**import matplotlib.pyplot as plt**

**import numpy as np**

plt.rcParams['font.sans-serif'] = ['Microsoft YaHei'] #解决中文乱码

基础

**保存图像**

fig=plt.figure()

…绘图命令…

fig.savefig("figure.png", bbox\_inches='tight')

**读取图像**

x=plt.imread('image.jpeg')

plt.imshow(x)

**显示灰度图**

plt.imshow(x, cmap=plt.cm.gray)

**利用numpy生成数据**

**range(a, b, step)**的步长只能是整数，常用于列表分片，生成列表

**np.arange(a, b, step)**：左闭右开，步长可以是小数，生成一维矩阵

**np.linspace(a, b, n)**: 采集[a, b)中均匀分布的n个点，可加endpoint=True使得左闭右闭，生成一维矩阵

**np.pi** #引用圆周率

**字体设定**

from matplotlib.font\_manager import FontProperties as fp

my\_font=fp(fname=r"C:\windows\fonts\simhei.ttf") #字体路径

需绘制中文字时，加上参数FontProperties=my\_font，还可以设置大小fontsize=18

**指定图的大小和分辨率：**

**fig,axes=plt.subplots(figsize=(m,n),dpi=num)** #figsize为长X高，num为像素

#具体像素为100mx100n, 每英寸num点

#plt.subplots()返回两个值，一个是图，一个是坐标轴

#后续画图需使用axes

画布设置

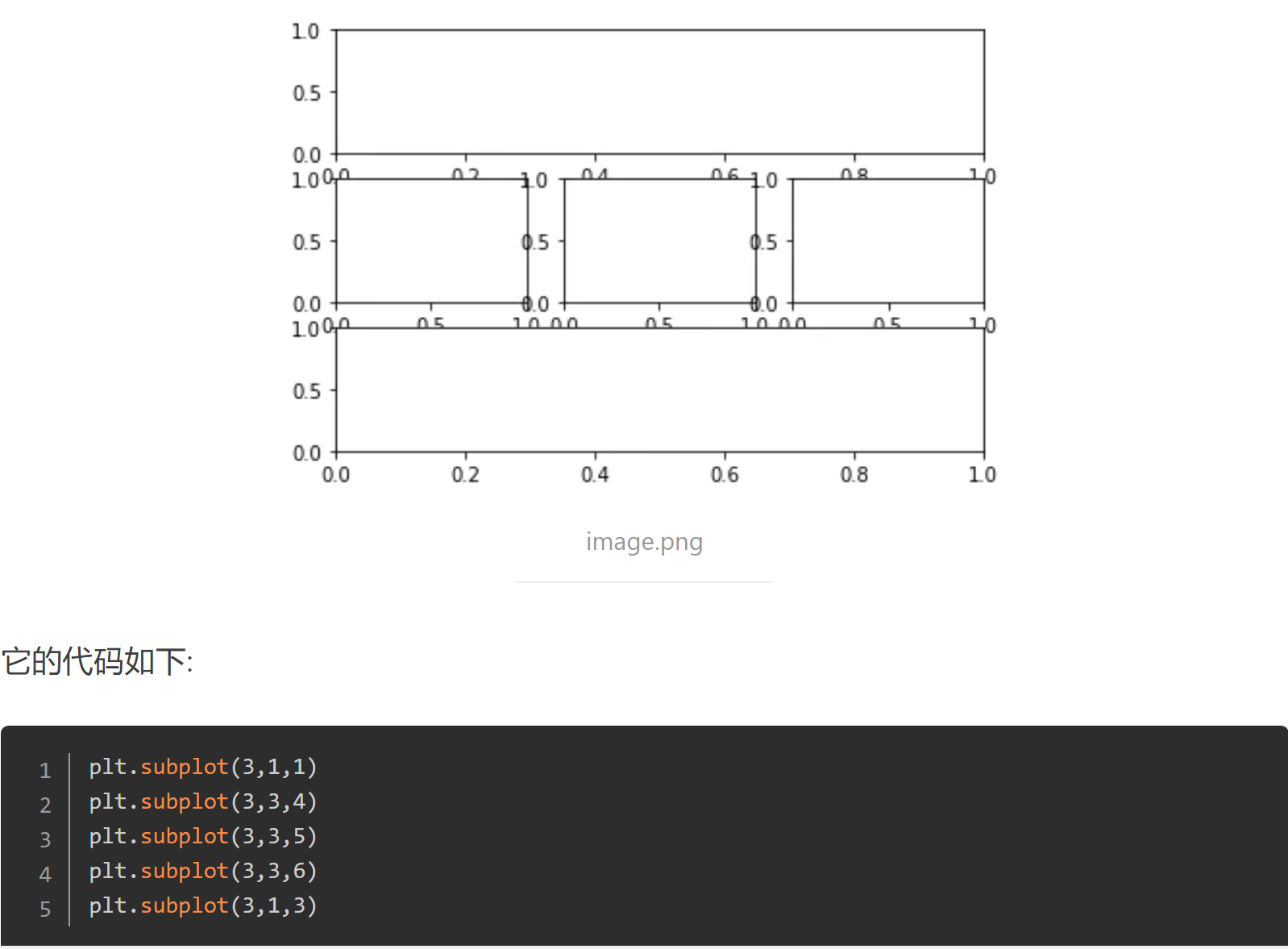
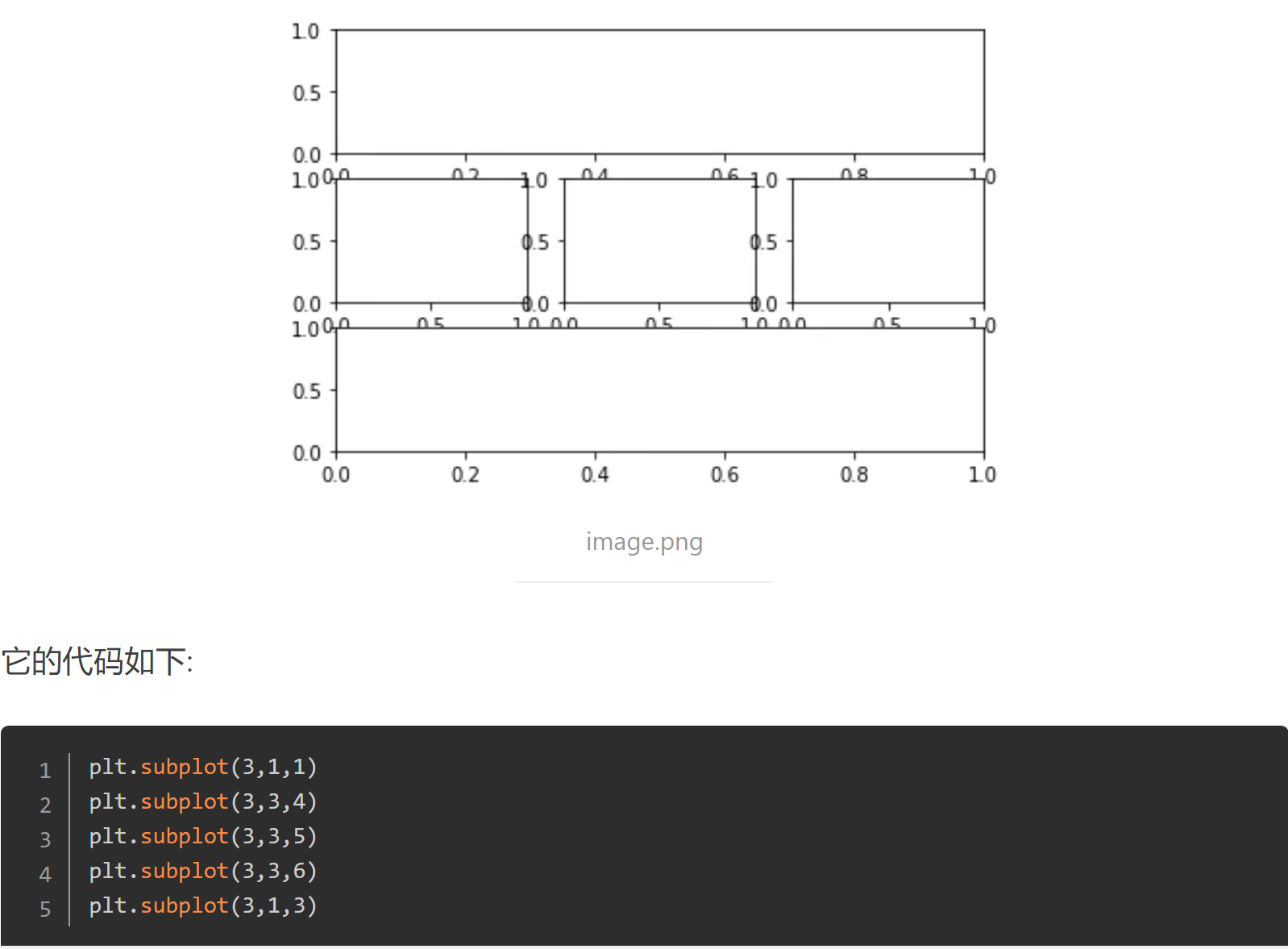
**画布图像设定**

