沪深300指数（见图5-2）每日开盘点位、最高点位、最低点位、收盘点位的数据框作为演示对象，运用subplot函数绘制相应的2×2图形（每一行有两个图、每一列也有两个图），具体的代码如下：

```
import pandas as pd

HS300_new = pd.read_excel('D:\Zhangzw\Python\Python金融数据分析\RawData\第4章\沪深300指数.xlsx',sheet_name="Sheet1",header=0,index_col=0) #注意导入的是sheet1

plt.figure(figsize=(11,9))

#####

plt.subplot(2,2,1)

plt.plot(HS300_new['开盘点位'],'r-',label=u'沪深300开盘点位',lw=2.0)

plt.xticks(fontsize=13,rotation=30)

plt.xlabel(u'日期',fontsize=13)

plt.yticks(fontsize=13)

plt.ylabel(u'点位',fontsize=13,rotation=30)

plt.legend(loc=0,fontsize=13)

plt.grid()
```

5.2.2 数据框可视化的函数

#####第1行第2个

```
plt.subplot(2,2,2)
plt.plot(HS300_new['最高点位'],'b-',label=u'沪深300最高点位',lw=2.0)
plt.xticks(fontsize=13,rotation=30)
plt.xlabel(u'日期',fontsize=13)
plt.yticks(fontsize=13)
plt.ylabel(u'点位',fontsize=13,rotation=30)
plt.legend(loc=0,fontsize=13)
plt.grid()
```

#####第2行第1个

```
plt.subplot(2,2,3)
plt.plot(HS300_new['最低点位'],'b-',label=u'沪深300最低点位',lw=2.0)
plt.xticks(fontsize=13,rotation=30)
plt.xlabel(u'日期',fontsize=13)
plt.yticks(fontsize=13)
plt.ylabel(u'点位',fontsize=13,rotation=30)
plt.legend(loc=0,fontsize=13)
plt.grid()
```

#####第2行第2个

```
plt.subplot(2,2,4)
plt.plot(HS300_new['收盘点位'],'b-',label=u'沪深300收盘点位',lw=2.0)
```

5.2.2 数据框可视化的函数

#####第2行第2个

```
plt.subplot(2,2,4)
```

```
plt.plot(HS300_new['收盘点位'],'b-',label=u'沪深300收盘点位',lw=2.0)
```

```
plt.xticks(fontsize=13,rotation=30)
```

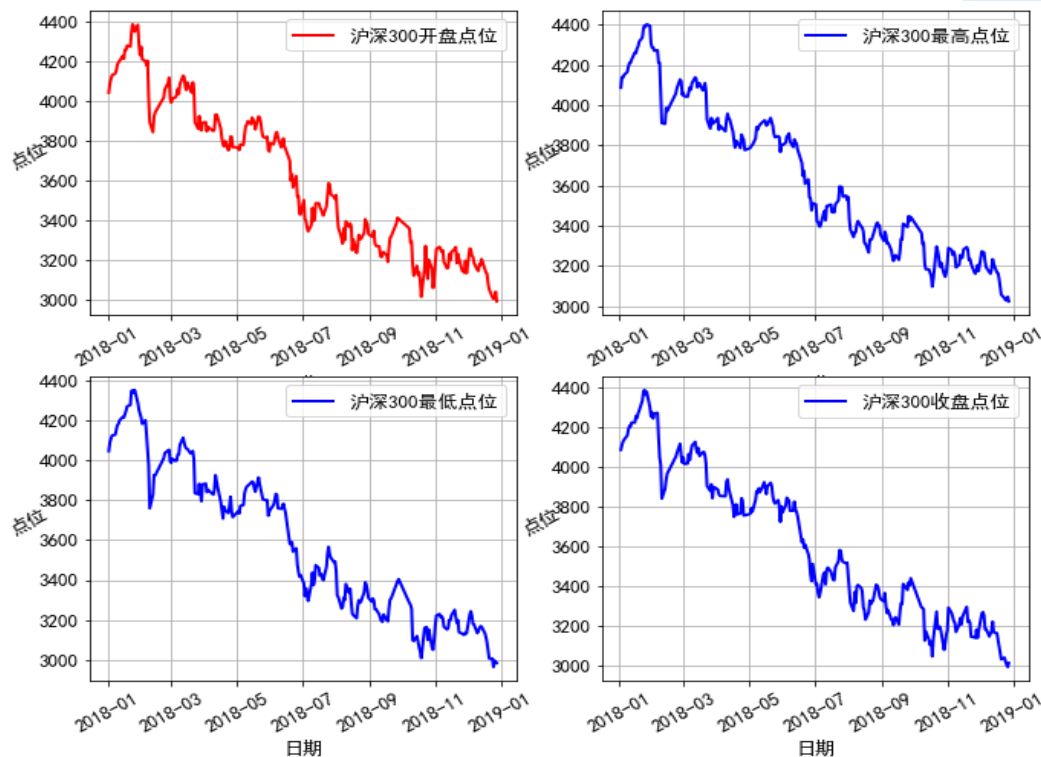
```
plt.xlabel(u'日期',fontsize=13)
```

```
plt.yticks(fontsize=13)
```

```
plt.ylabel(u'点位',fontsize=13,rotation=30)
```

```
plt.legend(loc=0,fontsize=13)
```

```
plt.grid()
```

