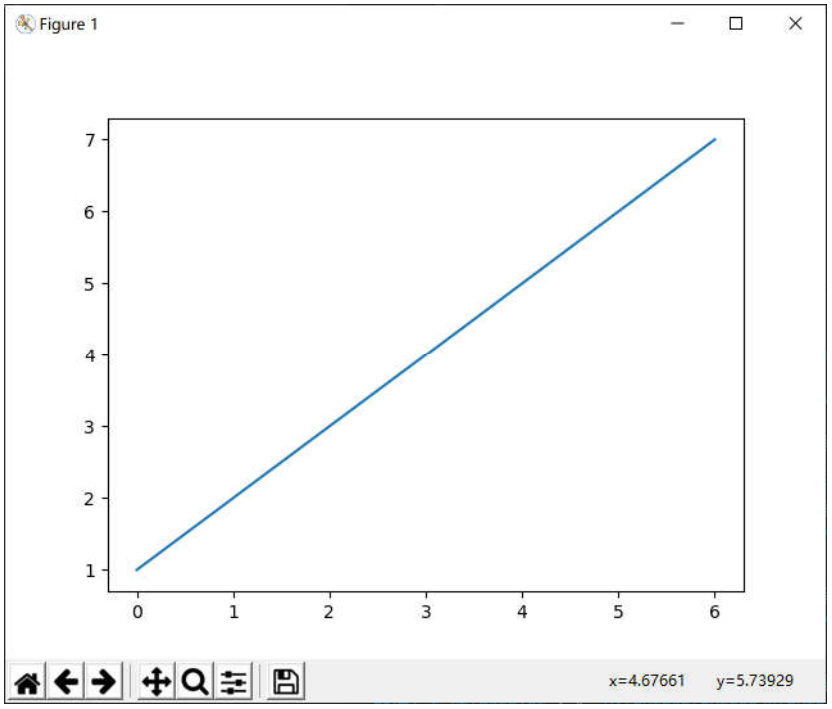
Python与人工智能实践



算法篇：

matplotlib 的基本使用

于泓
鲁东大学
信息与电气工程学院

Matplotlib简介

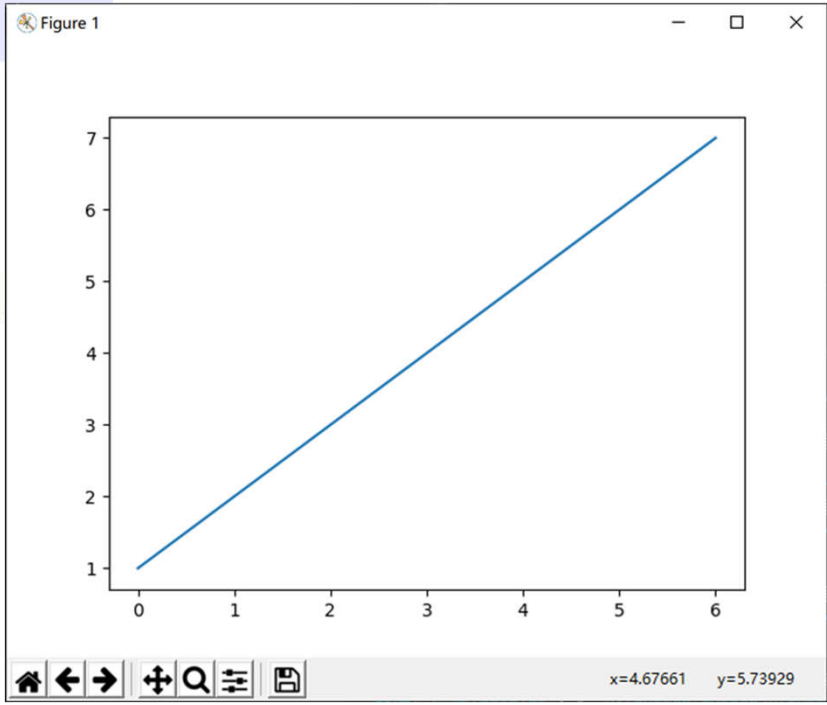
matplotlib是一个用于创建出版质量图表的桌面绘图包（主要是2D方面。

该项目是由John Hunter于2002年启动的， 其目的是为Python构建一个MATLAB式的绘图接口。

matplotlib支持各种操作系统上许多不同的GUI后端， 而且还能将图片导出为各种常见的矢量（vector） 和光栅（raster） 图：PDF、 SVG、 JPG、 PNG、 BMP、 GIF等

