

原文链接: [blog.ouyangsihai.cn](http://blog.ouyangsihai.cn) >> [matplotlib绘图入门详解](#)

matplotlib是受MATLAB的启发构建的。MATLAB是数据绘图领域广泛使用的语言和工具。MATLAB语言是面向过程的。利用函数的调用，MATLAB中可以轻松的利用一行命令来绘制直线，然后再用一系列的函数调整结果。

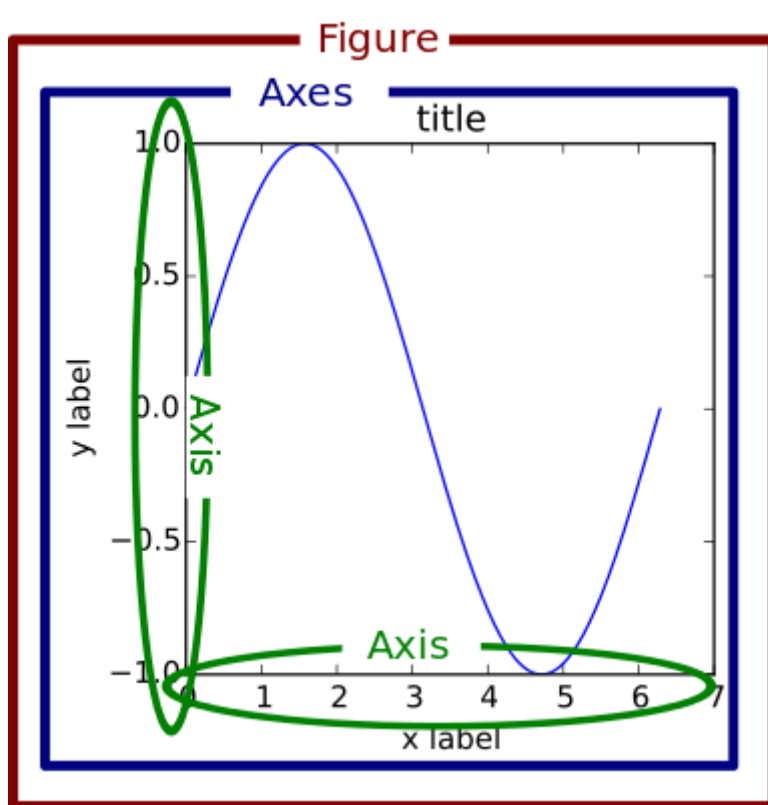
matplotlib有一套完全仿照MATLAB的函数形式的绘图接口，在matplotlib.pyplot模块中。这套函数接口方便MATLAB用户过度到matplotlib包