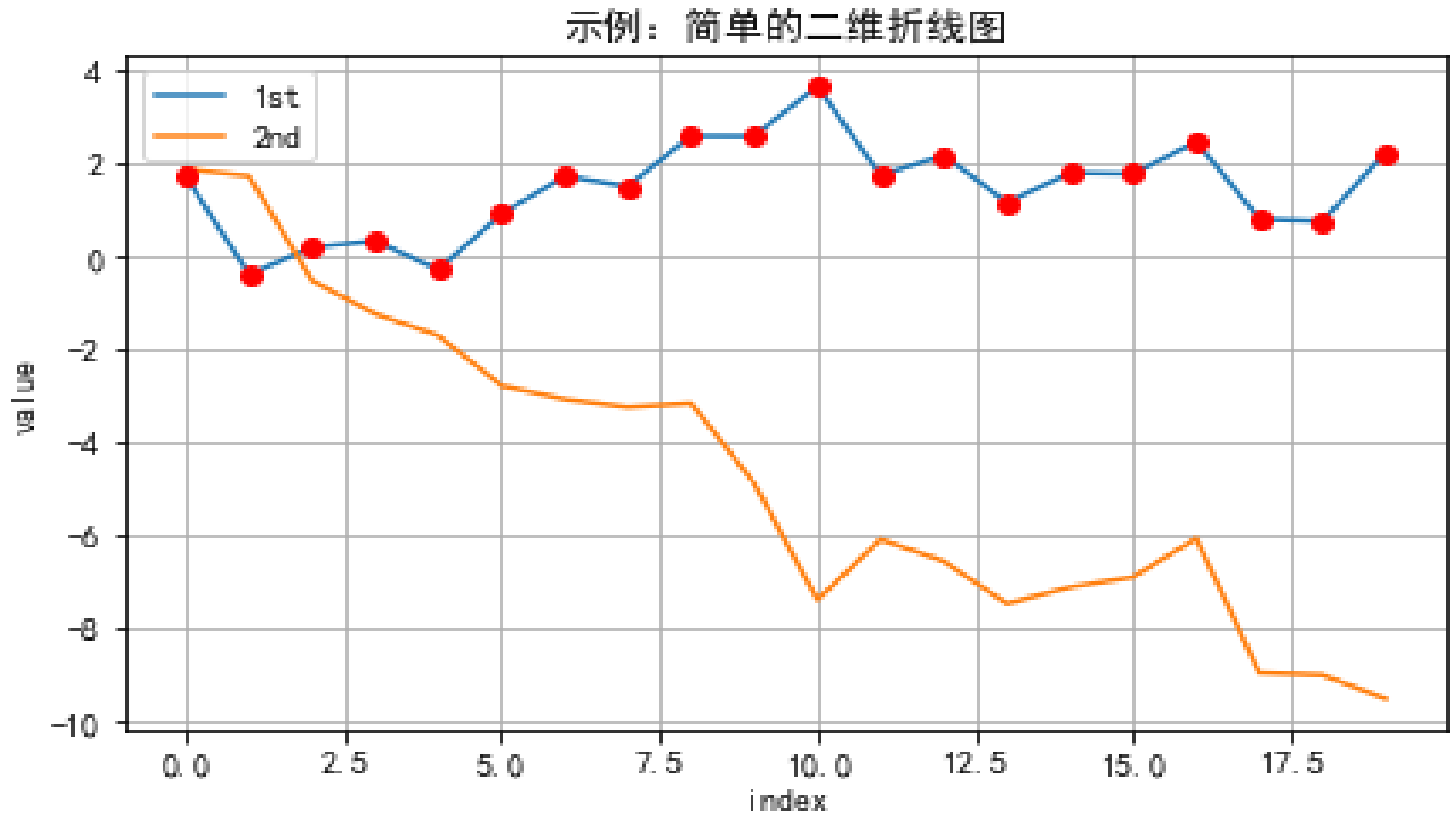


5.2.3 二维图与双轴图

【例5-2-1】利用标准正态分布生成两列数据，并且在一个坐标系中表示出来：

```
import numpy as np
import matplotlib.pyplot as plt
# 单轴双数据折线图
np.random.seed(2000)
y = np.random.standard_normal((20, 2)).cumsum(axis=0) #cumsum函数：沿着指定轴的元素累加和所组成的数组，其形状应与输入原数组一致
plt.figure(figsize=(7, 4))
plt.plot(y[:,0],lw=1.5,label='1st') #第一列数据折线
plt.plot(y[:,1],lw=1.5,label='2nd') #第二列数据折线
plt.plot(y[:,0],'ro') #用红色的点给第一列数据标出位置
plt.grid(True)
plt.legend(loc=0) #在最佳位置显示数据示例
plt.axis('tight') #所有数据可见（缩小限值）
plt.xlabel('index')
plt.ylabel('value')
plt.title('示例：简单的二维折线图')
```