[**https://www.jianshu.com/p/ec2dcd35d826**](https://www.jianshu.com/p/ec2dcd35d826)

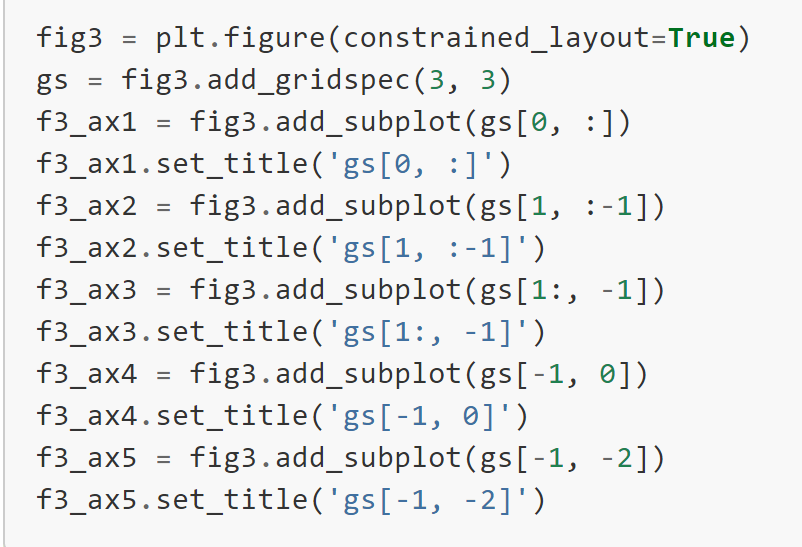
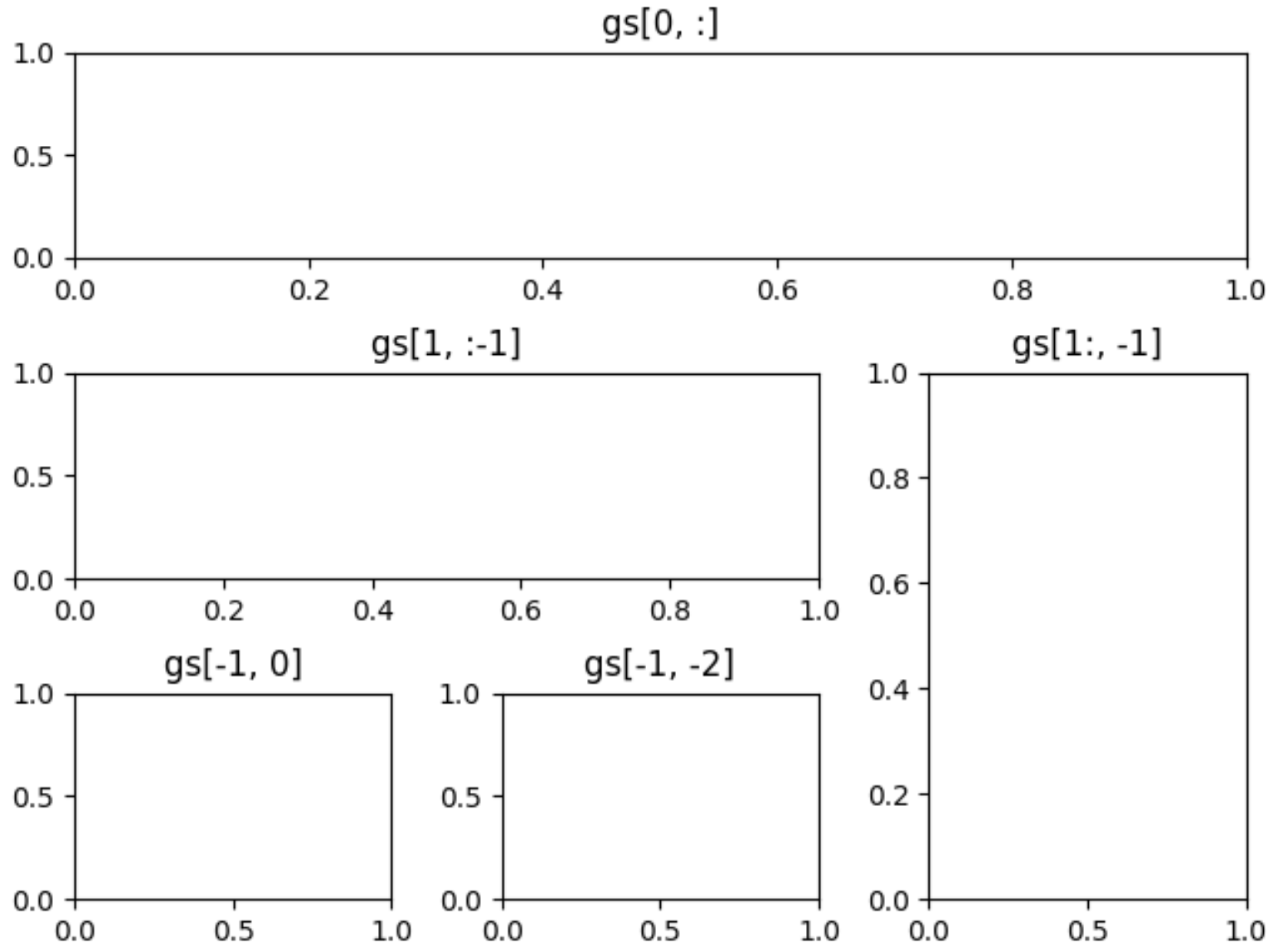
**fig=plt.figure()** #生成一个图像类

**fig\_n=plt.subplot(nrows=总行数, ncols=总列数, index=当前图序号)**

#用于在一行或一列上绘制多个子图,可以各个子图赋值给一个代号，nmax=总行数\*总列数



**plt.figure().add\_gridspec()函数进行布局定制：**



**三种绘制子图的方式**

**fig,axes=plt.subplots(nrows=总行数,ncols=总列数,figsize=画布尺寸元组,sharex=False)**

#fig是画布本身，axes即panel，是子图构成的二维集合，通过axes[a][b]调用每个子图并用类方法画图，如:

axes[i][j].plot(x,y)；

axes[i][j].set\_title(‘Name’);

axes[i][j].set\_xlabel(‘Xlab’, fontsize)

axes[i][j].set\_ylabel(‘ylab’ , fontsize)

axes[i][j].set\_zlabel(‘zlab’ , fontsize) #用于3D图

axes[i][j].set\_xticklabels(label\_list)

axes[i][j].set\_yticklabels(label\_list)