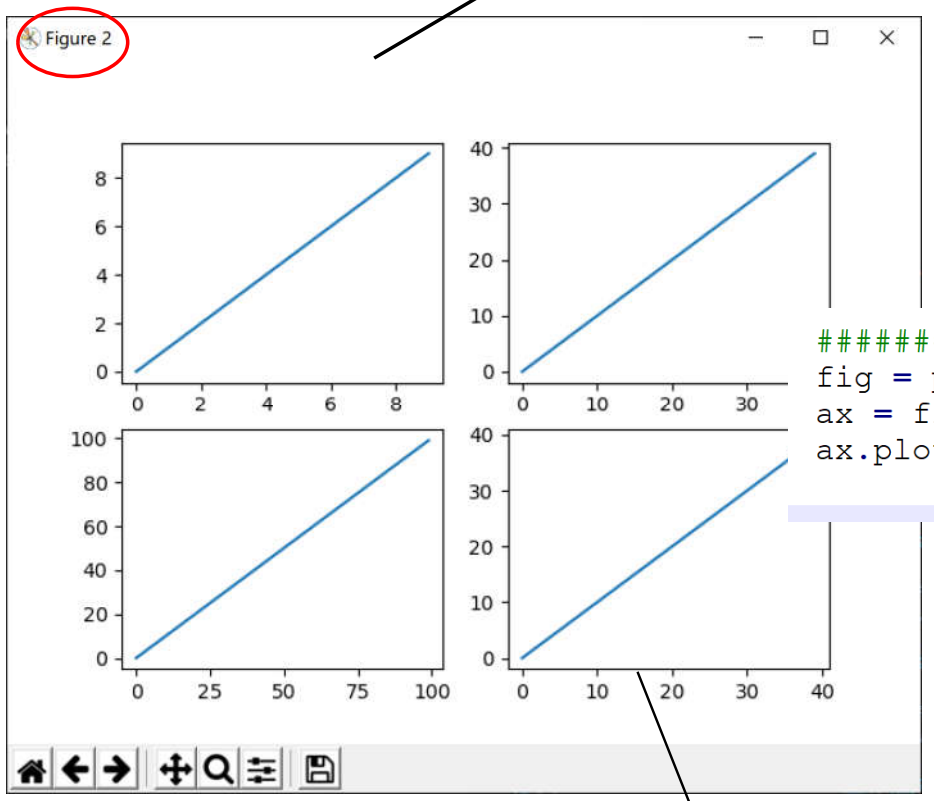
最基本的代码

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3
4  if __name__ == "__main__":
5      data = np.array([1,2,3,4,5,6,7])
6      plt.plot(data)
7      plt.show()
```



分屏幕显示

图像整体 **figure** 对象



整体图的上标

```
# 分屏显示
fig = plt.figure(2)
ax1 = fig.add_subplot(2, 2,1)
ax1.plot(np.arange(10))
                  add_subplot(2, 2, 2)
                  p.arange(40))
                  add_subplot(2, 2, 3)
                  p.arange(100))
ax4 = fig.add_subplot(2, 2, 4)
ax4.plot(np.arange(40))