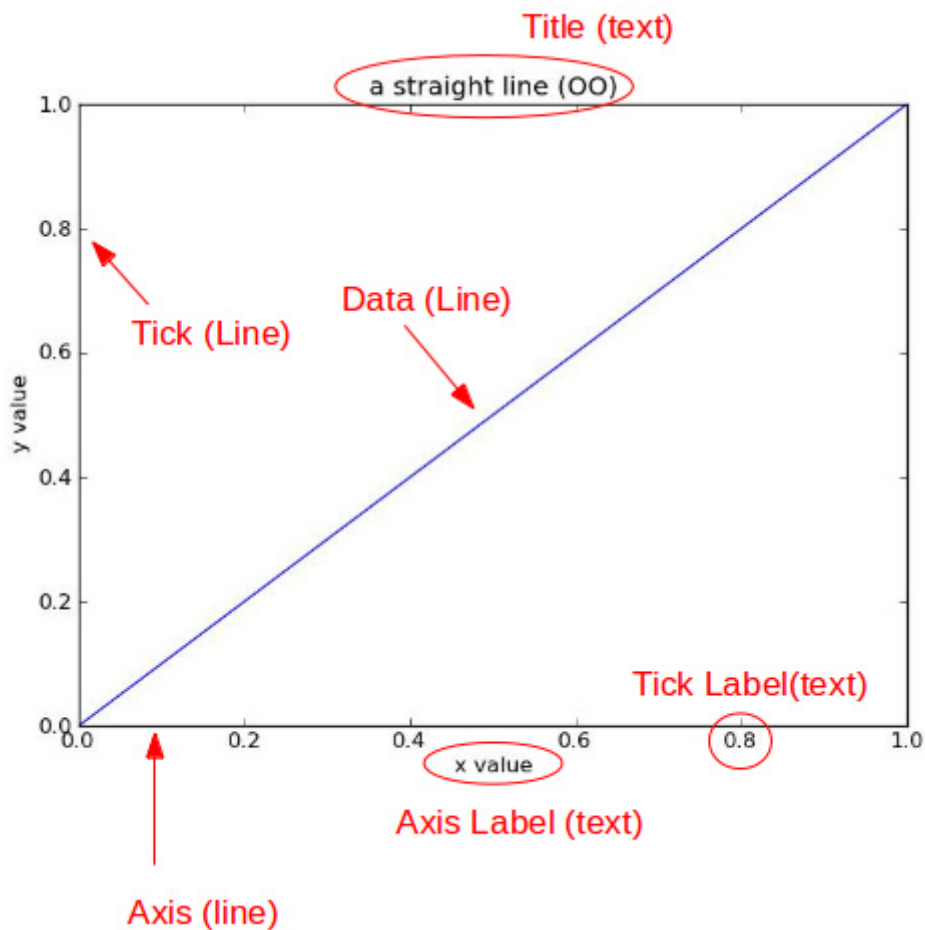
官网：<http://matplotlib.org/>

## 学习方式：从官网examples入门学习

- <http://matplotlib.org/examples/index.html>
- <http://matplotlib.org/gallery.html>

```
1 import matplotlib.pyplot as plt
```





