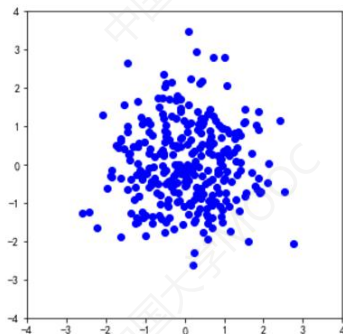




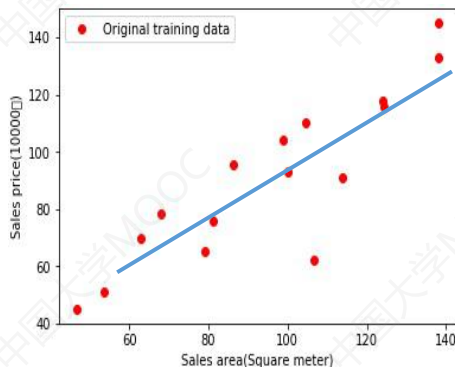
## 6.2 散点图

■ **散点图** (Scatter) : 是数据点在直角坐标系中的分布图

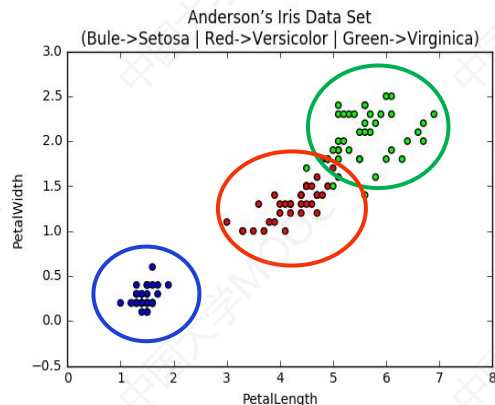
□ 数据分布规律



□ 数据变化趋势



□ 数据分组



## □ scatter() 函数

```
scatter( x, y, scale, color, marker, label )
```

参 数	说 明	默认值
x	数据点的x坐标	不可省略
y	数据点的y坐标	不可省略
scale	数据点的大小	36
color	数据点的颜色	
marker	数据点的样式	'o' (圆点)
label	图例文字	

## color参数——常用颜色

颜 色	缩略字符	颜 色	缩略字符
blue	b	black	k
green	g	white	w
red	r	cyan	c
yellow	y	magenta	m



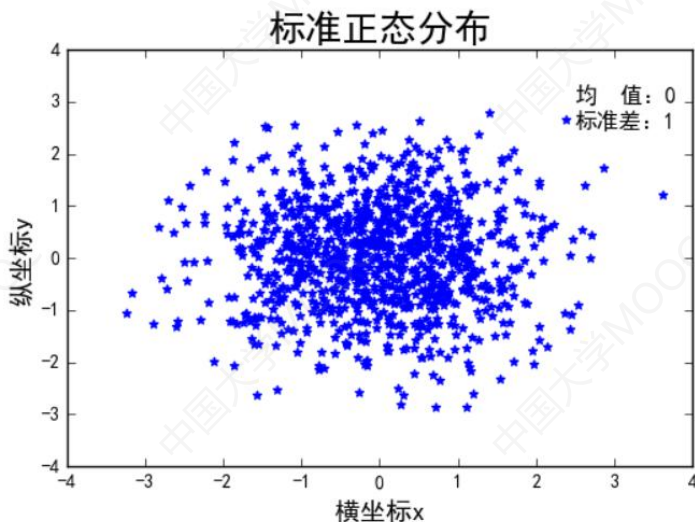
## marker参数——数据点样式

取值	中文描述	取值	中文描述	取值	中文描述
-	实线	1	朝下的三角	v	朝下的三角形
--	虚线	2	朝上的三角	^	朝上的三角形
-.	点线	3	朝左的三角	<	朝左的三角形
:	点虚线	4	朝右的三角	>	朝右的三角形
.	点	s	正方形	D	钻石形
,	像素	p	五角形	d	小版钻石形
o	圆形	*	星型		垂直线形
+	+号标记	h	1号六角形	—	水平线行
x	x号标记	H	2号六角形		



## 6.2 散点图

例：绘制**标准正态分布**的散点图



□ 设置默认字体为中文黑体

```
plt.rcParams['font.sans-serif']="SimHei"
```

□ 标准正态分布的散点坐标

```
n=1024
```

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

```
y = np.random.normal(0,1,n)
```

□ 绘制散点图

```
plt.scatter(x, y, color="blue",marker='*')
```

□ 设置标题

```
plt.title("标准正态分布",fontsize=20)
```



### □ 添加文字——`text()` 函数

```
text( x, y, s, fontsize,color )
```

参 数	说 明	默认值
x	文字的x坐标	不可省略
y	文字的y坐标	不可省略
s	显示的文字	不可省略
fontsize	文字的大小	12
color	文字的颜色	黑色

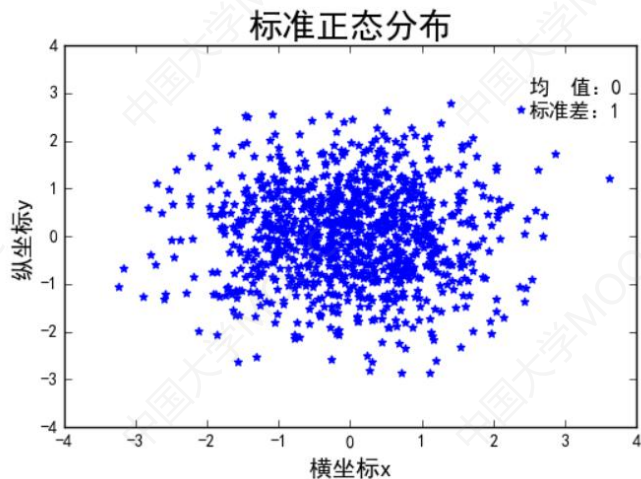
### □ 坐标轴设置

```
plt.rcParams["axes.unicode_minus"] = False
```

