



Matplotlib数据可视化基础

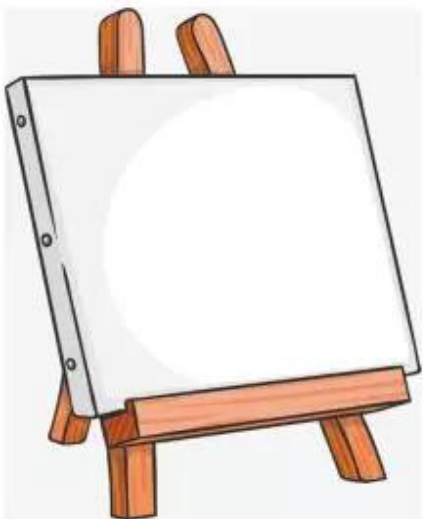
2020/5/19

目录

1	了解绘图基础语法与常用参数
2	折线图与散点图
3	饼图与直方图
4	小结

人类
绘画

画板



调色板



画画



Matplotlib
绘画

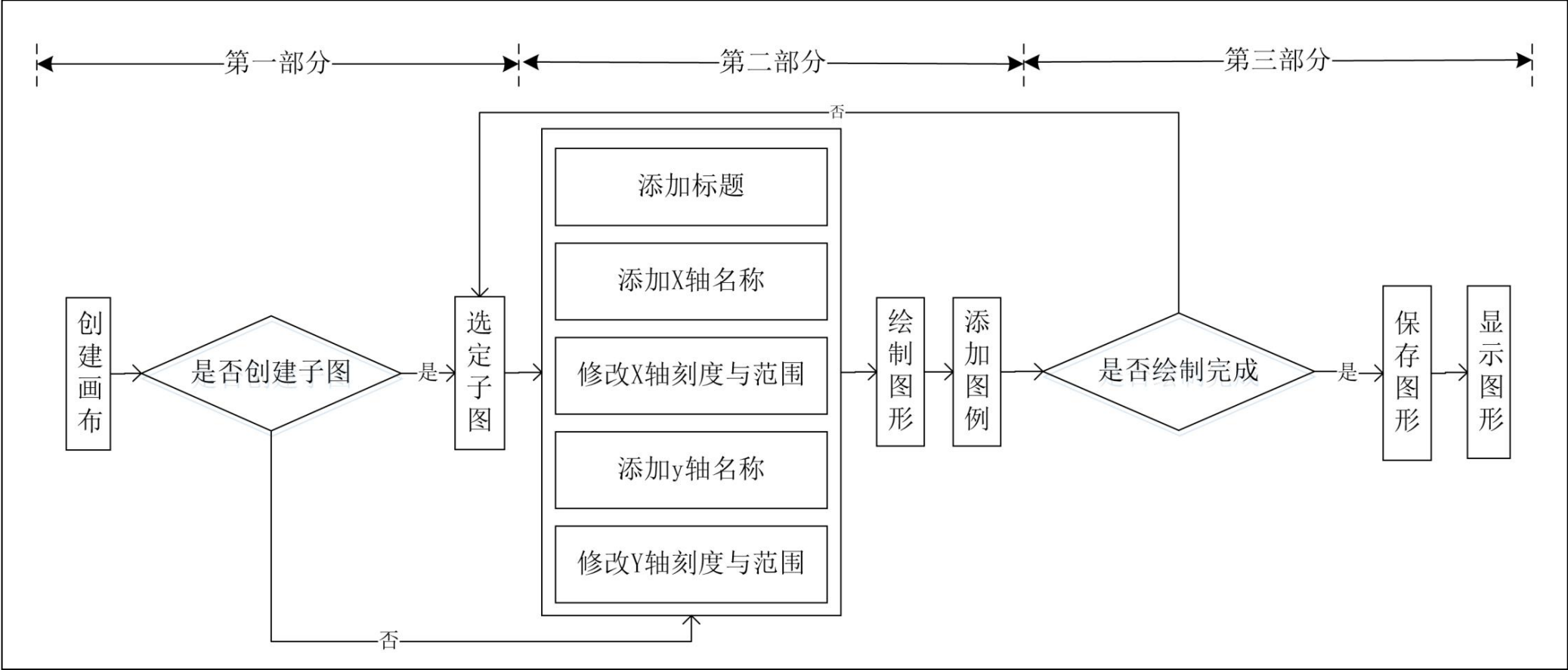
FigureCanvas

Renderer

Artist

掌握pyplot基础语法

基本绘图流程



掌握pyplot基础语法

1.创建画布与创建子图

第一部分主要作用是构建出一张空白的画布，并可以选择是否将整个画布划分为多个部分，方便在同一幅图上绘制多个图形的情况。最简单的绘图可以省略第一部分，而后直接在默认的画布上进行图形绘制。