在绘图结构中：

figure创建窗口，subplot创建子图。

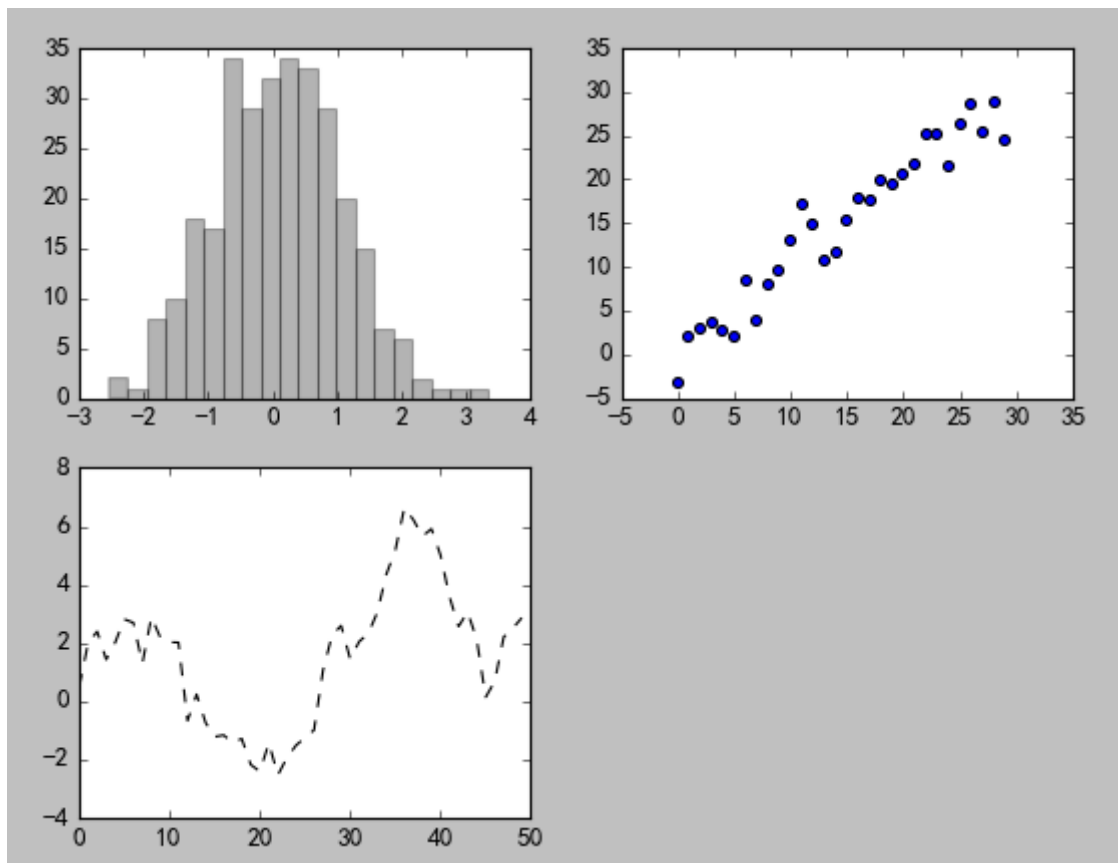
所有的绘画只能在子图上进行。

plt表示当前子图，若没有就创建一个子图。

所以，你会看到一些教程中使用plt进行设置，一些教程使用子图属性进行设置。他们往往存在对应功能函数。

Figure：面板(图)，matplotlib中的所有图像都是位于figure对象中，一个图像只能有一个figure对象。

Subplot：子图，figure对象下创建一个或多个subplot对象(即axes)用于绘制图像。



## 配置参数：

axex: 设置坐标轴边界和表面的颜色、坐标刻度值大小和网格的显示

figure: 控制dpi、边界颜色、图形大小、和子区( subplot)设置

font: 字体集 (font family) 、字体大小和样式设置

grid: 设置网格颜色和线性

legend: 设置图例和其中的文本的显示

line: 设置线条 (颜色、线型、宽度等) 和标记

patch: 是填充2D空间的图形对象，如多边形和圆。控制线宽、颜色和抗锯齿设置等。

savefig: 可以对保存的图形进行单独设置。例如，设置渲染的文件的背景为白色。

verbose: 设置matplotlib在执行期间信息输出，如silent、helpful、debug和debug-annoying。

xticks和yticks: 为x,y轴的主刻度和次刻度设置颜色、大小、方向，以及标签大小。

# 线条相关属性标记设置

| 线条风格linestyle或ls | 描述    |
|------------------|-------|
| '-'              | 实线    |
| '.'              | 虚线    |
| '-.'             | 破折线   |
| 'None' , ' ', '' | 什么都不画 |
| '-.'             | 点划线   |

# 线条标记

|    |                |          |
|----|----------------|----------|
| 1  | 标记maker        | 描述       |
| 2  |                |          |
| 3  | 'o'            | 圆圈       |
| 4  | '.'            | 点        |
| 5  | 'D'            | 菱形       |
| 6  | 's'            | 正方形      |
| 7  | 'h'            | 六边形1     |
| 8  | '*'            | 星号       |
| 9  | 'H'            | 六边形2     |
| 10 | 'd'            | 小菱形      |
| 11 | '_'            | 水平线      |
| 12 | 'v'            | 一角朝下的三角形 |
| 13 | '8'            | 八边形      |
| 14 | '<'            | 一角朝左的三角形 |
| 15 | 'p'            | 五边形      |
| 16 | '>'            | 一角朝右的三角形 |
| 17 | ','            | 像素       |
| 18 | '^'            | 一角朝上的三角形 |
| 19 | '+'            | 加号       |
| 20 | '\ '           | 竖线       |
| 21 | 'None', '', '' | 无        |
| 22 | 'x'            | X        |

# 颜色

|   |    |    |
|---|----|----|
| 1 | 别名 | 颜色 |
| 2 |    |    |

```
3 b 蓝色
4 g 绿色
5 r 红色
6 y 黄色
7 c 青色
8 k 黑色
9 m 洋红色
10 w 白色
```

如果这两种颜色不够用，还可以通过两种其他方式来定义颜色值：

- 1、使用HTML十六进制字符串 `color=' #123456'` 使用合法的HTML颜色名字（'red' , 'chartreuse' 等）。
- 2、也可以传入一个归一化到[0,1]的RGB元祖。 `color=(0.3,0.3,0.4)`

## 背景色

通过向如`matplotlib.pyplot.axes()`或者`matplotlib.pyplot.subplot()`这样的方法提供一个`axisbg`参数，可以指定坐标这的背景色。

```
subplot(111,axisbg=(0.1843,0.3098,0.3098))
```

## 以下示例需要引入的库包括

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from matplotlib.ticker import MultipleLocator
```

## 绘图操作步骤（以点图、线图为例）

```
1 #使用numpy产生数据
2 x=np.arange(-5,5,0.1)
3 y=x*3
4
5 #创建窗口、子图
6 #方法1：先创建窗口，再创建子图。（一定绘制）
7 fig = plt.figure(num=1, figsize=(15, 8),dpi=80) #开启一个窗口，同时设置大小，分辨率
8 ax1 = fig.add_subplot(2,1,1) #通过fig添加子图，参数：行数，列数，第几个。
9 ax2 = fig.add_subplot(2,1,2) #通过fig添加子图，参数：行数，列数，第几个。
10 print(fig,ax1,ax2)
11 #方法2：一次性创建窗口和多个子图。（空白不绘制）
12 fig,axarr = plt.subplots(4,1) #开一个新窗口，并添加4个子图，返回子图数组
```