函数	说 明
xlabel( x, y, s, fontsize,color)	设置x轴标签
ylabel( x, y, s, fontsize,color)	设置y轴标签
xlim(xmin, xmax)	设置x轴坐标的范围
ylim(ymin, ymax)	设置y轴坐标的范围
tick_params(labelsize)	设置刻度文字的字号



### 例：绘制标准正态分布的散点图



#### □ 设置文本

```
plt.text(2.5,2.5,"均 值: 0\n标准差: 1")
```

#### □ 设置坐标轴范围

```
plt.xlim(-4,4)  
plt.ylim(-4,4)
```

#### □ 设置坐标轴标签

```
plt.xlabel('横 坐标x', fontsize=14)  
plt.ylabel('纵 坐标y', fontsize=14)
```





## 6.2 散点图

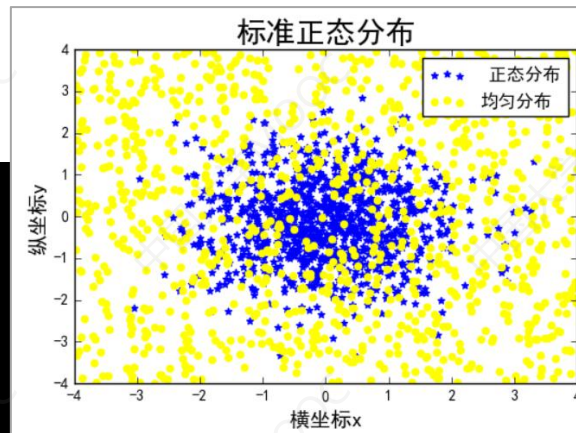
```
1 import matplotlib.pyplot as plt          # 导入绘图库
2 import numpy as np                      # 导入numpy库
3
4 plt.rcParams['font.sans-serif']="SimHei"  # 设置中文黑体为默认字体
5 plt.rcParams['axes.unicode_minus']=False  # 正常显示负号
6
7 n = 1024                                # 随机点个数: 1024
8 x = np.random.normal(0,1,n)             # 生成数据点x坐标
9 y = np.random.normal(0,1,n)             # 生成数据点y坐标
10
11 plt.scatter(x, y, color="blue",marker='*') # 绘制数据点
12
13 plt.title("标准正态分布",fontsize=20)    # 设置标题
14 plt.text(2.5,2.5,"均值: 0\n标准差: 1")  # 显示文本
15
16 plt.xlim(-4,4)                          # x轴范围
17 plt.ylim(-4,4)                          # y轴范围
18
19 plt.xlabel('横坐标x', fontsize=14)      # 设置x轴标签文本
20 plt.ylabel('纵坐标y', fontsize=14)      # 设置y轴标签文本
21
22 plt.show()                             # 显示绘图
```





**例：**绘制**标准正态分布**、**均匀分布**的散点图

```
1  n = 1024
2  x1 = np.random.normal(0,1,n)
3  y1 = np.random.normal(0,1,n)
4
5  x2=np.random.uniform(-4,4,(1,n))
6  y2=np.random.uniform(-4,4,(1,n))
7
8  plt.scatter(x1, y1, color="blue",marker='*')
9  plt.scatter(x2, y2, color="yellow",marker='o')
```



## 6.2 散点图

### 增加图例

指定图例内容

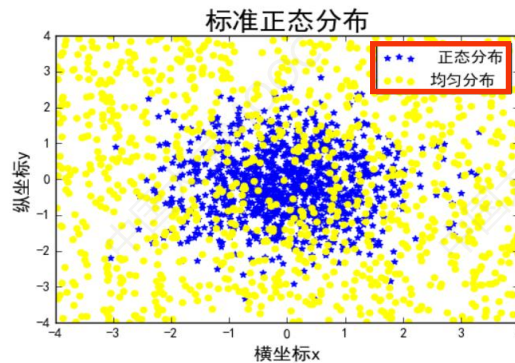
```
scatter( x, y, scale, color, marker, label )
```

```
legend(loc, fontsize)
```

显示图例

loc参数——指定图例的位置

取值	图例位置	取值	图例位置
0	best	6	center left
1	upper right	7	center right
2	upper left	8	lower center
3	lower left	9	upper center
4	lower right	10	center
5	right		



## 6.2 散点图

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 plt.rcParams['font.sans-serif']="SimHei"
5 plt.rcParams['axes.unicode_minus']=False
6
7 n = 1024
8 x1 = np.random.normal(0,1,n)
9 y1 = np.random.normal(0,1,n)
10
11 x2=np.random.uniform(-4,4,(1,n))
12 y2=np.random.uniform(-4,4,(1,n))
13
14
15 plt.scatter(x1, y1, color="blue",marker='*',label=" 正态分布" )
16 plt.scatter(x2, y2, color="yellow",marker='o' ,label="均匀分布")
17
18 plt.legend()
19 plt.title("标准正态分布",fontsize=20)
20
21 plt.xlim(-4,4)
22 plt.ylim(-4,4)
```

