##包括散点图，折线统计图，柱状图，直方图，饼图的绘制

1，散点图的绘制

```

python

import matplotlib.pyplot as plt

import numpy as np

plt.rcParams['font.family'] = 'SimHei' #设置中文

plt.rcParams['axes.unicode\_minus'] = False #使负数可以显示

n = 1024

x = np.random.normal(0,1,n)

y = np.random.normal(0,1,n) #生成1024个正态分布的点

plt.scatter(x, y,s = 12,c= 'blue',marker ='o',label= "正态分布")

#绘制散点图，颜色为blue,绘制的点为’.',c代表颜色，s代表点的大小,label为增加图例

plt.legend() #与label搭配使用，显示图例

plt.title('散点图',fontsize = 28) #设置标题，字号

plt.xlabel('X轴') #x轴的名字

plt.ylabel('Y轴')

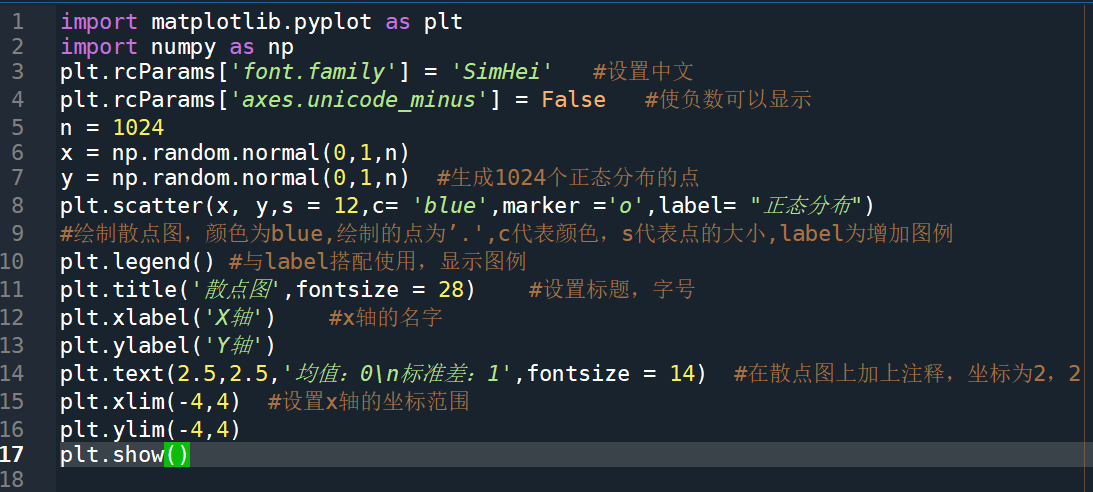
plt.text(2.5,2.5,'均值：0\n标准差：1',fontsize = 14) #在散点图上加上注释，坐标为2，2

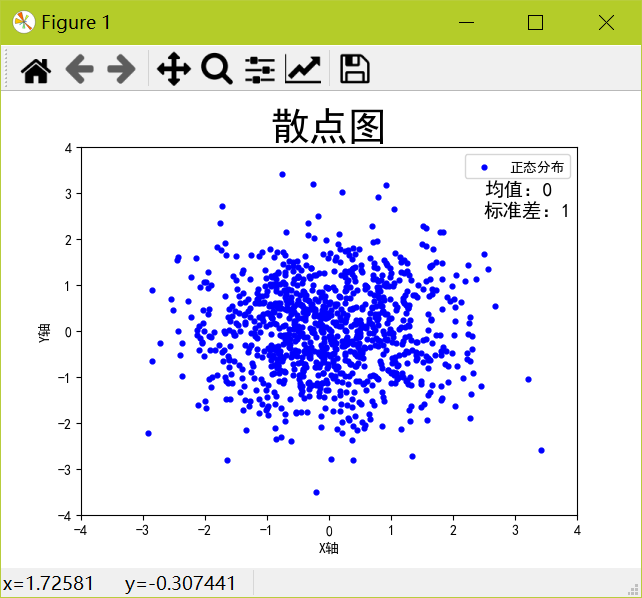
plt.xlim(-4,4) #设置x轴的坐标范围

plt.ylim(-4,4)

plt.show()