plt.show()
```

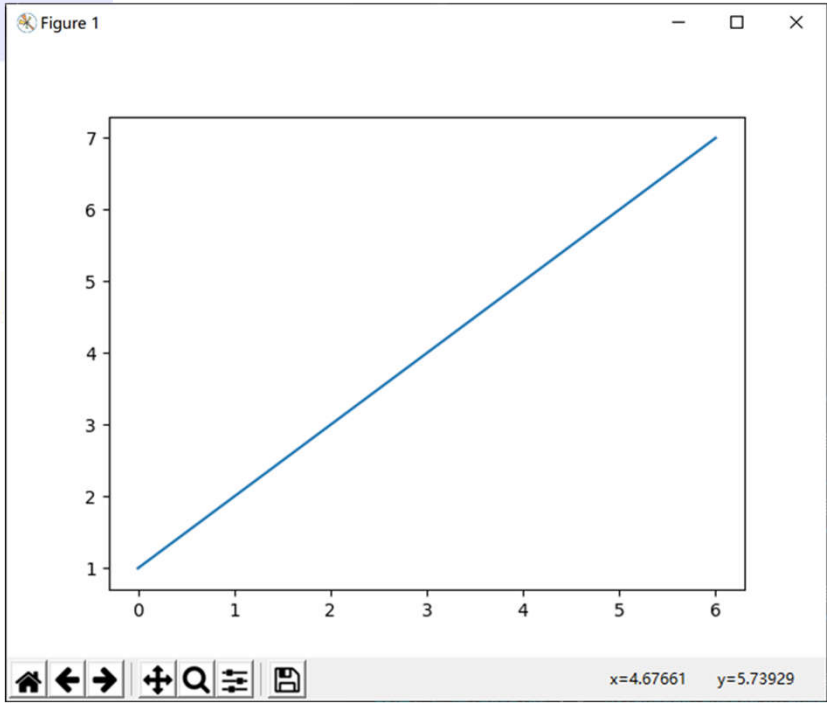
```
#######默认##############
fig = plt.figure(1)
ax = fig.add_subplot(1, 1,1)
ax.plot(data)
```

一般在代码中用**ax**来表示　　subplot

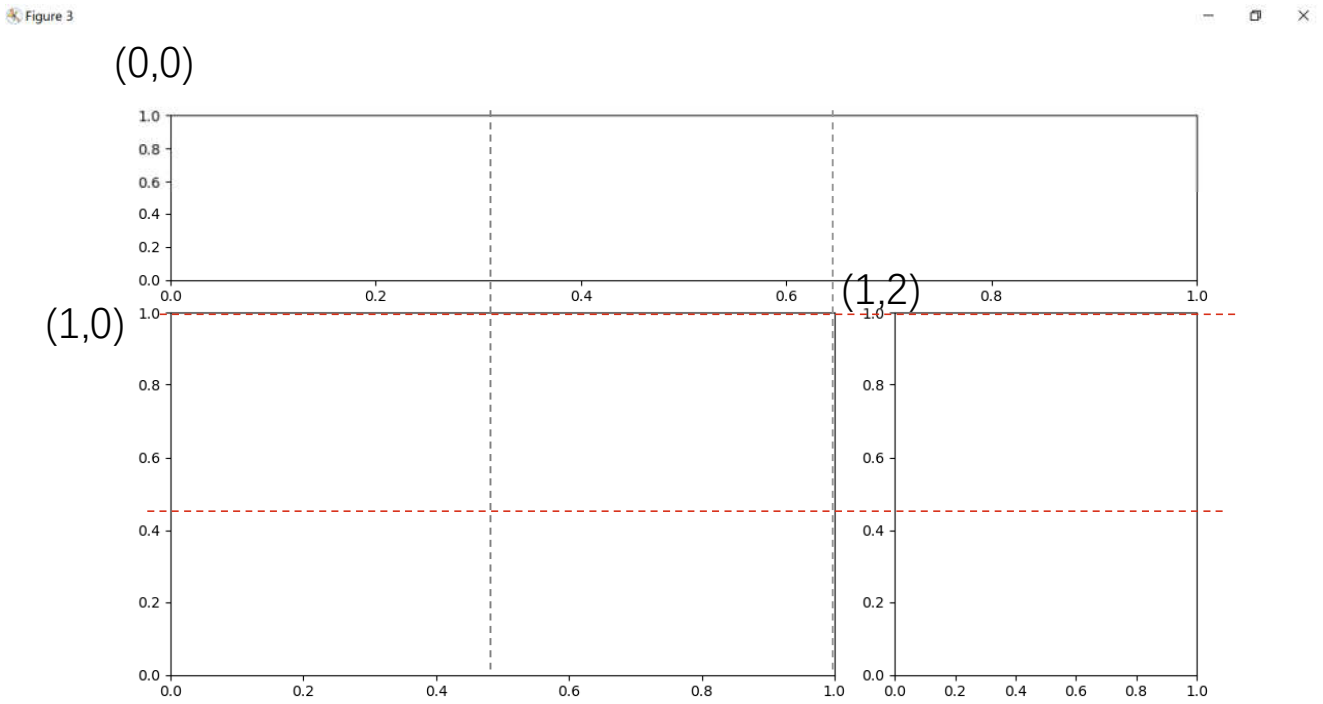
ax 就是用来绘图的区域

最基本的代码

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3
4  if __name__ == "__main__":
5      data = np.array([1,2,3,4,5,6,7])
6      plt.plot(data)
7      plt.show()


       #######默认###############
       fig = plt.figure(1)
       ax = fig.add_subplot(1, 1,1)
       ax.plot(data)
```



```
# 分屏2 不规则布局
fig = plt.figure(3)
ax1 = plt.subplot2grid((3,3),(0,0),colspan=3)
ax2 = plt.subplot2grid((3,3),(1,0),colspan=2,rowspan=2)
ax3 = plt.subplot2grid((3,3),(1,2),rowspan=2)
```