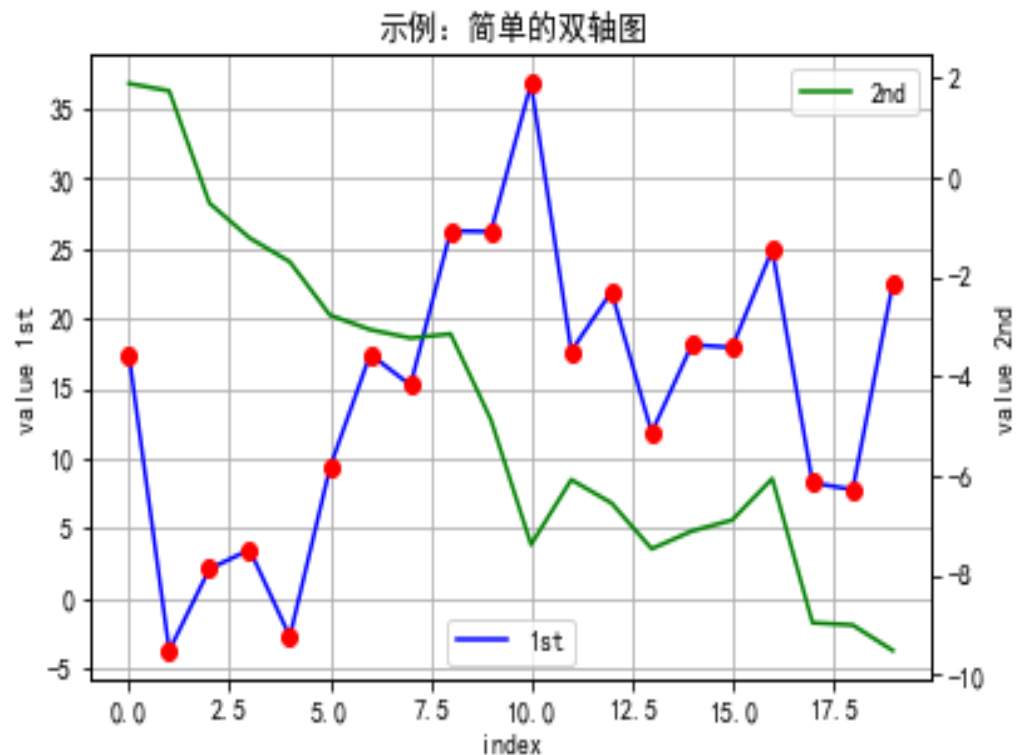
5.2.3 二维图与双轴图

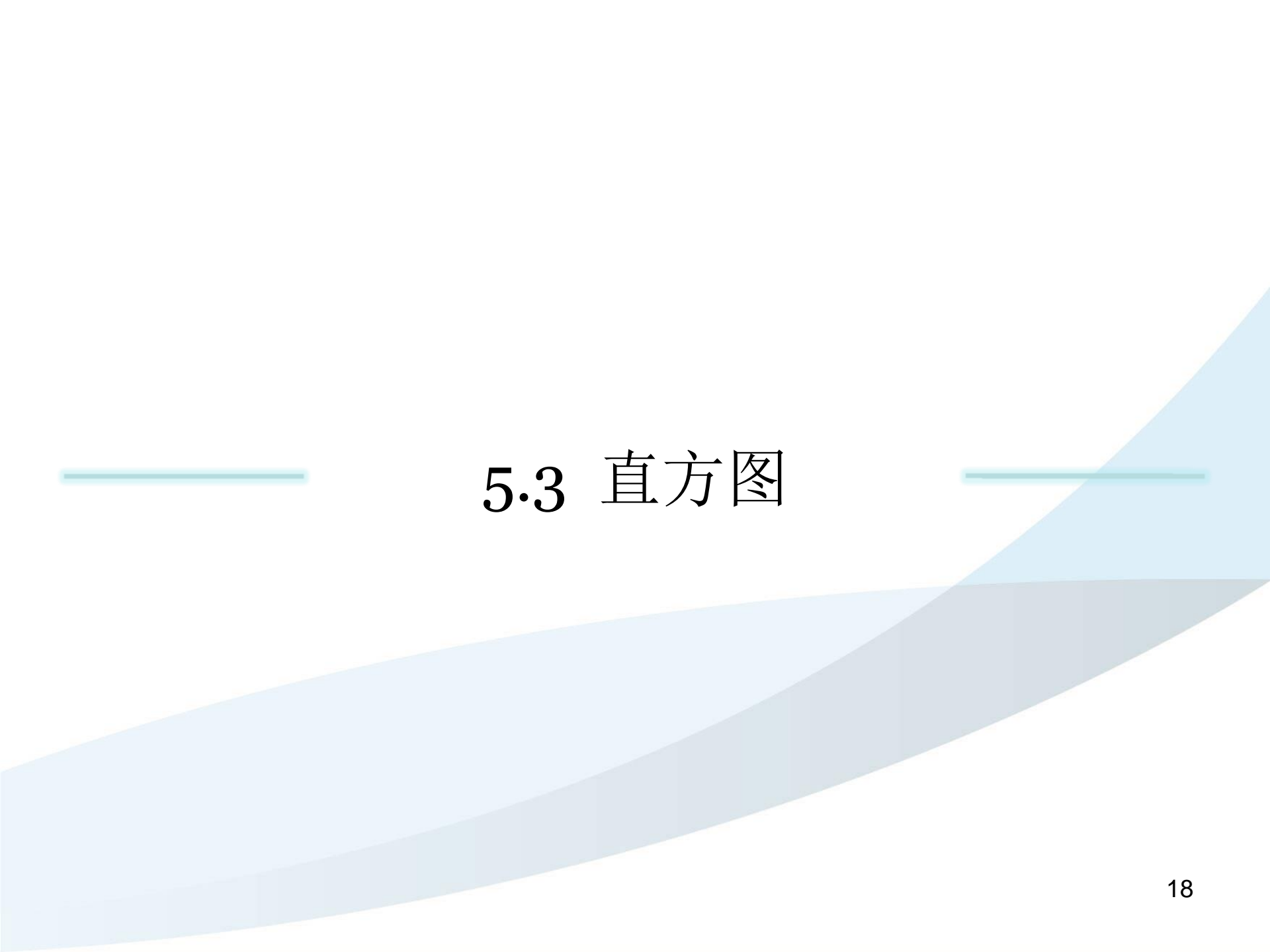


5.2.3 二维图与双轴图

【例5-2-2】利用标准正态分布生成两列数据，并且用双轴图表示出来：

```
y[:,0] = y[:,0] * 10  
fig, ax1 = plt.subplots()  
plt.plot(y[:, 0], 'b', lw=1.5, label='1st')  
plt.plot(y[:, 0], 'ro')  
plt.grid(True)  
plt.legend(loc=8)  
plt.axis('tight')  
plt.xlabel('index')  
plt.ylabel('value 1st')  
plt.title('示例：简单的双轴图')  
ax2 = ax1.twinx()  
plt.plot(y[:, 1], 'g', lw=1.5, label='2nd')  
plt.legend(loc=0)  
plt.ylabel('value 2nd')
```





5.3 直方图

5.3.1 单一样本的直方图

【例5-3】以3.5节讨论的运用 NumPy模块获取基于不同分布的随机数作为演示直方图的数据源，同时在本例中选择正态分布、对数正态分布、卡方分布以及贝塔分布（图5-3），并且从每个分布中随机抽取1000个样本，最后以 2×2 子图的方式呈现。具体的代码如下：

```
import numpy.random as npr
l=1000 #从每个分布中随机抽取1000个样本
x_norm = npr.normal(loc=0.8,scale=1.5,size=l)
x_logn = npr.lognormal(mean=0.5,sigma=1.0,size=l)
x_chi = npr.chisquare(df=4,size=l)
x_beta = npr.beta(a=2,b=4,size=l)

plt.figure(figsize=(12,10))
#####第1行第1个
plt.subplot(2,2,1)
plt.hist(x_norm,label=u'正态分布的抽样',bins=20,facecolor='y',edgecolor='k')
plt.xticks(fontsize=13)
plt.xlabel(u'样本值',fontsize=13)
plt.yticks(fontsize=13)
plt.ylabel(u'频数',fontsize=13,rotation=90)
plt.legend(loc=0,fontsize=13)
plt.grid(True)
```

5.3.1 单一样本的直方图

#####第1行第2个

```
plt.subplot(2,2,2)
plt.hist(x_logn,label=u'对数正态分布的抽样',bins=20,facecolor='r',edgecolor='k')
plt.xticks(fontsize=13)
plt.xlabel(u'样本值',fontsize=13)
plt.yticks(fontsize=13)
plt.ylabel(u'频数',fontsize=13,rotation=90)
plt.legend(loc=0,fontsize=13)
plt.grid(True)
```