```

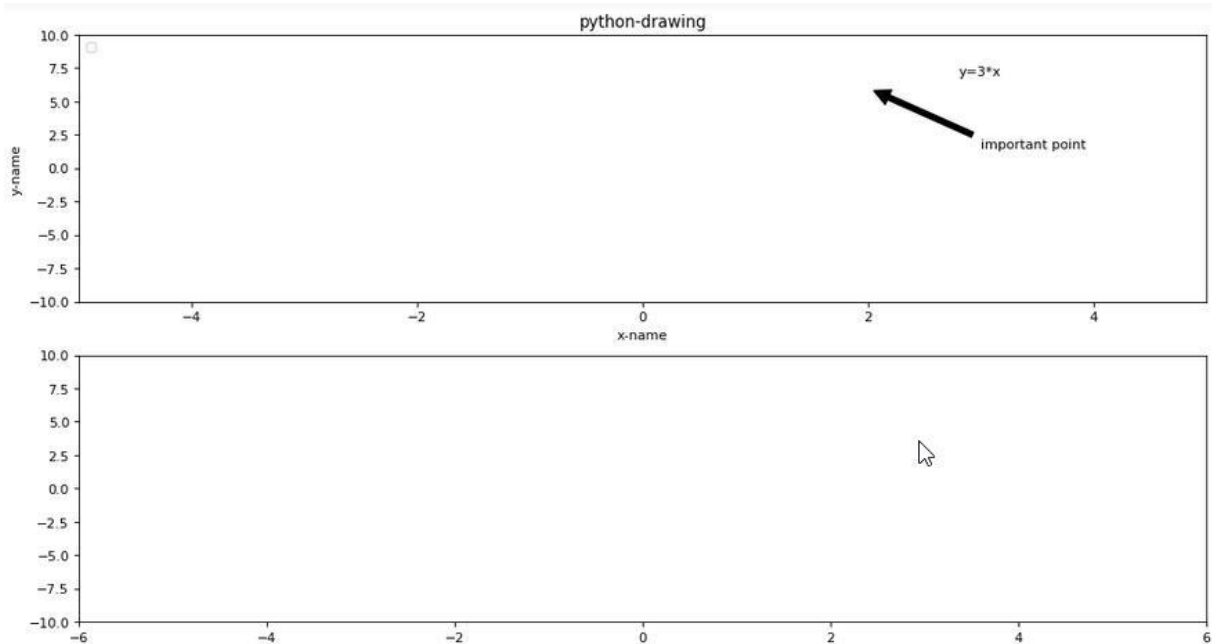
13 ax1 = axarr[0] #通过子图数组获取一个子图
14 print(fig,ax1)
15 #方法3: 一次性创建窗口和一个子图。(空白不绘制)
16 ax1 = plt.subplot(1,1,1,facecolor='white') #开一个新窗口, 创建1个子图。face
color设置背景颜色
17 print(ax1)
18 #获取对窗口的引用, 适用于上面三种方法
19 # fig = plt.gcf() #获得当前figure
20 # fig=ax1.figure #获得指定子图所属窗口
21
22 # fig.subplots_adjust(left=0) #设置窗口左内边距为0, 即左边留白为0。
23
24 #设置子图的基本元素
25 ax1.set_title('python-drawing') #设置图体, plt.title
26 ax1.set_xlabel('x-name') #设置x轴名称,plt.xlabel
27 ax1.set_ylabel('y-name') #设置y轴名称,plt.ylabel
28 plt.axis([-6,6,-10,10]) #设置纵横坐标轴范围, 这个在子图中被分解为下面两个函数
29 ax1.set_xlim(-5,5) #设置横轴范围, 会覆盖上面的横坐标,plt.xlim
30 ax1.set_ylim(-10,10) #设置纵轴范围, 会覆盖上面的纵坐标,plt.ylim
31
32 xmajorLocator = MultipleLocator(2) #定义横向主刻度标签的刻度差为2的倍数。就
是隔几个刻度才显示一个标签文本
33 ymajorLocator = MultipleLocator(3) #定义纵向主刻度标签的刻度差为3的倍数。就
是隔几个刻度才显示一个标签文本
34
35 ax1.xaxis.set_major_locator(xmajorLocator) #x轴 应用定义的横向主刻度格式。
如果不应用将采用默认刻度格式
36 ax1.yaxis.set_major_locator(ymajorLocator) #y轴 应用定义的纵向主刻度格式。
如果不应用将采用默认刻度格式
37
38 ax1.xaxis.grid(True, which='major') #x坐标轴的网格使用定义的主刻度格式
39 ax1.yaxis.grid(True, which='major') #x坐标轴的网格使用定义的主刻度格式
40
41 ax1.set_xticks([]) #去除坐标轴刻度
42 ax1.set_xticks((-5,-3,-1,1,3,5)) #设置坐标轴刻度
43 ax1.set_xticklabels(labels=['x1','x2','x3','x4','x5'],rotation=-30,fontsi
ze='small') #设置刻度的显示文本, rotation旋转角度, fontsize字体大小
44
45 plot1=ax1.plot(x,y,marker='o',color='g',label='legend1') #点图: marker图
标
46 plot2=ax1.plot(x,y,linestyle='--',alpha=0.5,color='r',label='legend2') #
线图: linestyle线性, alpha透明度, color颜色, label图例文本
47

```