```





2，折线统计图

```

python

#绘制24小时温度折线统计图

import matplotlib.pyplot as plt

import numpy as np

plt.rcParams['font.family'] = 'SimHei' #设置中文

plt.rcParams['axes.unicode\_minus'] = False #使负数可以显示

n =24

y1 = np.random.randint(30,40,n)

y2 = np.random.randint(10,30,n)#生成两个随机数列y1,y2

plt.plot(y1,c ='r',label ='长沙') #设置图例，c和’r’是color和red的缩写

plt.plot(y2,c ='b',label ='哈尔滨') #x可以省略，默认0，1....

plt.xlim(0,24)

plt.ylim(0,50) #设置横纵坐标轴取值范围

plt.xlabel('时间',fontsize =12)

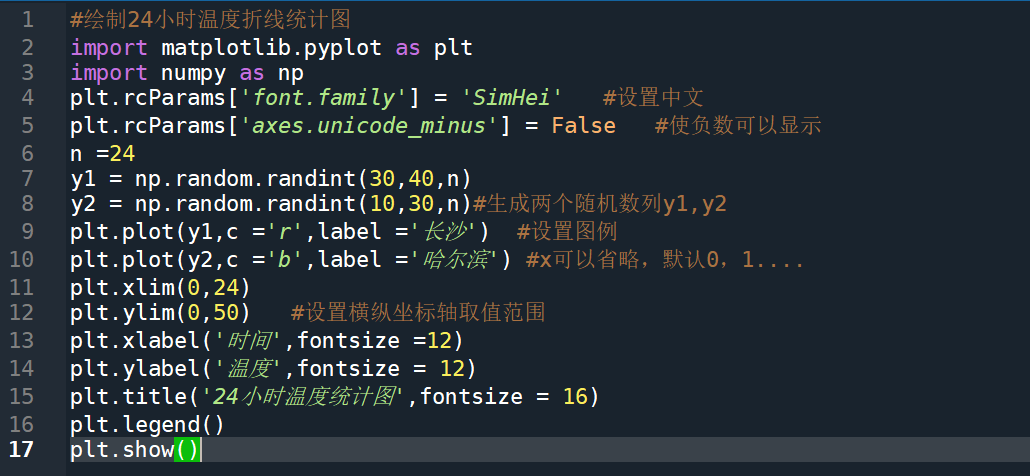
plt.ylabel('温度',fontsize = 12)

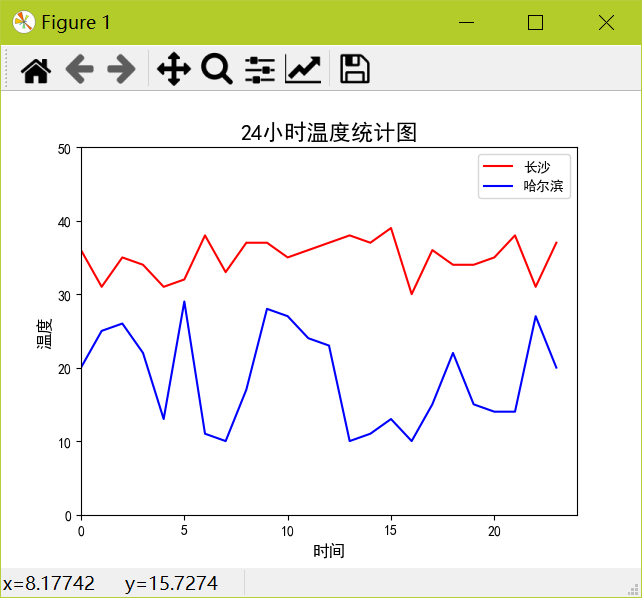
plt.title('24小时温度统计图',fontsize = 16)

plt.legend()

plt.show()

```





3，柱状图的绘制。

柱状图要用到bar函数

```

python

plt.bar(left,width,height,facecolor,edgecolor,label)