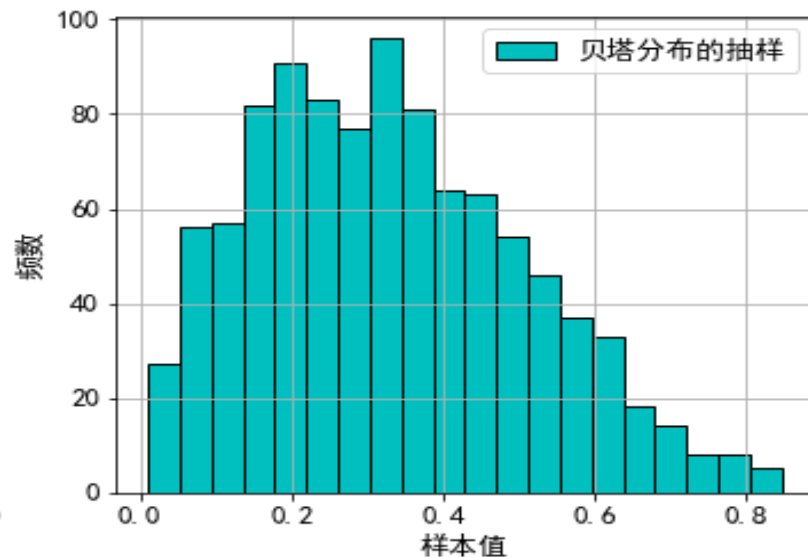
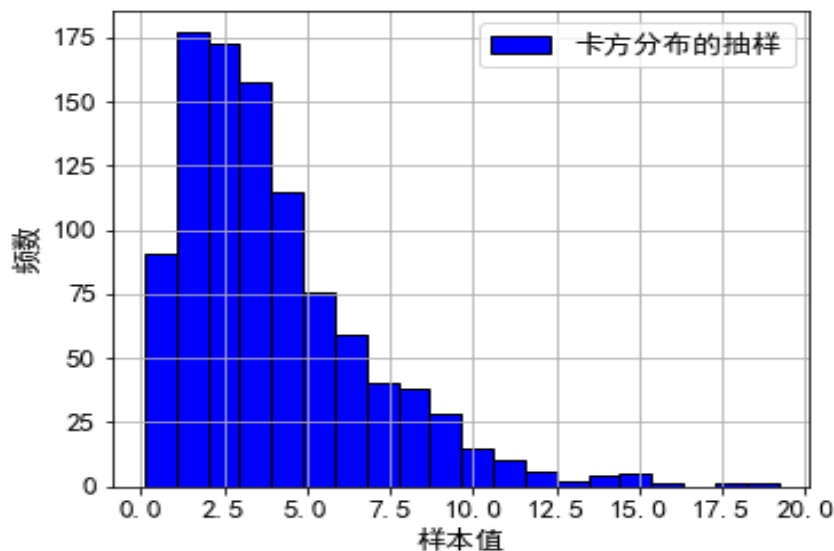
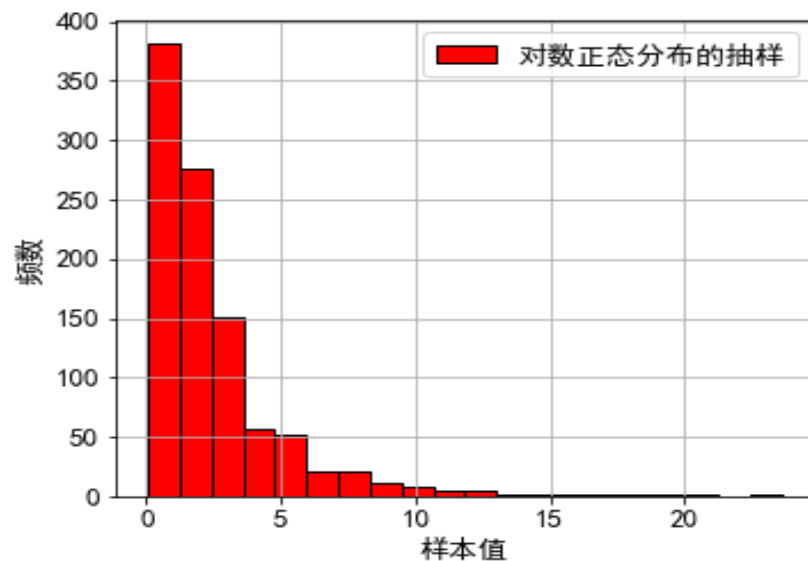
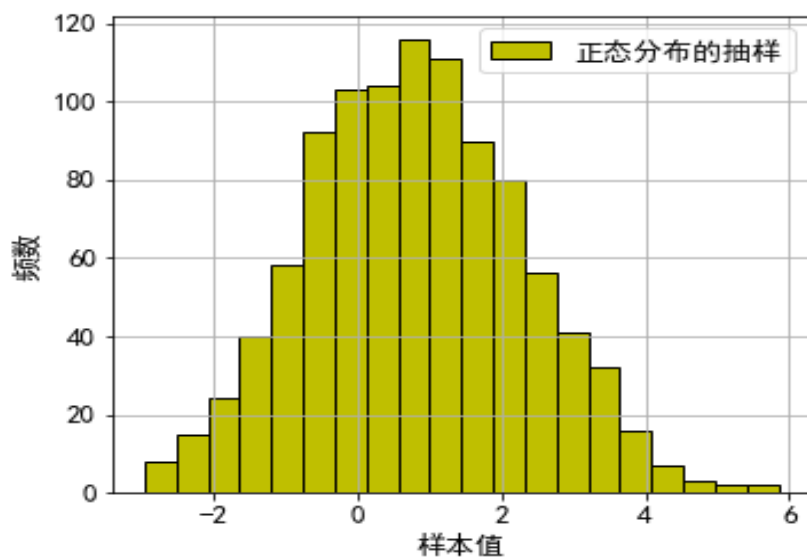
#####第2行第1个

```
plt.subplot(2,2,3)
plt.hist(x_chi,label=u'卡方分布的抽样',bins=20,facecolor='b',edgecolor='k')
plt.xticks(fontsize=13)
plt.xlabel(u'样本值',fontsize=13)
plt.yticks(fontsize=13)
plt.ylabel(u'频数',fontsize=13,rotation=90)
plt.legend(loc=0,fontsize=13)
plt.grid(True)
```

#####第2行第2个

```
plt.subplot(2,2,4)
plt.hist(x_beta,label=u'贝塔分布的抽样',bins=20,facecolor='c',edgecolor='k')
plt.xticks(fontsize=13)
plt.xlabel(u'样本值',fontsize=13)
plt.yticks(fontsize=13)
plt.ylabel(u'频数',fontsize=13,rotation=90)
plt.legend(loc=0,fontsize=13)
```