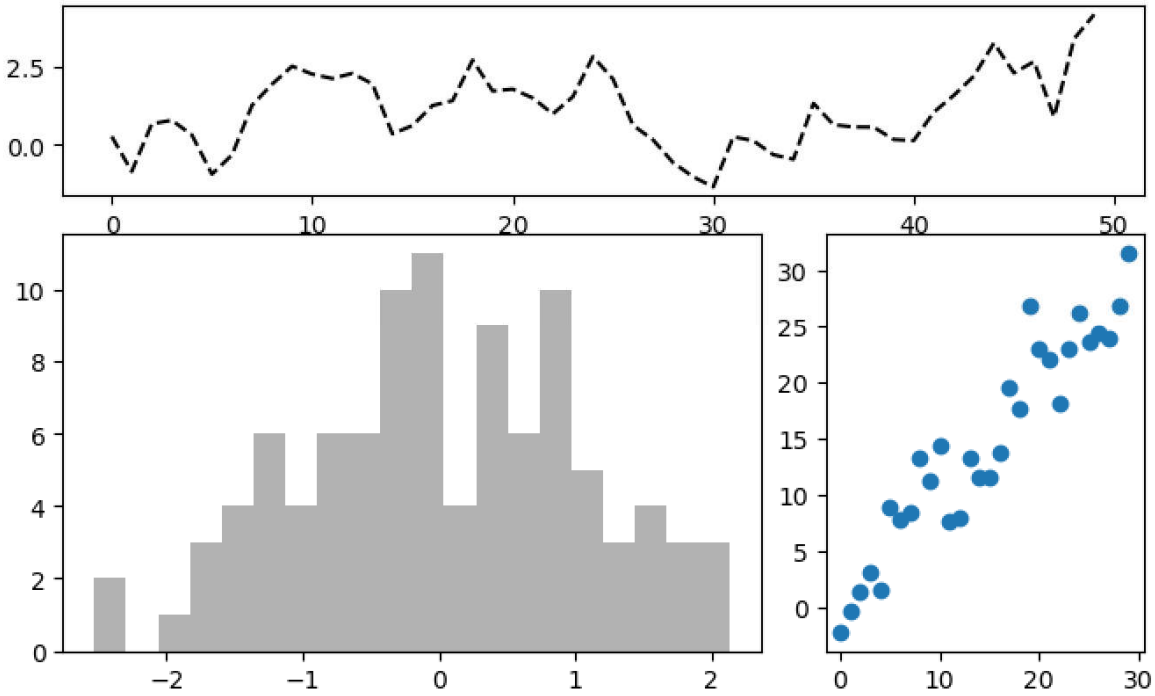
在ax内进行绘图

画线

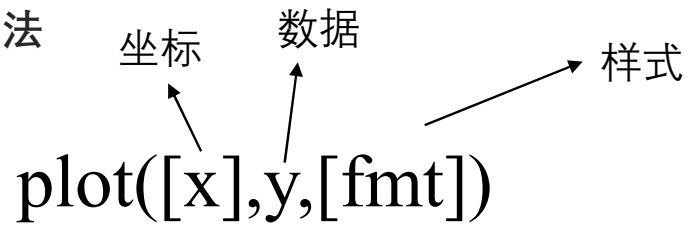
画直方图

画散点

```
# 分屏2 不规则布局
fig = plt.figure(3)
ax1 = plt.subplot2grid((3,3),(0,0),colspan=3)
ax2 = plt.subplot2grid((3,3),(1,0),colspan=2,rowspan=2)
ax3 = plt.subplot2grid((3,3),(1,2),rowspan=2)

ax1.plot(np.random.randn(50).cumsum(), 'k--')
ax2.hist(np.random.randn(100), bins=20, color='k', alpha=0.3)
ax3.scatter(np.arange(30), np.arange(30) + 3 * np.random.randn(30))
```