```

48 ax1.legend(loc='upper left') #显示图例,plt.legend()
49 ax1.text(2.8, 7, r'y=3*x') #指定位置显示文字,plt.text()
50 ax1.annotate('important point', xy=(2, 6), xytext=(3, 1.5), #添加标注, 参
数: 注释文本、指向点、文字位置、箭头属性
51     arrowprops=dict(facecolor='black', shrink=0.05),
52 )
53 #显示网格。which参数的值为major(只绘制大刻度)、minor(只绘制小刻度)、both, 默
认值为major。axis为'x','y','both'
54 ax1.grid(b=True,which='major',axis='both',alpha= 0.5,color='skyblue',lin
estyle='--',linewidth=2)
55
56 axes1 = plt.axes([.2, .3, .1, .1], facecolor='y') #在当前窗口添加一个子图,
rect=[左, 下, 宽, 高], 是使用的绝对布局, 不和以存在窗口挤占空间
57 axes1.plot(x,y) #在子图上画图
58 plt.savefig('aa.jpg',dpi=400,bbox_inches='tight') #savefig保存图片, dpi分
辨率, bbox_inches子图周边白色空间的大小
59 plt.show() #打开窗口, 对于方法1创建在窗口一定绘制, 对于方法2方法3创建的窗口,
若坐标系全部空白, 则不绘制

```



plot时可以设置的属性包括如下:

- 1 属性 值类型
- 2 alpha 浮点值
- 3 animated [True / False]
- 4 antialiased or aa [True / False]
- 5 clip\_box matplotlib.transform.Bbox 实例
- 6 clip\_on [True / False]
- 7 clip\_path Path 实例, Transform, 以及Patch实例

```

8 color or c 任何 matplotlib 颜色
9 contains 命中测试函数
10 dash_capstyle ['butt' / 'round' / 'projecting']
11 dash_joinstyle ['miter' / 'round' / 'bevel']
12 dashes 以点为单位的连接/断开墨水序列
13 data (np.array xdata, np.array ydata)
14 figure matplotlib.figure.Figure 实例
15 label 任何字符串
16 linestyle or ls [ '-' / '--' / '-.' / ':' / 'steps' / ... ]
17 linewidth or lw 以点为单位的浮点值
18 lod [True / False]
19 marker [ '+' / ',' / '.' / '1' / '2' / '3' / '4' ]
20 markeredgcolor or mec 任何 matplotlib 颜色
21 markeredgewidth or mew 以点为单位的浮点值
22 markerfacecolor or mfc 任何 matplotlib 颜色
23 markersize or ms 浮点值
24 markevery [ None / 整数值 / (startind, stride) ]
25 picker 用于交互式线条选择
26 pickradius 线条的拾取选择半径
27 solid_capstyle ['butt' / 'round' / 'projecting']
28 solid_joinstyle ['miter' / 'round' / 'bevel']
29 transform matplotlib.transforms.Transform 实例
30 visible [True / False]
31 xdata np.array
32 ydata np.array
33 zorder 任何数值

```

## 一个窗口多个图

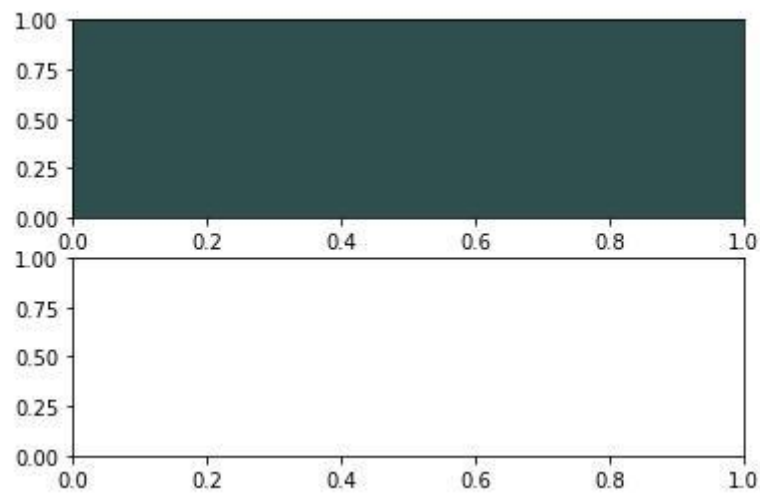
```