5.3.1 单一样本的直方图



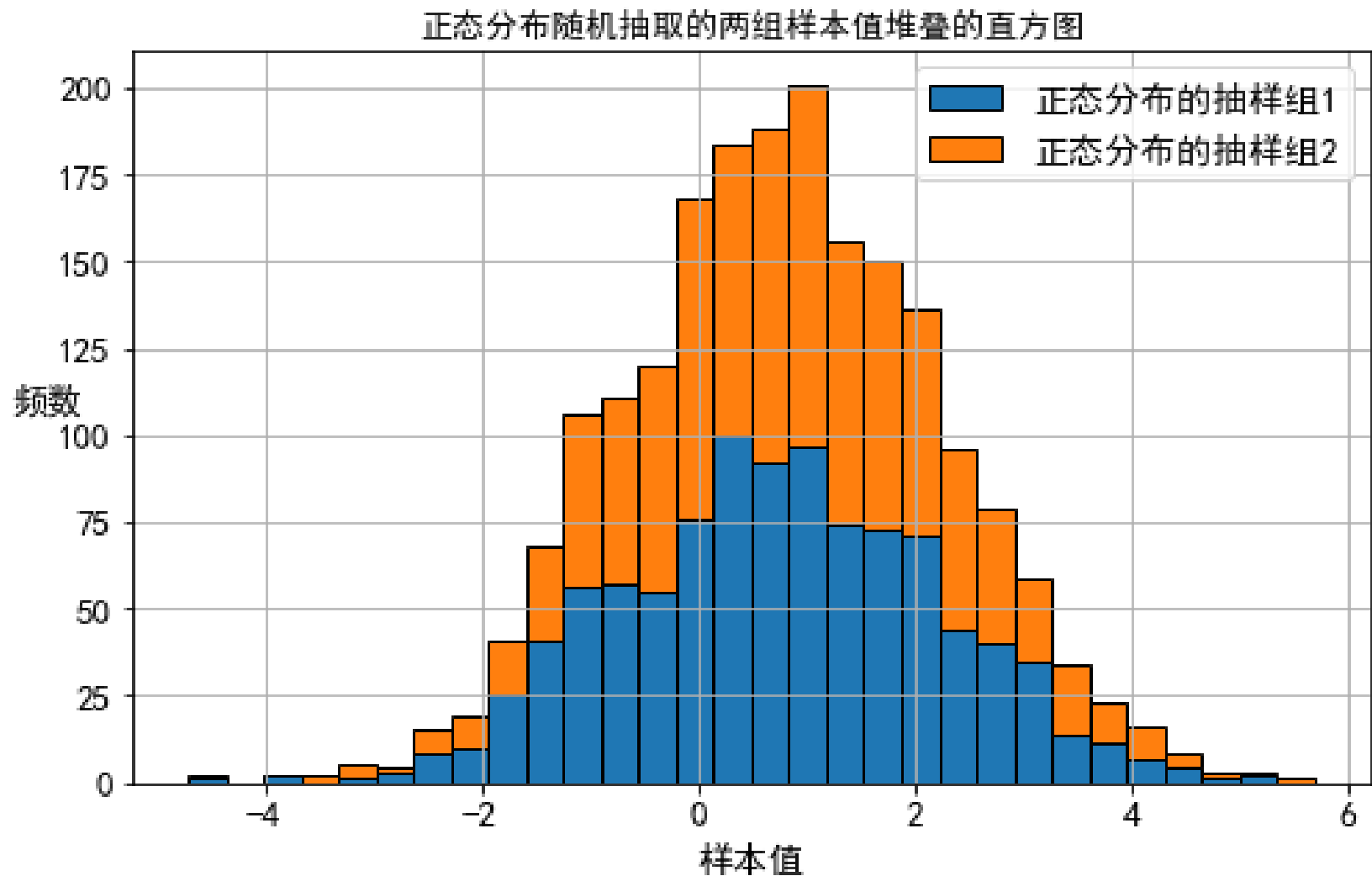
5.3.2 多个样本的直方图

1、堆叠展示

【例5-4】从均值为0.8、标准差为1.5的正态分布中随机抽取两组样本值，每组各1000个样本值，通过直方图并且以堆叠的形式展示（见图5-4），具体的代码如下：

```
x_norml = npr.normal(loc=0.8,scale=1.5,size=(1,2))
plt.figure(figsize=(8,5))
plt.hist(x_norml,label=[u'正态分布的抽样组1',u'正态分布的抽样组
2'],stacked=True,bins=30,edgecolor='k')
plt.xticks(fontsize=13)
plt.xlabel(u'样本值',fontsize=13)
plt.yticks(fontsize=13)
plt.ylabel(u'频数',fontsize=13,rotation=0)
plt.title(u'正态分布随机抽取的两组样本值堆叠的直方图')
plt.legend(loc=0,fontsize=13)
plt.grid(True)
```