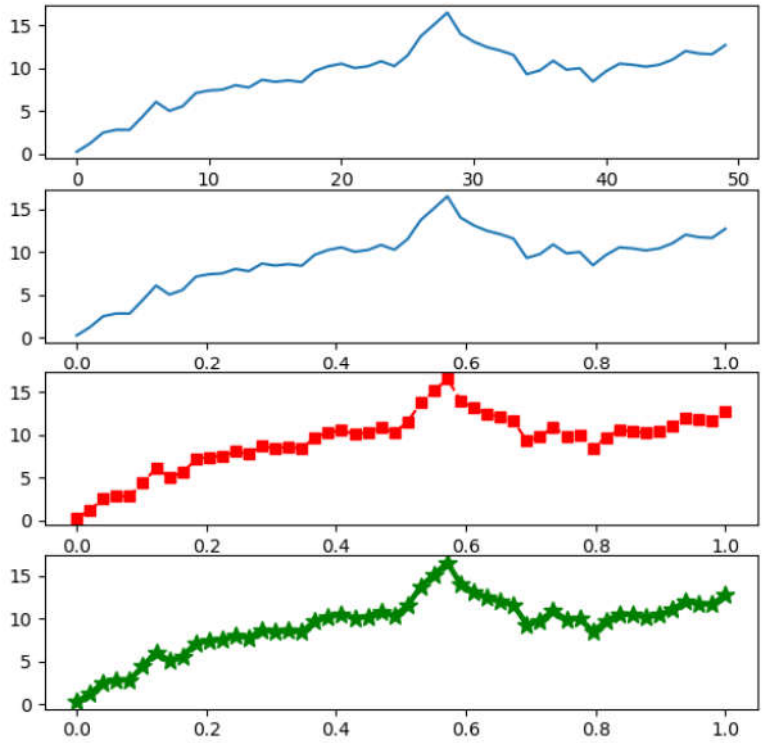


Plot的基本用法



样式
- 颜色
- 线型: 直线，虚线、点划线
- 标记： *、三角、方块

```
# plot 函数的样例
data = np.random.randn(50).cumsum()
fig = plt.figure(4)
# 直接画
ax1 = fig.add_subplot(4, 1,1)
ax1.plot(data)

# 添加横坐标
ax2 = fig.add_subplot(4, 1,2)
x = np.linspace(0,1,50)
ax2.plot(x,data)

# 添加横坐标+线型（格式 [color][marker][line]）
ax3 = fig.add_subplot(4, 1,3)
ax3.plot(x,data,'rs--')

# 添加横坐标+线型（格式 [color][marker][line]）
# + 更为详细的参数
ax4 = fig.add_subplot(4, 1,4)
ax4.plot(x,data,color=(0,0.5,0),marker ='*',linestyle = '-',linewidth=3, markersize=10)
```



默认颜色

character	color
``'b'``	blue 蓝
``'g'``	green 绿
``'r'``	red 红
``'c'``	cyan 蓝绿
``'m'``	magenta 洋红
``'y'``	yellow 黄
``'k'``	black 黑
``'w'``	white 白

默认的 marker

character	description
``'.'``	point marker
``','``	pixel marker
``'o'``	circle marker
``'v'``	triangle_down marker
``'^'``	triangle_up marker
``'<'``	triangle_left marker
``'>'``	triangle_right marker
``'1'``	tri_down marker
``'2'``	tri_up marker
``'3'``	tri_left marker
``'4'``	tri_right marker
``'s'``	square marker
``'p'``	pentagon marker
``'*'``	star marker
``'h'``	hexagon1 marker
``'H'``	hexagon2 marker
``'+'``	plus marker
``'x'``	x marker
``'D'``	diamond marker
``'d'``	thin_diamond marker
``'|'``	vline marker
``'_'``	hline marker

默认的线型

character	description
``'-'``	solid line style 实线
``'--'``	dashed line style 虚线
``'-.'``	dash-dot line style 点画线
``':'``	dotted line style 点线

直方图

数据

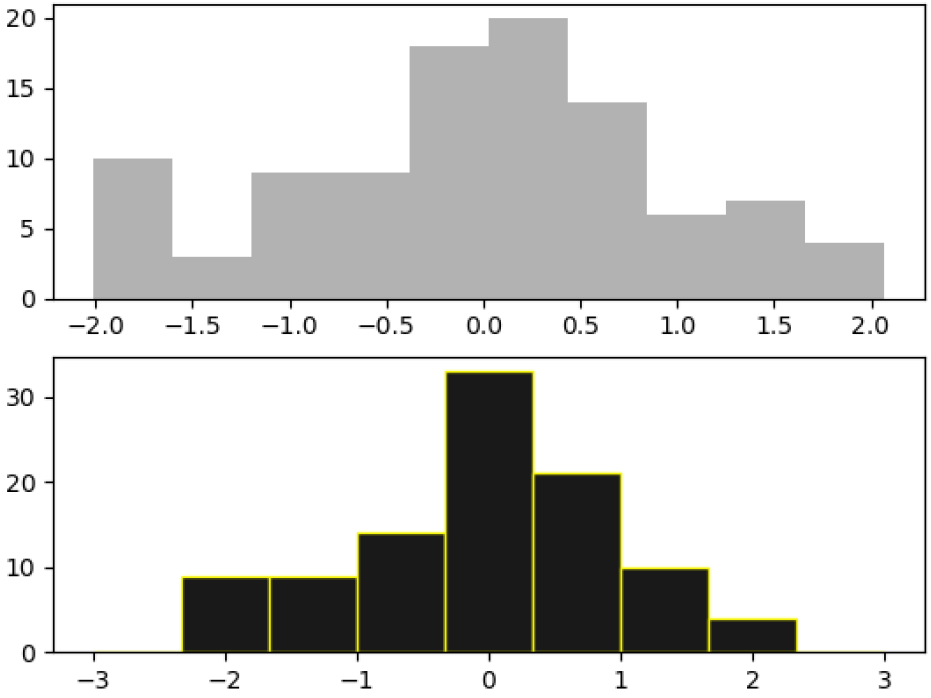
直方图区间
整数/序列

bins=10 把数据的值域平均分成个10个区间进行统计

bins =[1,3,5,7,9] 区间分别为[1,3)、 [3,5)、[5,7),[7,9)

hist(data, bins, color, alpha=0.3)