命令	作用
<code>import matplotlib.pyplot as plt</code>	导入绘图函数库
<code>plt.show()</code>	脚本中绘图显示，放到代码最后
<code>%matplotlib</code> <code>#IPython交互绘图</code>	任何一个plt会打开图形窗口，增加命令 图形更新
<code>plt.figure</code>	创建一个空白画布，可以指定画布大小，像素。

掌握pyplot基础语法

2.添加画布内容

第二部分是绘图的主体部分。其中添加标题，坐标轴名称，绘制图形等步骤是并列的，没有先后顺序，可以先绘制图形，也可以先添加各类标签。但是添加图例一定要在绘制图形之后。

函数名称	函数作用
plt.title	在当前图形中添加标题，可以指定标题的名称、位置、颜色、字体大小等参数。
plt.xlabel	在当前图形中添加x轴名称，可以指定位置、颜色、字体大小等参数。
plt.ylabel	在当前图形中添加y轴名称，可以指定位置、颜色、字体大小等参数。
plt.xlim	指定当前图形x轴的范围，只能确定一个数值区间，而无法使用字符串标识。
plt.ylim	指定当前图形y轴的范围，只能确定一个数值区间，而无法使用字符串标识。
plt.xticks	指定x轴刻度的数目与取值。
plt.yticks	指定y轴刻度的数目与取值。
plt.legend	指定当前图形的图例，可以指定图例的大小、位置、标签。