5.3.2 多个样本的直方图



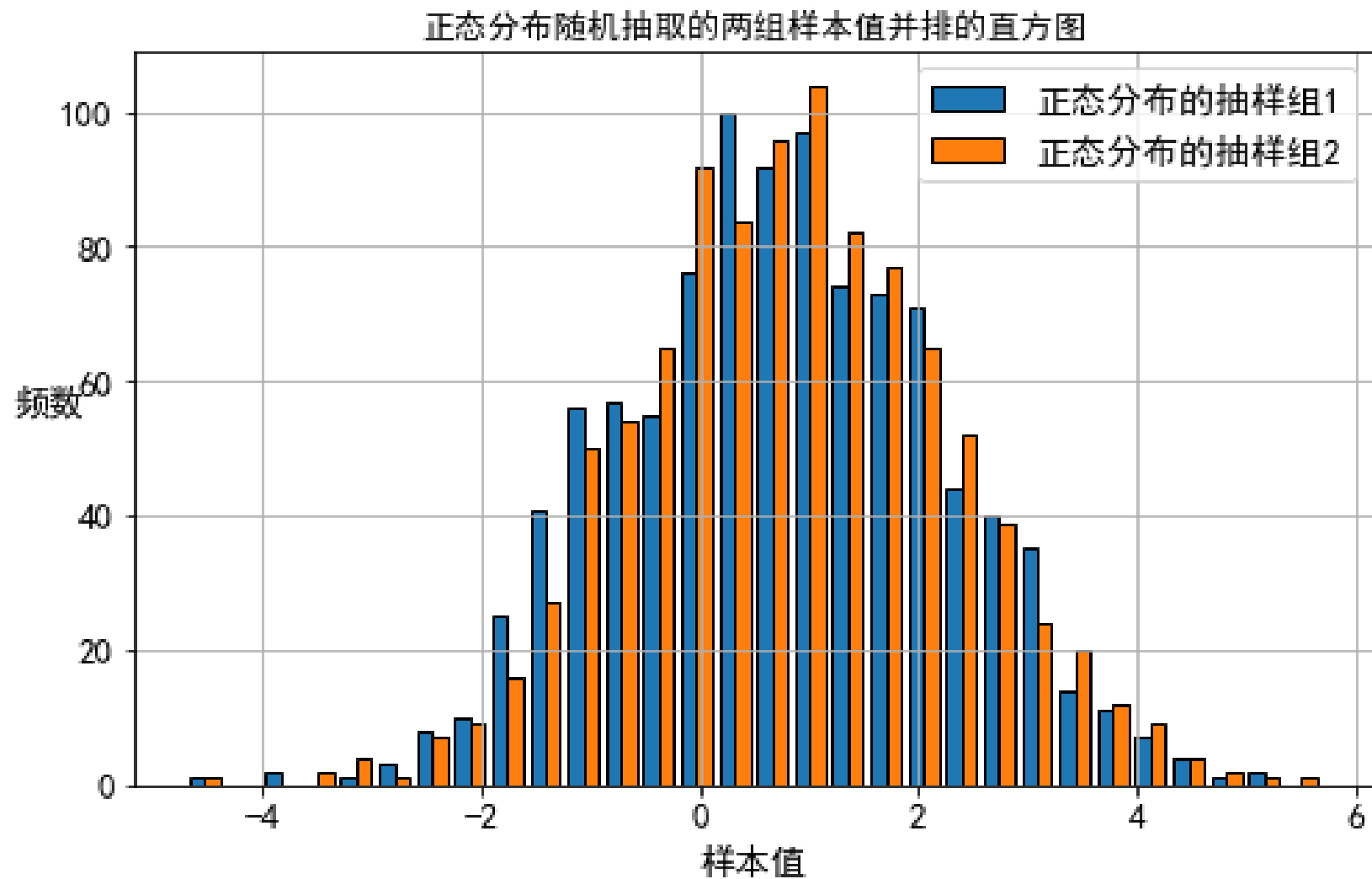
5.3.2 多个样本的直方图

2、并排展示

【例5-5】沿用例5-4中的两组样本值数据，通过直方图并且以并排的形式展示（见图5-5），具体的代码如下：

```
plt.figure(figsize=(8,5))
plt.hist(x_norml,label=[u'正态分布的抽样组1',u'正态分布的抽样组2'],bins=30,edgecolor='k')
plt.xticks(fontsize=13)
plt.xlabel(u'样本值',fontsize=13)
plt.yticks(fontsize=13)
plt.ylabel(u'频数',fontsize=13,rotation=0)
plt.title(u'正态分布随机抽取的两组样本值并排的直方图')
plt.legend(loc=0,fontsize=13)
plt.grid(True)
```


5.3.2 多个样本的直方图





5.4 条形图

5.4.1 垂直条形图

【例5-6】沿用4.1节的例4-1的信息，针对表5-5描述的2018年9月3日至9月7日这5个交易日中相关股票的涨跌幅情况，生成2018年9月3日-7日这4只股票的涨跌幅的条形图，并且以 2×2 子图的方式呈现，需要运用生成条形图的函数bar。具体代码如下：

股票简称	9月3日	9月4日	9月5日	9月6日	9月7日
中国石油	0.3731%	2.1066%	-0.4854%	0.6098%	-0.6060%
工商银行	-0.1838%	0.1842%	-1.6544%	-0.3738%	0.3752%
上汽集团	-0.3087%	-0.0344%	-3.3391%	0.7123%	0.4597%
宝钢股份	-2.4112%	1.1704%	-2.9563%	-1.4570%	1.6129%

5.4.1 垂直条形图

```
import pandas as pd
import numpy as np

return_list = [0.003731,    0.021066,    -0.004854,    0.006098,    -0.006060, -0.001838,    0.001842,    -
0.016544,    -0.003738,    0.003752, -0.003087,    -0.000344,    -0.033391,    0.007123,    0.004597, -0.024112,
               0.011704,    -0.029563,    -0.014570,    0.016129]

return_array = np.array(return_list)
return_array = return_array.reshape(4,5)
date=['2018-9-3','2018-9-4','2018-9-5','2018-9-6','2018-9-7']
stock=['中国石油','工商银行','上汽集团','宝钢股份']
return_dataframe=pd.DataFrame(data=return_array.T,index=date,columns=stock)

plt.figure(figsize=(12,10))
#####第1行第1个
plt.subplot(2,2,1)
plt.bar(x=return_dataframe.columns,height=return_dataframe.iloc[0],width=0.5,label=u'2018年9月3日涨跌幅',facecolor='y')
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.ylim(-0.035, 0.025)
plt.ylabel(u'涨跌幅',fontsize=13,rotation=90)
plt.legend(loc=0,fontsize=13)
plt.grid(True)
```

5.4.1 垂直条形图

```
#####第1行第2个
plt.subplot(2,2,2)
plt.bar(x=return_dataframe.columns,height=return_dataframe.iloc[2],width=0.5,label=u'2018年9月5日涨跌幅',facecolor='y')
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.ylim(-0.035, 0.025)
plt.ylabel(u'涨跌幅',fontsize=13,rotation=90)
plt.legend(loc=0,fontsize=13)
plt.grid(True)
#####第2行第1个
plt.subplot(2,2,3)
plt.bar(x=return_dataframe.columns,height=return_dataframe.iloc[3],width=0.5,label=u'2018年9月6日涨跌幅',facecolor='y')
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.ylim(-0.035, 0.025)
plt.ylabel(u'涨跌幅',fontsize=13,rotation=90)
plt.legend(loc=0,fontsize=13)
plt.grid(True)
```

5.4.1 垂直条形图

#####第2行第2个

```
plt.subplot(2,2,4)
```

```
plt.bar(x=return_dataframe.columns,height=return_dataframe.iloc[3],width=0.5,label=u'2018年9月7日涨跌幅',facecolor='y')
```

```
plt.xticks(fontsize=13)
```