1 #一个窗口，多个图，多条数据
2 sub1=plt.subplot(211,facecolor=(0.1843,0.3098,0.3098)) #将窗口分成2行1列，
   在第1个作图，并设置背景色
3 sub2=plt.subplot(212) #将窗口分成2行1列，在第2个作图
4 sub1.plot(x,y) #绘制子图
5 sub2.plot(x,y) #绘制子图
6
7 axes1 = plt.axes([.2, .3, .1, .1], facecolor='y') #添加一个子坐标系，rect=
   [左，下，宽，高]
8 plt.plot(x,y) #绘制子坐标系，
9 axes2 = plt.axes([0.7, .2, .1, .1], facecolor='y') #添加一个子坐标系，rect=
   [左，下，宽，高]
10 plt.plot(x,y)

```



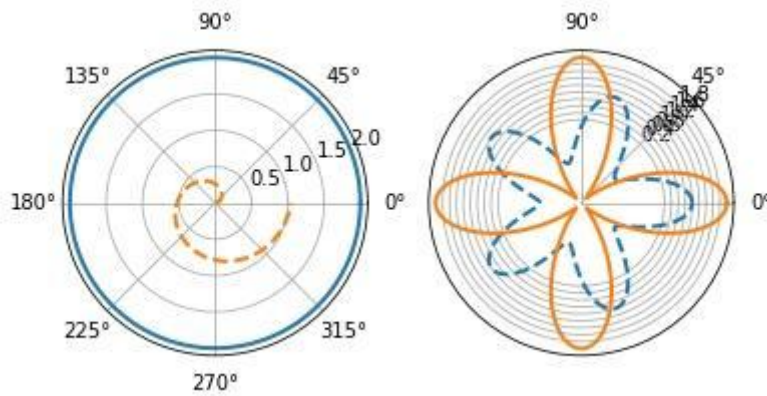
```
11 plt.show()
```



## 极坐标

属性设置同点图、线图中。

```
1 fig = plt.figure(2) #新开一个窗口
2 ax1 = fig.add_subplot(1,2,1,polar=True) #启动一个极坐标子图
3 theta=np.arange(0,2*np.pi,0.02) #角度数列值
4 ax1.plot(theta,2*np.ones_like(theta),lw=2) #画图，参数：角度，半径，lw线宽
5 ax1.plot(theta,theta/6,linestyle='--',lw=2) #画图，参数：角度，半径，linestyle样式，lw线宽
6
7 ax2 = fig.add_subplot(1,2,2,polar=True) #启动一个极坐标子图
8 ax2.plot(theta,np.cos(5*theta),linestyle='--',lw=2)
9 ax2.plot(theta,2*np.cos(4*theta),lw=2)
10
11 ax2.set_rgrids(np.arange(0.2,2,0.2),angle=45) #距离网格轴，轴线刻度和显示位置
12 ax2.set_thetagrids([0,45,90]) #角度网格轴，范围0-360度
13
14 plt.show()
```



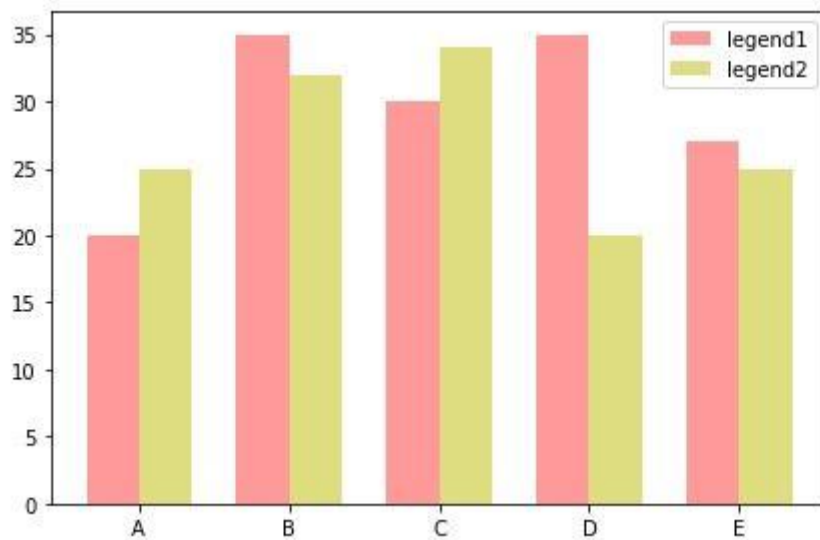
## 柱形图

属性设置同点图、线图中。