掌握pyplot基础语法

3.存与展示图形

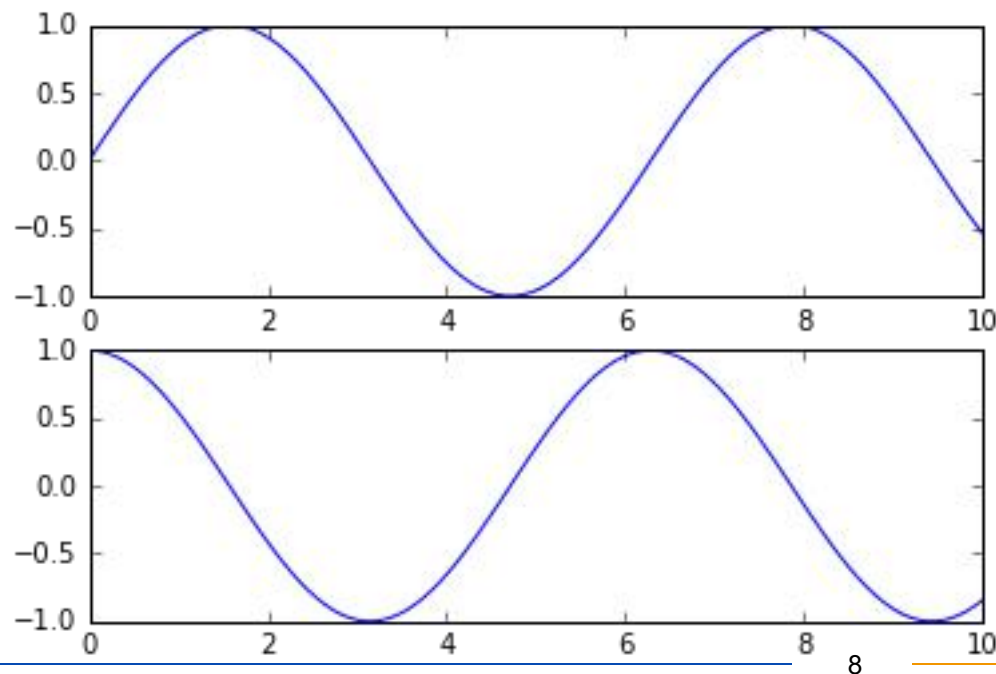
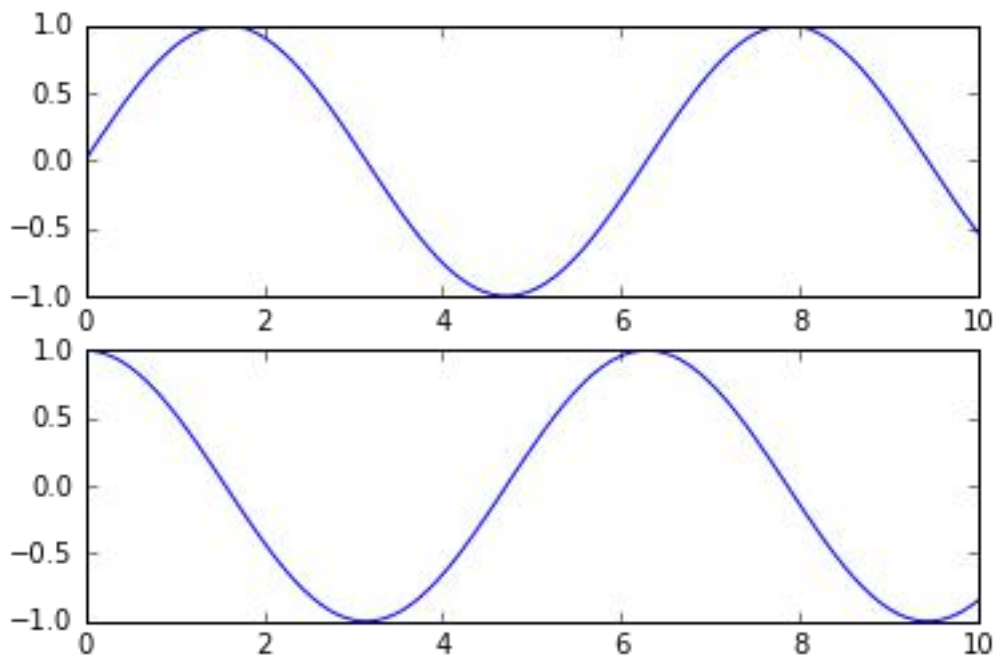
第三部分主要用于保存和显示图形。