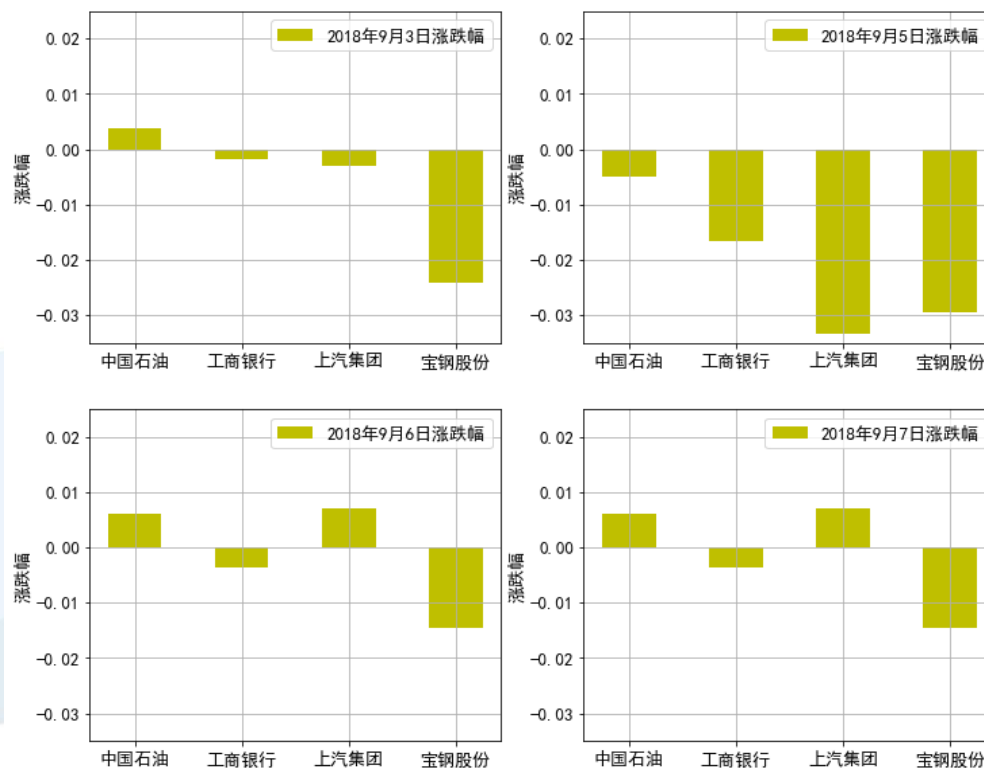
```
plt.yticks(fontsize=13)
```

```
plt.ylim(-0.035, 0.025)
```

```
plt.ylabel(u'涨跌幅',fontsize=13,rotation=90)
```

```
plt.legend(loc=0,fontsize=13)
```

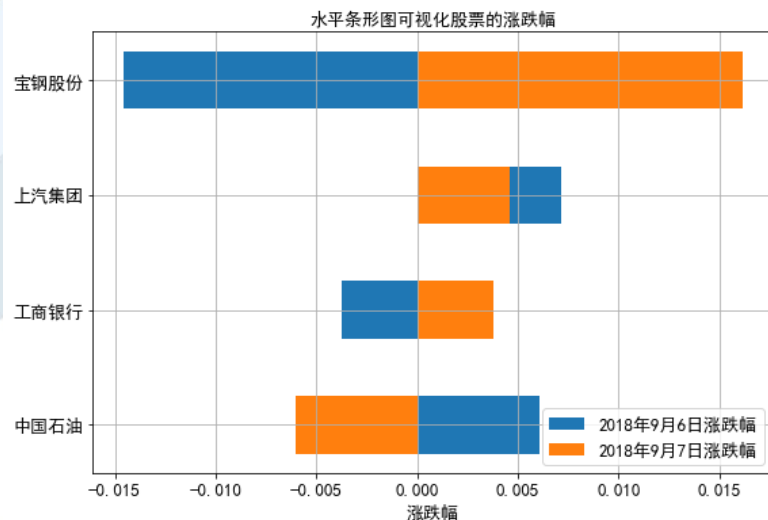
```
plt.grid(True)
```



5.4.2 水平条形图

【例5-7】沿用例5-6的相关信息，将2018年9月6日和9月7日这两个交易日的4只股票的涨跌幅放置在一张水平条形图中进行展示，需要运用生成水平条形图的函数`barh`，具体的代码如下：

```
plt.figure(figsize=(9,6))  
plt.barh(y=return_dataframe.columns,width=return_dataframe.iloc[3],height=0.5,label=u'2018年9月6日  
涨跌幅')  
plt.barh(y=return_dataframe.columns,width=return_dataframe.iloc[4],height=0.5,label=u'2018年9月7日  
涨跌幅')  
plt.xticks(fontsize=13)  
plt.xlabel(u'涨跌幅',fontsize=13)  
plt.yticks(fontsize=13)  
plt.title(u'水平条形图可视化股票的涨跌幅',fontsize=13)  
plt.legend(loc=0,fontsize=13)  
plt.grid(True)
```



5.5 散点图

【例5-8】以沪深300指数与上证180指数2016年至2018年的日涨跌幅作为分析对象，用于演示散点图的绘制方法，具体分为两个步骤。

第1步：从外部导入相关数据的 Excel表格并生成数据框，查看该数据框的相关特征及考察两个时间序列的相关系数，具体的代码如下：

```
import pandas as pd
```

```
HS300_sz180 = pd.read_excel('D:\Zhangzw\Python\Python金融数据分析\RawData\第5章\沪深300指数与上证180指数的日涨跌幅_2016_2018.xlsx',header=0,index_col=0) #注意导入的是sheet1
```

```
HS300_sz180.describe()
```

```
Out[18]:
```

```
    沪深300涨跌幅    上证180涨跌幅
```

```
count  731.000000  731.000000
```

```
mean   -0.000223  -0.000154
```

```
std     0.011826   0.011400
```

```
min     -0.070206  -0.067217
```

```
25%     -0.005144  -0.004876
```

```
50%      0.000350   0.000126
```

```
75%      0.005075   0.005022
```

```
max       0.043167   0.041031
```

5.8 散点图

```
HS300_sz180.corr()
```

```
Out[19]:
```

```
沪深300涨跌幅 上证180涨跌幅
```

```
沪深300涨跌幅 1.000000 0.985249
```

```
上证180涨跌幅 0.985249 1.000000
```

第2步：绘制这两个指数日涨跌幅的散点图。具体的代码如下：

```
plt.figure(figsize=(9,6))
```

```
plt.scatter(x=HS300_sz180.iloc[:,0],y=HS300_sz180.iloc[:,1],c='b',marker='o')
```

```
plt.xticks(fontsize=14)
```