```

1 plt.figure(3)
2 x_index = np.arange(5) #柱的索引
3 x_data = ('A', 'B', 'C', 'D', 'E')
4 y1_data = (20, 35, 30, 35, 27)
5 y2_data = (25, 32, 34, 20, 25)
6 bar_width = 0.35 #定义一个数字代表每个独立柱的宽度
7
8 rects1 = plt.bar(x_index, y1_data, width=bar_width,alpha=0.4, color='b',label='legend1') #参数: 左偏移、高度、柱宽、透明度、颜色、图例
9 rects2 = plt.bar(x_index + bar_width, y2_data, width=bar_width,alpha=0.5,color='r',label='legend2') #参数: 左偏移、高度、柱宽、透明度、颜色、图例
10 #关于左偏移, 不用关心每根柱的中心不中心, 因为只要把刻度线设置在柱的中间就可以了
11 plt.xticks(x_index + bar_width/2, x_data) #x轴刻度线
12 plt.legend() #显示图例
13 plt.tight_layout() #自动控制图像外部边缘, 此方法不能够很好的控制图像间的间隔
14 plt.show()

```

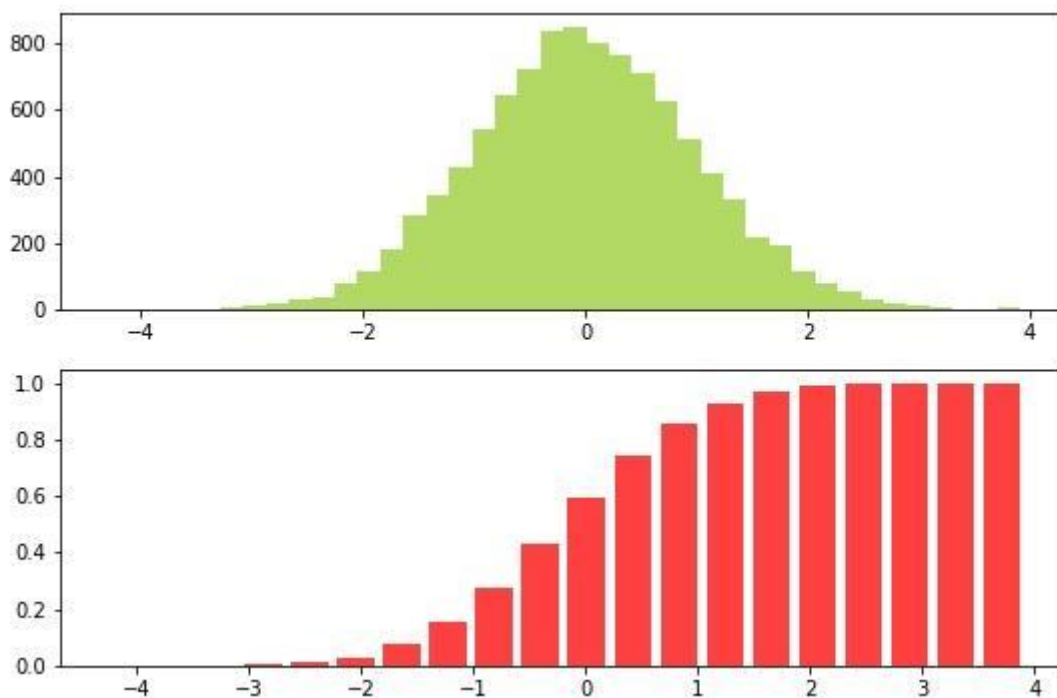


## 直方图

```

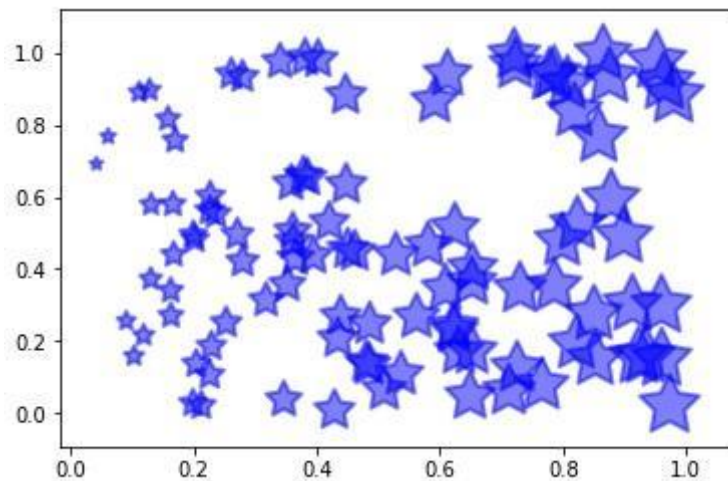
1 fig, (ax0, ax1) = plt.subplots(nrows=2, figsize=(9, 6)) #在窗口上添加2个子图
2 sigma = 1 #标准差
3 mean = 0 #均值
4 x=mean+sigma*np.random.randn(10000) #正态分布随机数
5 ax0.hist(x, bins=40, normed=False, histtype='bar', facecolor='yellowgreen', alpha=0.75) #normed是否归一化, histtype直方图类型, facecolor颜色, alpha透明度
6 ax1.hist(x, bins=20, normed=1, histtype='bar', facecolor='pink', alpha=0.75, cumulative=True, rwidth=0.8) #bins柱子的个数, cumulative是否计算累加分布, rwidth柱子宽度
7 plt.show() #所有窗口运行

```



## 散点图