函数名称	函数作用
plt.savefig	保存绘制的图片，可以指定图片的分辨率、边缘的颜色等参数。
plt.show	在本机显示图形。

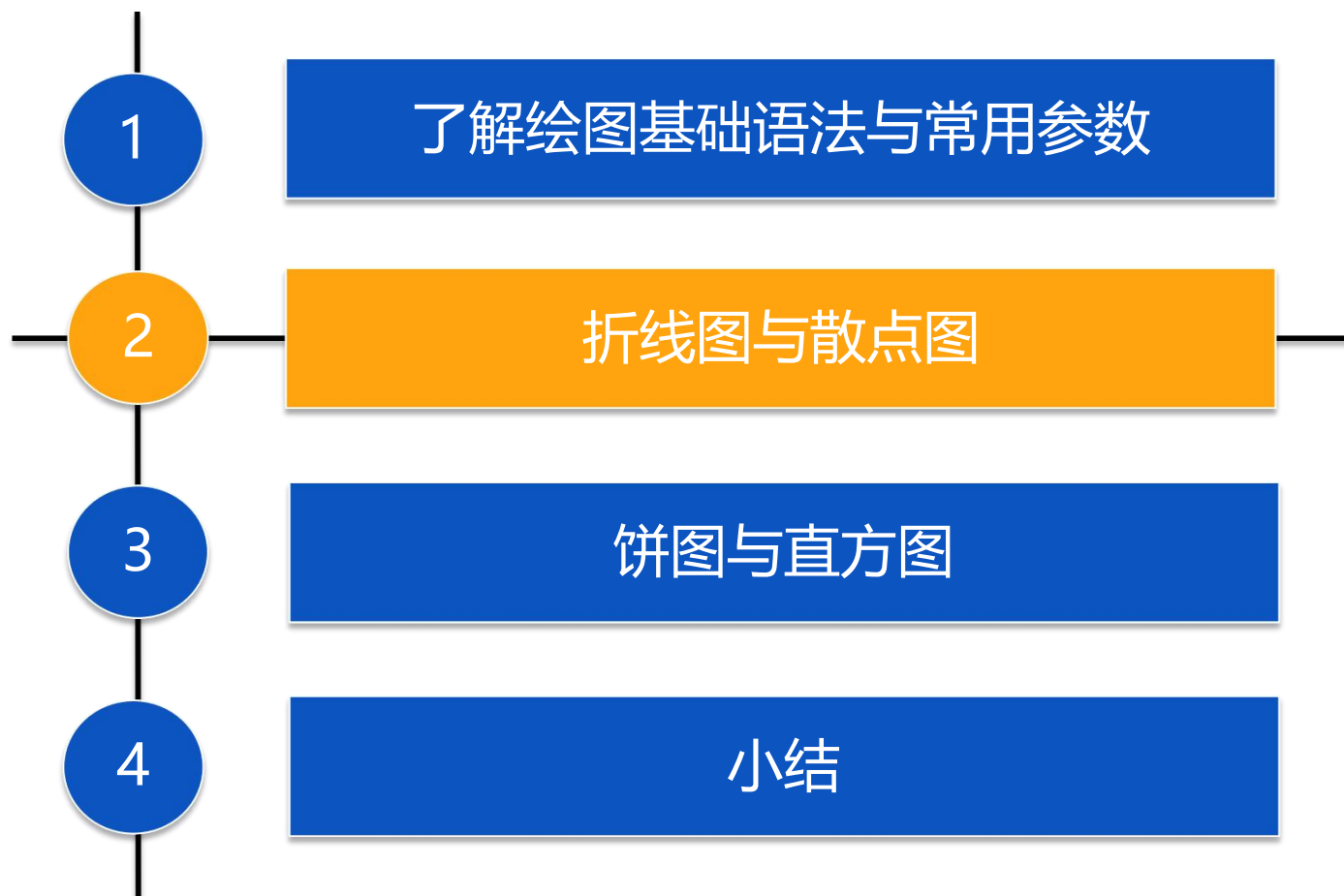
两种画图接口

```
x=np.linspace(0,10,100)
plt.figure() # create a plot figure 传统方法创建子图
# create the first of two panels and set current axis
plt.subplot(2, 1, 1) # (rows, columns, panel number)
plt.plot(x, np.sin(x))
# create the second panel and set current axis
plt.subplot(2, 1, 2)
plt.plot(x, np.cos(x))
```

```
# 面向对象方法创建子图 先创建图形网格
# ax子图数组
fig, ax = plt.subplots(2) # subplots(2,2)
# Call plot() method on the appropriate object
ax[0].plot(x, np.sin(x)) #ax[0,0]
ax[1].plot(x, np.cos(x))
```



目录



绘制折线图

plot函数

```
plt.plot(*args, **kwargs)
```

plot函数在官方文档的语法中只要求填入不定长参数，实际可以填入的主要参数主要如下。

参数名称	说明
x, y	接收array。表示x轴和y轴对应的数据。无默认。
color	接收特定string。指定线条的颜色。默认为None。
linestyle	接收特定string。指定线条类型。默认为 “-” 。
marker	接收特定string。表示绘制的点的类型。默认为None。
alpha	接收0-1的小数。表示点的透明度。默认为None。
linewidth	线条宽度 取0-10之间的数值
markersize	点的大小