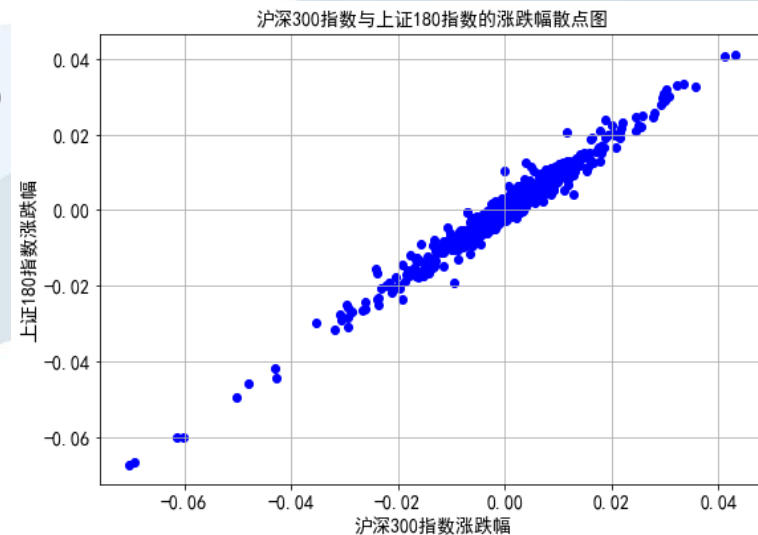
```
plt.xlabel(u'沪深300指数涨跌幅',fontsize=14)
```

```
plt.yticks(fontsize=14)
```

```
plt.ylabel(u'上证180指数涨跌幅',fontsize=14,rotation=90)
```

```
plt.title(u'沪深300指数与上证180指数的涨跌幅散点图',fontsize=14)
```

```
plt.grid()
```



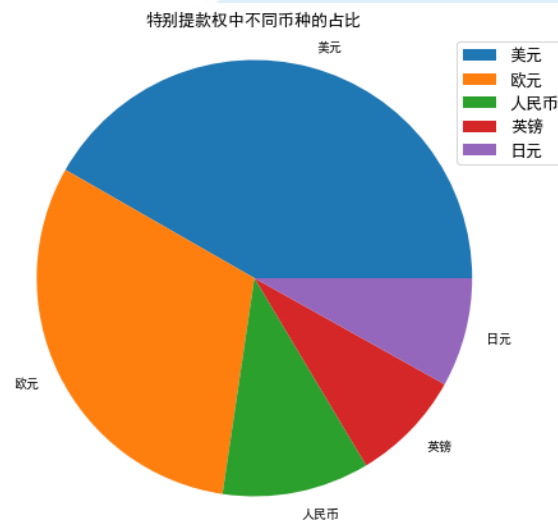
5.6 饼图

5.6 饼图

【例5-9】国际货币基金组织（IMF）的特别提款权被称为“纸黄金”，是该组织分配给会员国的一种使用资金的权利。目前，特别提款权的价值是由美元、欧元、人民币、日元、英镑等一篮子储备货币所决定，从2016年10月1日至今，这5种货币在特别提款权中的比重由表5-6所示。

币种	美元	欧元	人民币	日元	英镑
占比	41.73%	30.9%	10.92%	8.33%	8.09%

```
currency = ['美元','欧元','人民币','英镑','日元']
perc = [0.4173,0.3093,0.1092,0.0833,0.0809]
plt.figure(figsize=(9,7))
plt.pie(x=perc,labels=currency)
plt.axis('equal')
plt.legend(loc=1,fontsize=13)
plt.title(u'特别提款权中不同币种的占比',fontsize=13)
```



5.7 股价蜡烛图

5.7 蜡烛图

目前，`mplfinance`库实际上已经从`matplotlib`中独立出来了。更新后的`mplfinance`库较以往更加简单易用，增加了很多新功能（`renko`砖形图、`volume`柱形图、`ohlc`图等），且支持多种风格，可定制多种颜色（因为国外K线图颜色与国内相反，容易产生误导），线条（默认线条较粗，影响观感）等。

安装`mplfinance`库要求`pandas`和`matplotlib`，然后在prompt中操作：

```
pip install mplfinance --upgrade --user -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
import mplfinance as mpf
```

```
import pandas as pd
```

```
def import_csv():          # 导入股票数据
```

```
    df = pd.read_csv('D:\Zhangzw\Python\Python金融数据分析\RawData\第5章\A_20200801_20200831.csv')
```

```
    df.rename(
```

```
        columns={
```

```
            'date': 'Date', 'open': 'Open',
```

```
            'high': 'High', 'low': 'Low',
```

```
            'close': 'Close', 'volume': 'Volume'},
```

```
        inplace=True)
```

```
    df['Date'] = pd.to_datetime(df['Date']) # 转换为日期格式
```

```
    df.set_index(['Date'], inplace=True) # 将日期列作为行索引
```

```
    return df
```

5.7 蜡烛图

```
df = import_csv()
```

```
kwargs = dict(type='candle', mav=(2, 5, 7), volume=True, title='Stock Price', ylabel='price',  
ylabel_lower='volume',figratio=(10, 8), figscale=1.3, linecolor='g')
```

```
mc = mpf.make_marketcolors(# 设置marketcolors
```

```
    up='red', # up:设置K线线柱颜色, up意为收盘价大于等于开盘价
```

```
    down='green', # down:与up相反, 这样设置与国内K线颜色标准相符
```

```
    edge='i', # edge:K线线柱边缘颜色(i代表继承自up和down的颜色), 下同。详见官方文档)
```

```
    wick='i', # wick:灯芯(上下影线)颜色
```

```
    volume='in', # volume:成交量直方图的颜色
```

```
    inherit=True)# inherit:是否继承, 选填
```

```
mystyle = mpf.make_mpf_style(# 设置图形风格
```

```
    gridaxis='both', # gridaxis:设置网格线位置
```

```
    gridstyle='-.', # gridstyle:设置网格线线型
```

```
    y_on_right=False, # y_on_right:设置y轴位置是否在右
```

```
    marketcolors=mc)
```

```
mpf.plot(df, **kwargs, style=mystyle)
```

#如果我们不确定要往函数中传入多少个参数, 或者我们想往函数中以列表和元组的形式传参数时, 那就使要用*args;
如果我们不知道要往函数中传入多少个关键词参数, 或者想传入字典的值作为关键词参数时, 那就要使用**kwargs



**Your appreciation makes me a miracle.
Thank you!**

**Frank Ziwei Zhang
18117228563
frank8027@163.com**



上海對外經貿大學
SHANGHAI UNIVERSITY OF INTERNATIONAL BUSINESS AND ECONOMICS