透明度



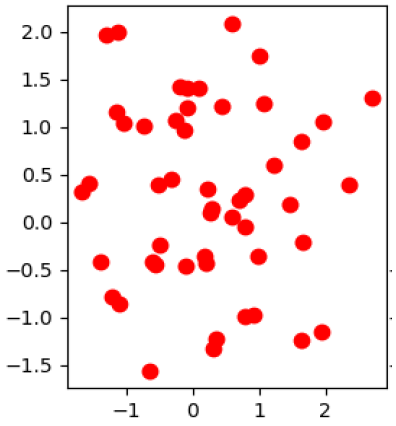

```
# 直方图示例
data = np.random.randn(100)

fig = plt.figure(5)

# 整数bin
ax1 = fig.add_subplot(2,1,1)
ax1.hist(data, bins=10, color='k', alpha=0.3)

# 区间bin
ax2 = fig.add_subplot(2,1,2)
m_bin = np.linspace(-3,3,10)
ax2.hist(data, bins=m_bin,ec="yellow", fc="k",alpha=0.9)
```

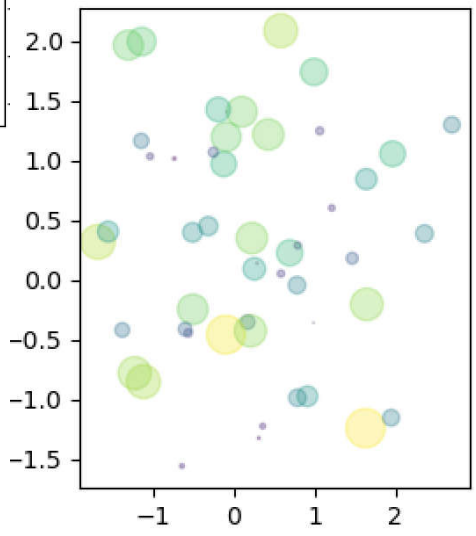
颜色设置时
还可以指定边缘颜色
以及柱体颜色

scatter 绘制散点图

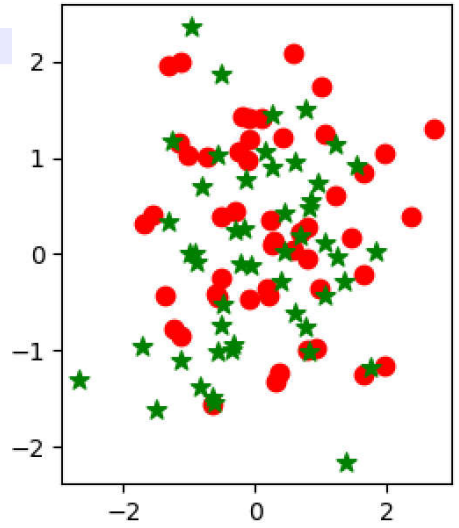
scatter(x, y, s=None, c=None, marker=None, cmap=None)

横纵坐标　点的尺寸　点的颜色　点的形状　色表（与c一起表示颜色）

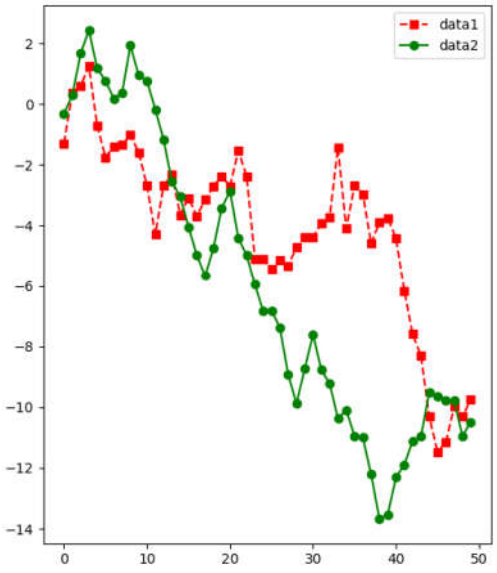




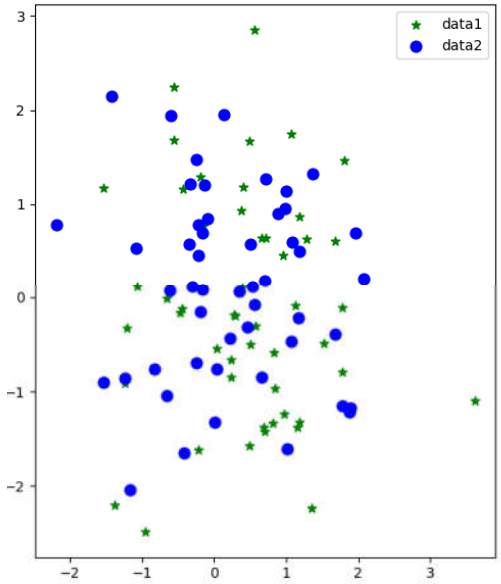
```
# 散点图示例
data = np.random.randn(2,50)
x = data[0,:]
y = data[1,:]
value = np.random.rand(50)
fig = plt.figure(6)
ax1 = fig.add_subplot(1,3,1)
ax2 = fig.add_subplot(1,3,2)
ax3 = fig.add_subplot(1,3,3)

# 简单显示
ax1.scatter(x,y,s=56,c='r',marker='o')

# 颜色以及点的尺寸会随value的变化而变化
sizes = ((value)*16)**2
ax2.scatter(x,y,s=sizes,c=value,marker='o',cmap='viridis',alpha=0.3)

# 显示两组数据
ax3.scatter(x,y,s=56,c='r',marker='o')
data = np.random.randn(2,50)
x = data[0,:]
y = data[1,:]
ax3.scatter(x,y,s=66,c='g',marker='*')
```