绘制折线图

plot函数

color参数的8种常用颜色的缩写。

颜色缩写	代表的颜色	颜色缩写	代表的颜色
b	蓝色	m	品红
g	绿色	y	黄色
r	红色	k	黑色
c	青色	w	白色

linestyle取值	意义	linestyle取值	意义
-	实线	-.	点线
--	长虚线	:	短虚线

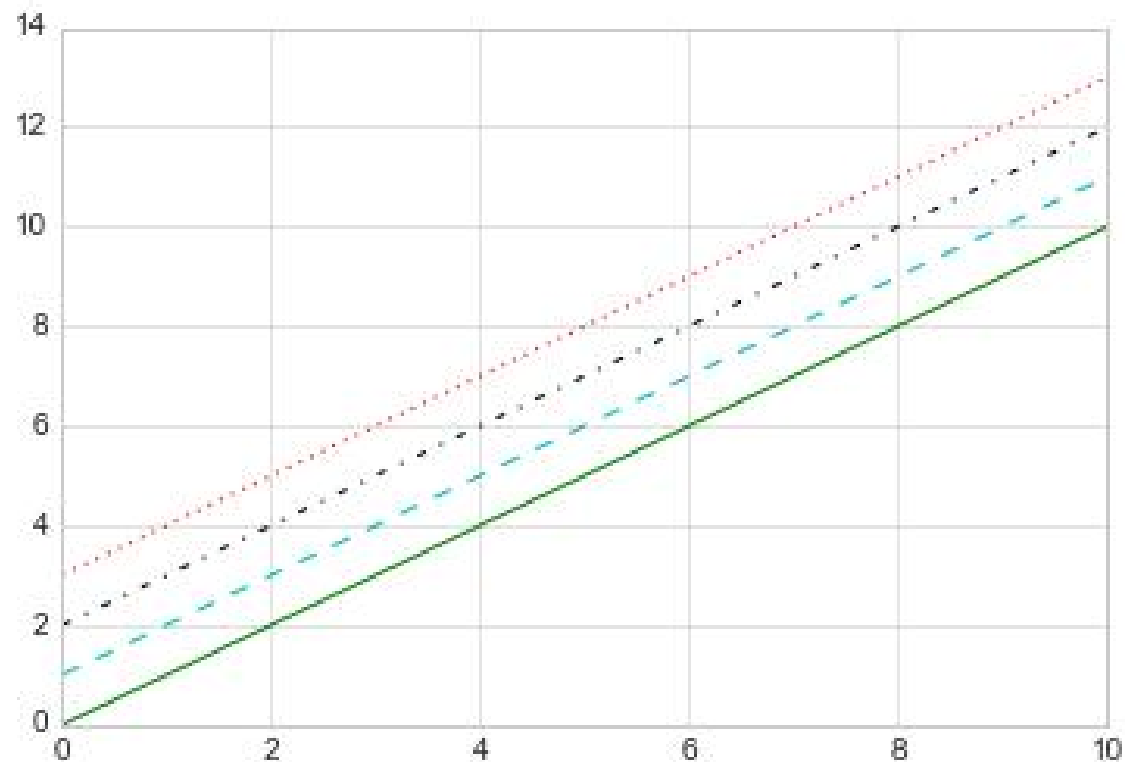
线条标记解释

marker取值	意义	marker取值	意义
'o'	圆圈	'.'	点
'D'	菱形	's'	正方形
'h'	六边形1	'*'	星号
'H'	六边形2	'd'	小菱形
'-'	水平线	'v'	一角朝下的三角形
'8'	八边形	'<'	一角朝左的三角形
'p'	五边形	'>'	一角朝右的三角形
' , '	像素	'^'	一角朝上的三角形
'+'	加号	'\'	竖线
'None'	无	'x'	X

任务实现

线图绘制实例

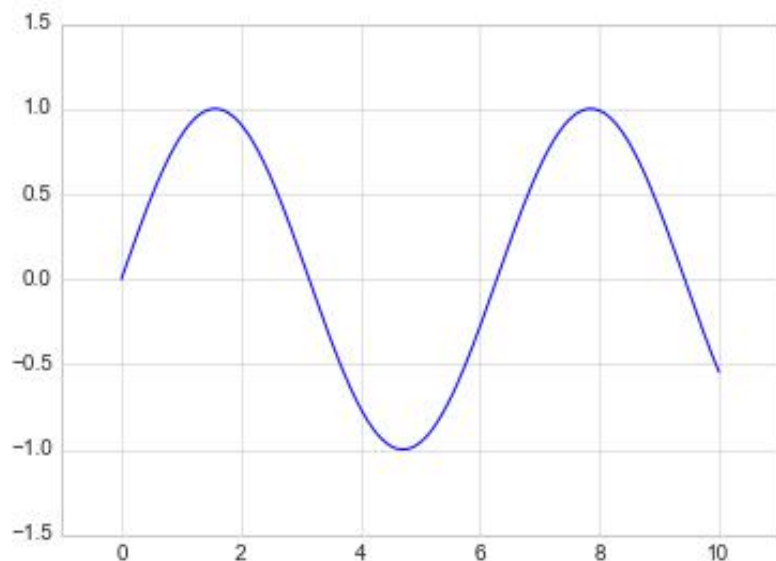
- `x = np.linspace(0, 10, 1000)`
- `plt.plot(x, x, '-g')` # 实线 green
- `# plt.plot(x, x, color='g',linestyle='-')`
- `plt.plot(x, x + 1, '--c')` # 虚线 cyan
- `plt.plot(x, x + 2, '-.k')` # 点划线 black
- `plt.plot(x, x + 3, ':r')` # 点实线 red