```

其中，left为直方图左边的坐标值（通常取x值），width代表直方图的宽度，默认为0.8，隔一厘米绘制一个直方图。height为直方图的高度（通常取y）。left,height均不能省略

```

python

import matplotlib.pyplot as plt

import numpy as np

plt.rcParams['font.family'] = 'SimHei' #设置中文

plt.rcParams['axes.unicode\_minus'] = False #使负数可以显示

n =8

y1 = np.random.randint(5,30,n) #生成两个随机数列y1,y2,

y2 = np.random.randint(-30,-10,n)#通常需要导入数据，而不是生成随机数列，

plt.bar(range(len(y1)),y1,facecolor = 'g',edgecolor = 'w',label = '统计量1')

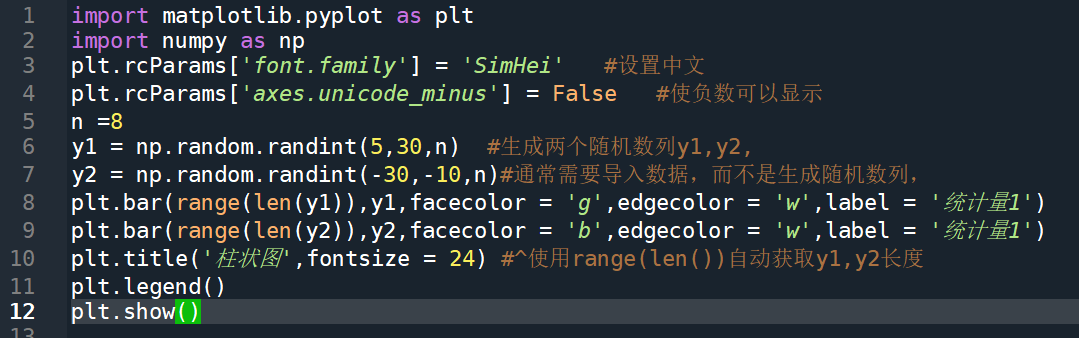
plt.bar(range(len(y2)),y2,facecolor = 'b',edgecolor = 'w',label = '统计量1')

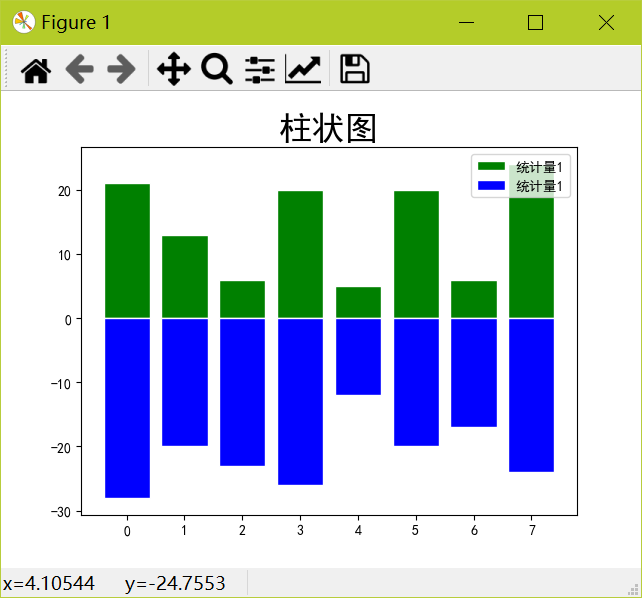
plt.title('柱状图',fontsize = 24) #^使用range(len())自动获取y1,y2长度

plt.legend()

plt.show()

```





4,直方图的绘制

hist和bar的区别：

hist是制作一个频率分布图，比如说把一个数据分成10个部分，每个部分的频率是多少。就是大概看一下数据的分布。 bar是用来把已经总结好的数据画出来，可以用来对比各个组的数据。 hist只是专门制作频率分布的，而bar的应用范围很广，可以同时对比多个组。

直方图的绘制需要调用到hist函数，调用方式为：

```

python

plt.hist(arr,bins,normed,facecolor,edgecolor,alpha,histtype)

```

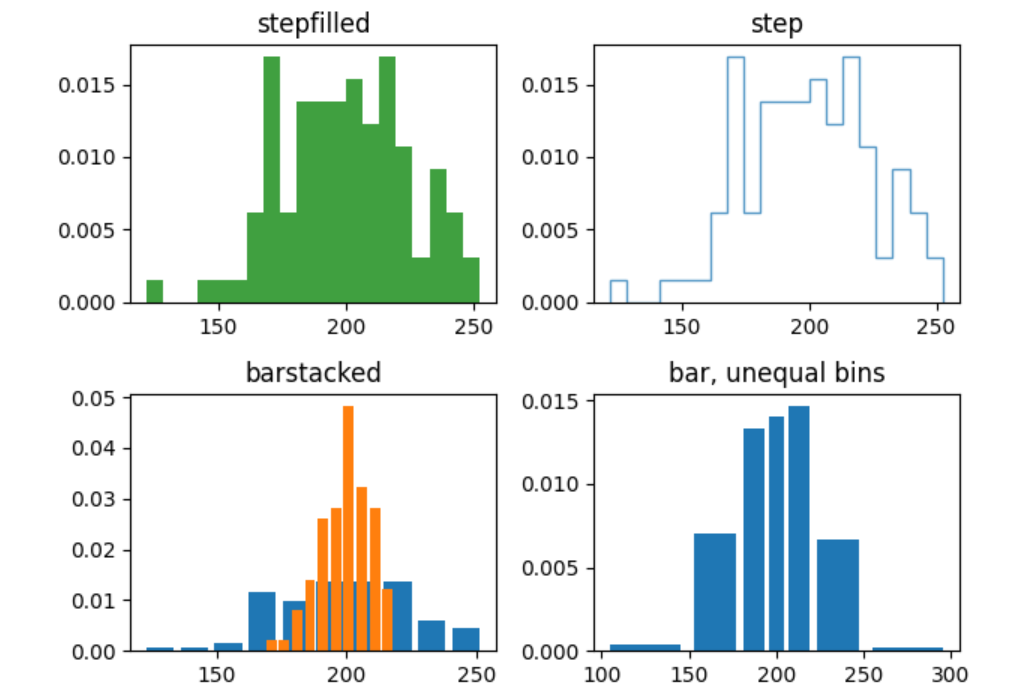
arr为必需的，指需要绘出的数据，而其它的都有默认值。

bins默认值为10，表示绘制的直方图的柱子数量，bins越大，反映的数据分布越精细

normed 默认值为（False）0，表示是否归一化，如果设为1，纵坐标将变为直方图的百分比概率。

alpha为透明度；

histtype指直方图的类型，‘bar’, ‘barstacked’, ‘step’, ‘stepfilled‘；默认为’bar’