```
1 fig = plt.figure(4) #添加一个窗口
2 ax =fig.add_subplot(1,1,1) #在窗口上添加一个子图
3 x=np.random.random(100) #产生随机数组
4 y=np.random.random(100) #产生随机数组
5 ax.scatter(x,y,s=x*1000,c='y',marker=(5,1),alpha=0.5,lw=2,facecolors='none') #x横坐标, y纵坐标, s图像大小, c颜色, marker图片, lw图像边框宽度
6 plt.show() #所有窗口运行
```



## 三维图

```
1 fig = plt.figure(5)
2 ax=fig.add_subplot(1,1,1,projection='3d') #绘制三维图
3
4 x,y=np.mgrid[-2:2:20j,-2:2:20j] #获取x轴数据, y轴数据
5 z=x*np.exp(-x**2-y**2) #获取z轴数据
6
7 ax.plot_surface(x,y,z,rstride=2,cstride=1,cmap=plt.cm.coolwarm,alpha=0.8)
#绘制三维图表面
8 ax.set_xlabel('x-name') #x轴名称
9 ax.set_ylabel('y-name') #y轴名称
10 ax.set_zlabel('z-name') #z轴名称
11
12 plt.show()
```

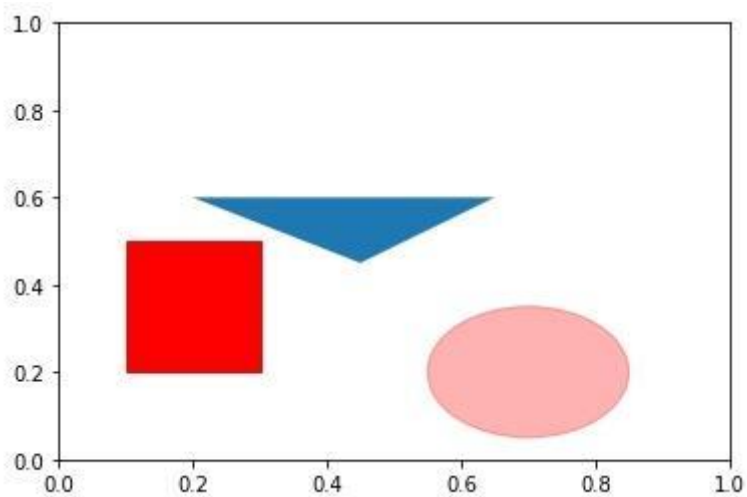
## 画矩形、多边形、圆形和椭圆

```
1 fig = plt.figure(6) #创建一个窗口
2 ax=fig.add_subplot(1,1,1) #添加一个子图
```

```

3 rect1 = plt.Rectangle((0.1,0.2),0.2,0.3,color='r') #创建一个矩形，参数：
(x,y),width,height
4 circ1 = plt.Circle((0.7,0.2),0.15,color='r',alpha=0.3) #创建一个椭圆，参
数：中心点，半径，默认这个圆形会跟随窗口大小进行长宽压缩
5 pgon1 = plt.Polygon([[0.45,0.45],[0.65,0.6],[0.2,0.6]]) #创建一个多边形，参
数：每个顶点坐标
6
7 ax.add_patch(rect1) #将形状添加到子图上
8 ax.add_patch(circ1) #将形状添加到子图上
9 ax.add_patch(pgon1) #将形状添加到子图上
10
11 fig.canvas.draw() #子图绘制
12 plt.show()

```



原文：

<https://blog.csdn.net/luanpeng825485697/article/details/78508819>

文章有不当之处，欢迎指正，如果喜欢微信阅读，你也可以关注我的微信公众号：[cplus人工智能算法后端技术](#)，获取优质学习资源。