任务实现

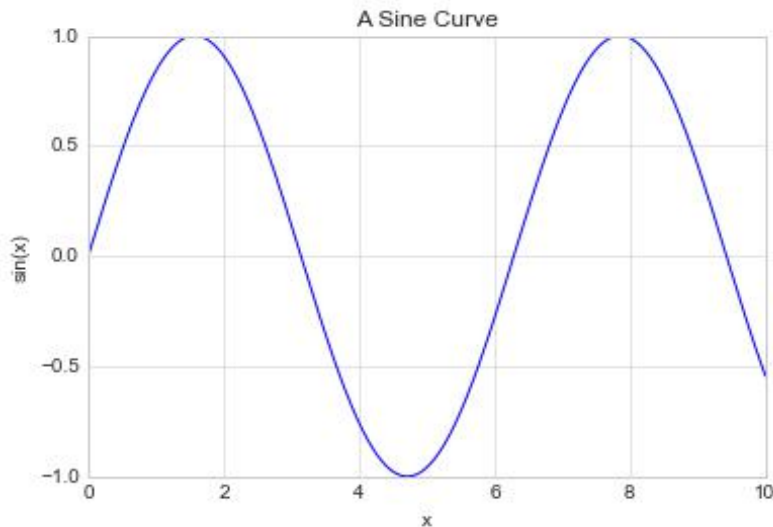
#坐标轴上下限

```
plt.plot(x, np.sin(x))  
plt.xlim(-1, 11)  
plt.ylim(-1.5, 1.5)  
#plt.axis([-1, 11, -1.5, 1.5])也可
```



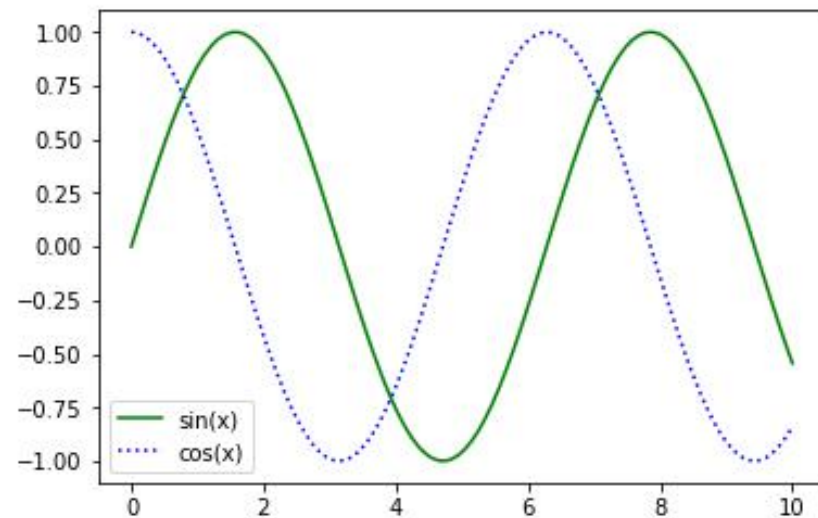
#设置标题和标签

```
plt.plot(x, np.sin(x))  
plt.title("A Sine Curve")  
plt.xlabel("x")  
plt.ylabel("sin(x)");
```