#如果是seaborn命令，则用sns.绘图命令(ax=axes[i][j],…)

也可以使用axes = axes.flatten()，然后用axes[i]来依次调用从左到右、从上至下每一幅图

for i, ax in enumerate(axes.flat): ax.operation()

**②**

**fig = plt.figure(figsize=(16,8))**

**ax = fig.add\_subplot(1,2,1,projection='3d')**

ax.plot(x,y)

ax.set\_xlabel(‘Xlab’, fontsize)

**③**

**plt.figure(figsize)**

**plt.subplot(nrows, ncols, index)**

**plt.plot()**

**设置多图的布局**

**plt.tight\_layout()**

**获取色板**

**cmap=plt.cm.get\_cmap('Blues')**

图例、元素和标注

**添加画布大标题**：**fig.suptitle(‘Name’,fontsize=20)**

**添加次级标题**：**fig.title(‘Name’,fontsize=20)**

**自定义画布：**

**figure=plt.figure(num='画布名', figsize=(长, 高), dpi=像素点数, facecolor='画布颜色代号',edgecolor='边框颜色代号')**

**线类型： 颜色：**

‘solid’ 实线 ‘-.’ 虚点线 ‘b’ 蓝色 ’r’ 红色 ‘y’ 黄色

‘dashed‘ 虚线 ‘:’ 点线 ‘g’ 绿色 ‘c’ 青色 ‘b’ 黑色

**点类型：**



**设置多曲线图的图例**

首先对各曲线:

**plt.plot(x,y, color=’颜色代号’, linewidth=线宽, linestyle=’线型’,label=’名称’,marker=‘点类型‘,markersize=点大小)**

然后：**legend(loc=’图例位置’)** #图例位置：upper left, upper right等

最后：**plt.show()**

**plt.grid(True)** #画网格线

**plt.xlim(a,b)** #设定X轴坐标范围

**plt.ylim(a,b)** #设定Y轴坐标范围

**plt.xlabel(‘X轴名’, fontsize)** #设定X轴坐标名

**plt.ylabel(‘Y轴名’, fontsize)** #设定Y轴坐标名

**plt.axis(‘off’)** #隐藏坐标轴

**plt.title(‘标题’, fontsize)** #设定标题

**plt.xticks(坐标数列,labels=坐标标签,rotation=字体旋转角度)** #自定义X轴坐标值

**plt.yticks(坐标数列,,labels=坐标标签,rotation=字体旋转角度)** #自定义Y轴坐标值

以上函数可以通用**plt.set()**通过传参的方式来设定

**plt.text(x,y,text)** #指定位置添加文本

**plt.legend(loc=位置,labels=标签名,fontsize,)**

**plt.show()** #一句plt.show()出一张图，多个plot如果用同一句plt.show(),会画在一张图上

**plt.axhline(y)** #添加水平线

**plt.axvline(x)** #添加竖直线

**plt.vlines(x, ymin, ymax)** #添加竖直线段

**plt.hlines(y,xmin,xmax)** #添加水平线段

**plt.fill\_between(x,y\_up,y\_bottom,facecolor=颜色,alpha=透明度) #**区域颜色填充

**plt.** **add\_patch(matplotlib.patches.Rectangle(左下角坐标元组,宽,高))** #添加方框

#https://stackoverflow.com/questions/37435369/matplotlib-how-to-draw-a-rectangle-on-image