图例： 可以在绘图函数中添加label 属性 再调用 legend(loc='best') 进行显示




```
# 图例的示例
fig = plt.figure(7)
ax1 = fig.add_subplot(1,2,1)
ax2 = fig.add_subplot(1,2,2)
# 多plot
data1 = np.random.randn(50).cumsum()
data2 = np.random.randn(50).cumsum()
ax1.plot(data1,'rs--',label = 'data1')
ax1.plot(data2,'go-',label = 'data2')
ax1.legend(loc='best')

# 多scatter
data = np.random.randn(2,50)
ax2.scatter(data[0,:],data[1,:],s=40,c='g',marker='*',label = 'data1')
data = np.random.randn(2,50)
ax2.scatter(data[0,:],data[1,:],s=60,c='b',marker='o',label = 'data2')
ax2.legend(loc='best')
```



轴标签和标题 x轴 y轴的取值范围

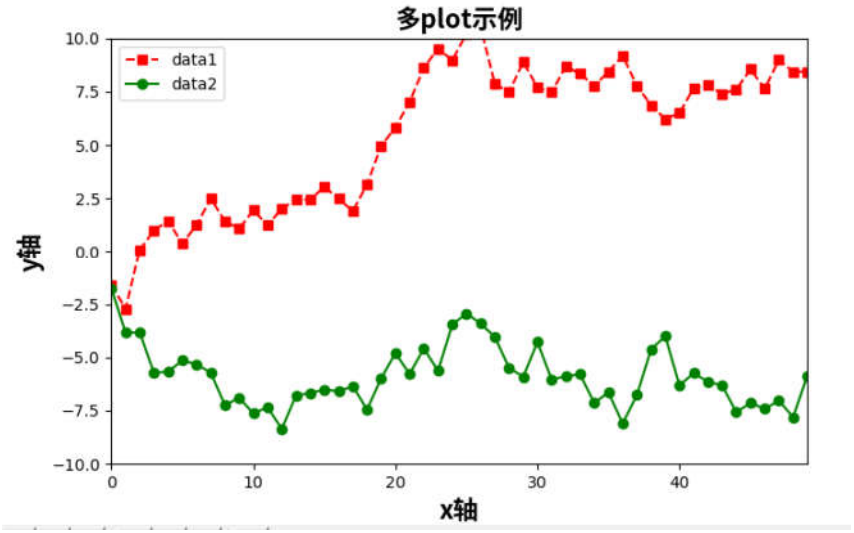
标题 **plt.title()** **ax.**set_title()

x轴/y轴 **plt.**xlabel() **plt.**ylabel()
ax.set_xlabel() **ax.set_ylabel()**

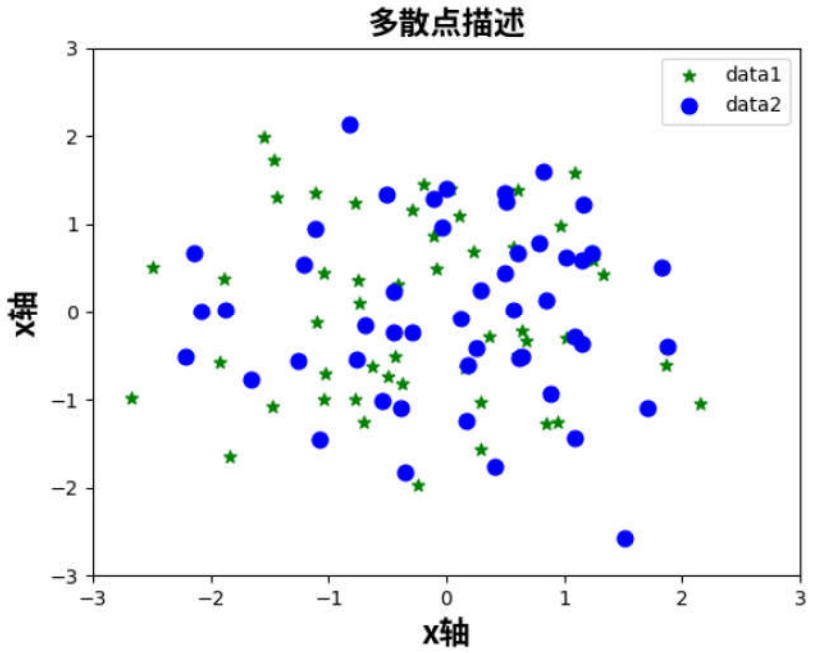
坐标范围 **plt.xlim([a,b])** **plt.ylim([a,b])**
ax.set_xlim([a,b]) **ax.set_ylim([a,b])**