#设置图例

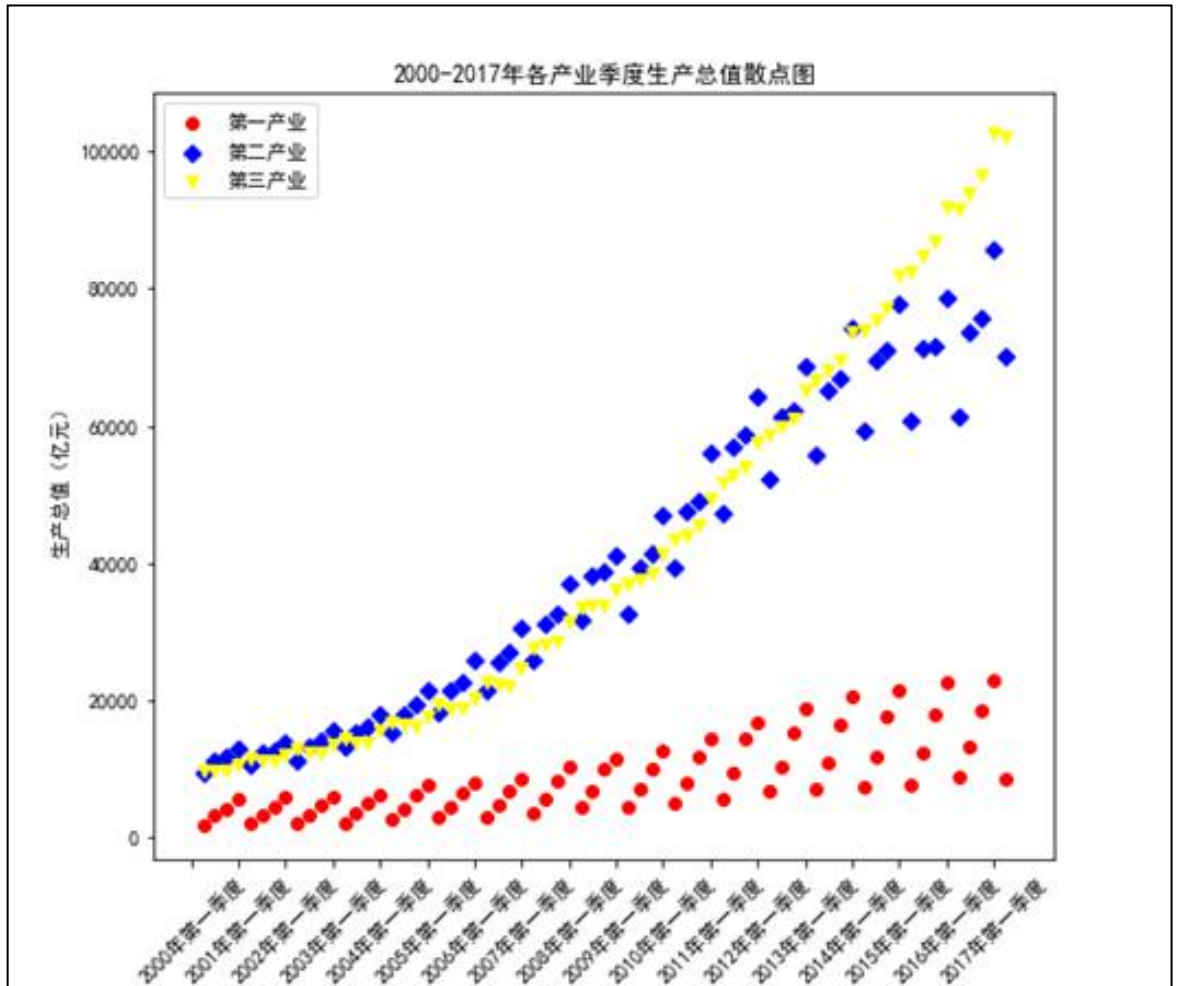
```
plt.plot(x, np.sin(x), '-g', label='sin(x)')  
plt.plot(x, np.cos(x), ':b', label='cos(x)')  
plt.legend()
```