**plt.figtext(x, y, s=text) #**添加文字，x,y∈[0,1]

**plt.legend().remove()** #删除图例

2D绘图

**绘制曲线图**

**plt.plot(x,y, color=’颜色代号’, linewidth=线宽, linestyle=’线型’,label=’名称’,marker=‘点类型‘,markersize=点大小)**

**散点图**

**plt.scatter(x,y,marker=点型,color=颜色,markersize=点大小,edgecolors=’颜色’)**

**气泡图**

**plt.scatter(x,y,s=面积,marker=点型,color=颜色,markersize=点大小,edgecolors=’颜色’)**

**竖直条形图**

heights,binedges,bars=plt.bar(x, height, alpha=.5,color=’颜色代号’, width=条块宽度, align=’center/edge’)

#height为柱的高度（Y值矩阵），align参数设定条块在坐标轴上的分布（中心或边缘）

#返回条块高度、区间端点值、条块指针

**水平条形图**

plt.barh(…)

**堆叠条形图**

plt.bar(x,y1,color=color1,label=label1)

plt.bar(x,y2,bottom=y1,color=color2,label=label2)

plt.bar(x,y3,bottom=y2+y3,color=color3,label=label3)

#y1,y2,y3需为ndarray

**绘制直方图**

**n,bins,patches=plt.hist(x,bins=分堆数,range=坐标轴范围,density=False,bottom=纵轴起线,align=’center/edge’,orientation=’horizontal’/’vertical’,log=False,color=条块颜色 , facecolor='直方图颜色', edgecolor='条块边色',alpha=透明度，histtype='bar/barstacked', rwidth=条块间距(0-1),cumulative=False)**

#edgecolor的值为一个三元数组[a,b,c]，分别代表条块边色的三原色比例

返回值n为直方图向量；bins为各个区间范围；patches为每个bin里包含的数据，list类型

**画饼图：**饼图是根据每个数据的百分比来画图的

**plt.pie(percentages, explode=[每个分块的拖出值],colors=块色, labels=标签,labeldistance=标签距中心距离(0-1),autopct=数据标注格式,pctdistance=数据标注距中心的距离(0-1),startangle=旋转角度,radius=半径,shadow=饼图阴影,) #explode中0的值表示不拖出分块**

#数据标注格式可以是浮点数控制符，如%1.1f%%表示宽度1，保留1位小数并加上百分号。%表示去转义

#一种加数值标签的方法

labels=[level+str(num) for level,num in zip(levels,percentages)]

**箱型图**

**plt.boxplot(data,notch=槽型或箱型,vert=竖直与否,showfliers=True,positions=各箱位置)**

**阶层聚类树**





3D绘图

**绘制三维图：**

**from matplotlib import cm**

**import matplotlib.pylab as plt**

**from mpl\_toolkits.mplot3d import Axes3D**

**fig = plt.figure()**

**ax = Axes3D(fig)**

用ax画图

ax.plot\_trisurf(X,Y,Z)#以小三角形构成曲面单元，使用x,y,z是等长的1D array，x,y,z对应的元素，组成维空间的一个点。

ax.plot\_surface(X,Y,Z)#以菱形构成曲面单元, 使用的是np.meshgrid产生的数据，x,y是两个二维坐标矩阵，Z是二维矩阵代表XY形成的每个网格点上的高

**绘制三维散点**

**from mpl\_toolkits.mplot3d import Axes3D**

**fig = plt.figure()**

**ax = fig.add\_subplot(projection='3d')**

**ax.scatter(x,y,z)**

**绘制等高线：**

**CS=plt.contour(X,Y,Z,圈数)**

**plt.clabel(CS，inline=1，fontsize=字体大小)** #inline=1时，将图上数字下的线条截断，避免数字和线条重叠; inline=0则不截断。CS只是图的代名

**plt.show()**

**绘制三维等高线：**

**from matplotlib import cm**

**import matplotlib.pylab as plt**

**from mpl\_toolkits.mplot3d import Axes3D**

**fig = plt.figure()**

**ax = Axes3D(fig)**

**X,Y=np.meshgrid(Xi,Yi)**

**ax2.contour(X,Y,Z,colors='颜色代号')**

**plt.show()**