示例：

```

python

import matplotlib.pyplot as plt

import numpy as np

plt.rcParams['font.family'] = 'SimHei' #设置中文

plt.rcParams['axes.unicode\_minus'] = False #使负数可以显示

np.random.seed(0)

mu,sigma = 100,20 #设定均值和方差

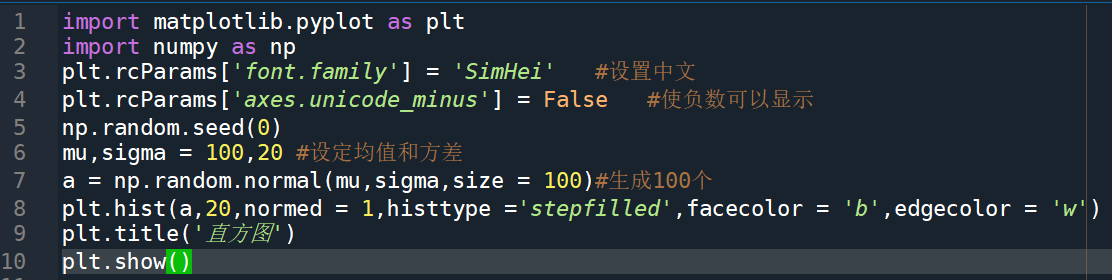
a = np.random.normal(mu,sigma,size = 100)#生成100个

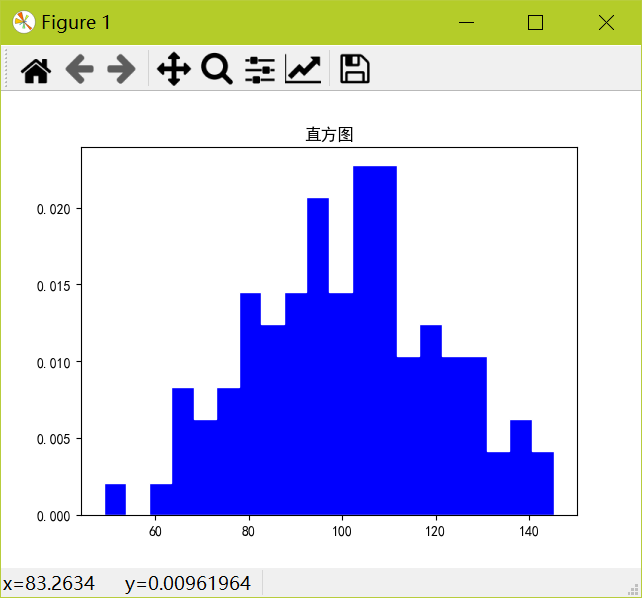
plt.hist(a,20,normed = 1,histtype ='stepfilled',facecolor = 'b',edgecolor = 'w')

plt.title('直方图')

plt.show()

```





5,饼图的绘制

饼图的绘制需要用到pie函数

其调用方式为

```

python

plt.pie(x,labels,explode,autopct,shadow,startangle)

```

x :(每一块)的比例，如果sum(x) > 1会使用sum(x)归一化；

labels :(每一块)饼图外侧显示的说明文字；

explode :(每一块)离开中心距离；

autopct :控制饼图内百分比设置,可以使用format字符串或者format function

'%1.1f'指小数点前后位数(没有用空格补齐)；

startangle :起始绘制角度,默认图是从x轴正方向逆时针画起,如设定=90则从y轴正方向画起；

shadow :在饼图下面画一个阴影。默认值：False，即不画阴影；

实例：

```

python

import matplotlib.pyplot as plt

import numpy as np

plt.rcParams['font.family'] = 'SimHei' #设置中文

plt.rcParams['axes.unicode\_minus'] = False #使负数可以显示

labels = '东部','西部','南部','中部','北部'

x = [15,30,35,5,15] #设置对应比例

explode =(0,0.1,0,0,0) #使第二个突出0.1

plt.pie(x,explode = explode,labels = labels,autopct = '%1.1f%%',shadow =True,startangle =90)

plt.axis('equal') #使饼图长宽相等

plt.title('公司销售分布区域图')

plt.show()

```