中文显示问题

```
# 坐标轴及标题设置
# 字体
zhfont1 = matplotlib.font_manager.FontProperties(fname="NotoSansCJK-Bold.ttc", size=16)
fig = plt.figure(8)
ax1 = fig.add_subplot(1,1,1)

# 多plot
data1 = np.random.randn(50).cumsum()
data2 = np.random.randn(50).cumsum()
ax1.plot(data1,'rs--',label = 'data1')
ax1.plot(data2,'go-',label = 'data2')
ax1.legend(loc='best')

ax1.set_xlim([0,49])
ax1.set_ylim([-10,10])
ax1.set_xlabel('x轴',fontproperties=zhfont1)
ax1.set_ylabel('y轴',fontproperties=zhfont1)
ax1.set_title('多plot示例',fontproperties=zhfont1)
```



```
fig = plt.figure(9)
# 多scatter
data = np.random.randn(2,50)
plt.scatter(data[0,:],data[1,:],s=40,c='g',marker='*',label = 'data1')
data = np.random.randn(2,50)
plt.scatter(data[0,:],data[1,:],s=60,c='b',marker='o',label = 'data2')
plt.legend(loc='best')
plt.xlim([-3,3])
plt.ylim([-3,3])
plt.xlabel('x轴',fontproperties=zhfont1)
plt.ylabel('x轴',fontproperties=zhfont1)
plt.title("多散点描述",fontproperties=zhfont1)
```

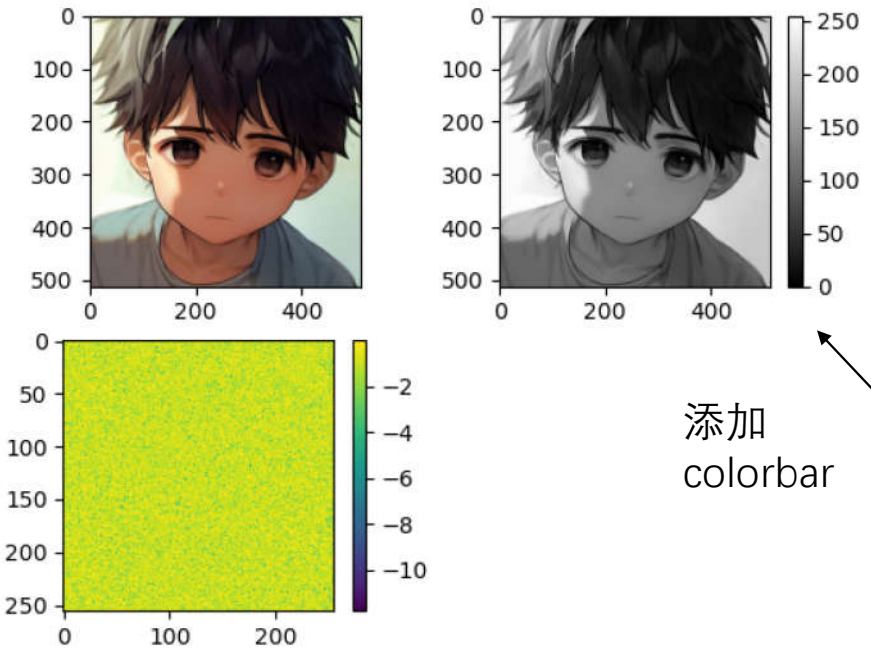


plt 显示图像

(M,N) 结合cmap 进行颜色显示

imshow(data, cmap)

(M,N,3) 直接显示图像



添加 colorbar

```
# 图像显示
fig = plt.figure('IMG')
ax1 = fig.add_subplot(2,2,1)
ax2 = fig.add_subplot(2,2,2)
ax3 = fig.add_subplot(2,2,3)

img = cv2.imread("0.jpg")
img1 = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 显示彩色图
ax1.imshow(img1)

# 显示灰度图
im_g = ax2.imshow(gray,cmap='gray',vmin=0,vmax =255)
plt.colorbar(im_g,ax =ax2 )

# 将一个随机矩阵 进行颜色显示
data = np.log(np.random.rand(256,256))
img_r = ax3.imshow(data)
plt.colorbar(img_r,ax = ax3)
```

图像的保存

```
# 最基本的使用
data = np.array([1,2,3,4,5,6,7])
# plt.figure(1)
# plt.plot(data)

#######默认###############
fig = plt.figure(1)
ax = fig.add_subplot(1, 1,1)
ax.plot(data)

plt.savefig("1.jpg")
```