绘制散点图

散点图

- 散点图 (scatter diagram) 又称为散点分布图, 是以一个特征为横坐标, 另一个特征为纵坐标, 利用坐标点 (散点) 的分布形态反映特征间的统计关系的一种图形。
- 值是由点在图表中的位置表示, 类别是由图表中的不同标记表示, 通常用于比较跨类别的数据。



绘制散点图

scatter函数

```
plt.scatter(x, y, s=None, c=None, marker=None, alpha=None, **kwargs)
```

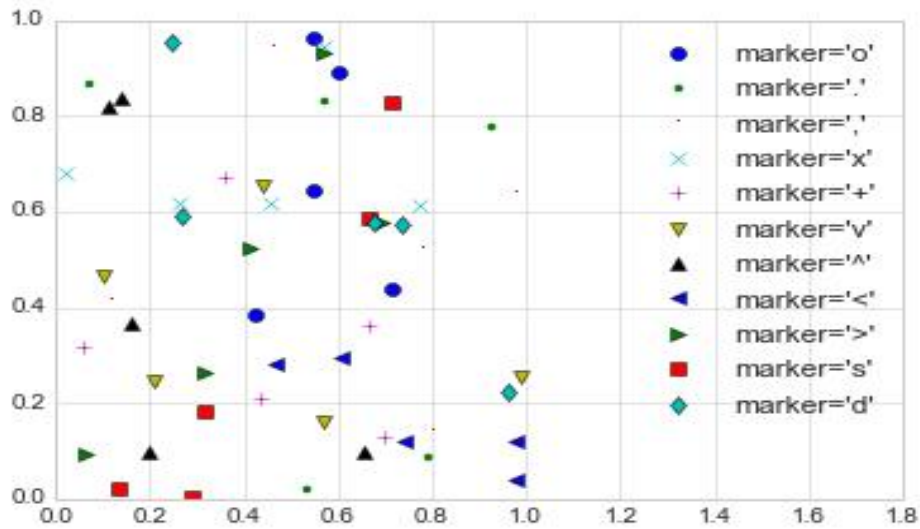
➤ 常用参数及说明如下表所示。

参数名称	说明
x, y	接收array。表示x轴和y轴对应的数据。无默认。
s	接收数值或者一维的array。指定点的大小，若传入一维array则表示每个点的大小。默认为None。
c	接收颜色或者一维的array。指定点的颜色，若传入一维array则表示每个点的颜色。默认为None
marker	接收特定string。表示绘制的点的类型。默认为None。
alpha	接收0-1的小数。表示点的透明度。默认为None。

绘制散点图

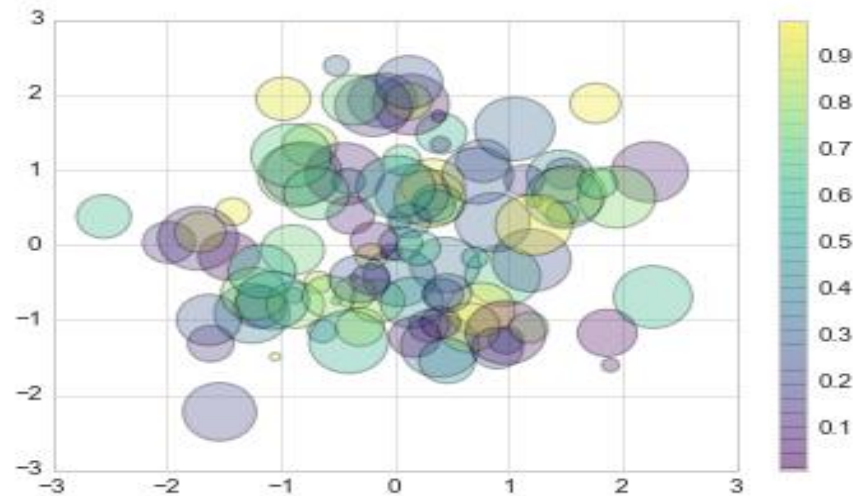
plot绘制散点图

```
rng = np.random.RandomState(0)
for marker in ['o', '.', ',', 'x', '+', 'v', '^', '<', '>', 's', 'd']:
    plt.plot(rng.rand(5), rng.rand(5), marker,
             label="marker='{0}'".format(marker))
plt.legend()
plt.xlim(0, 1.8)
```

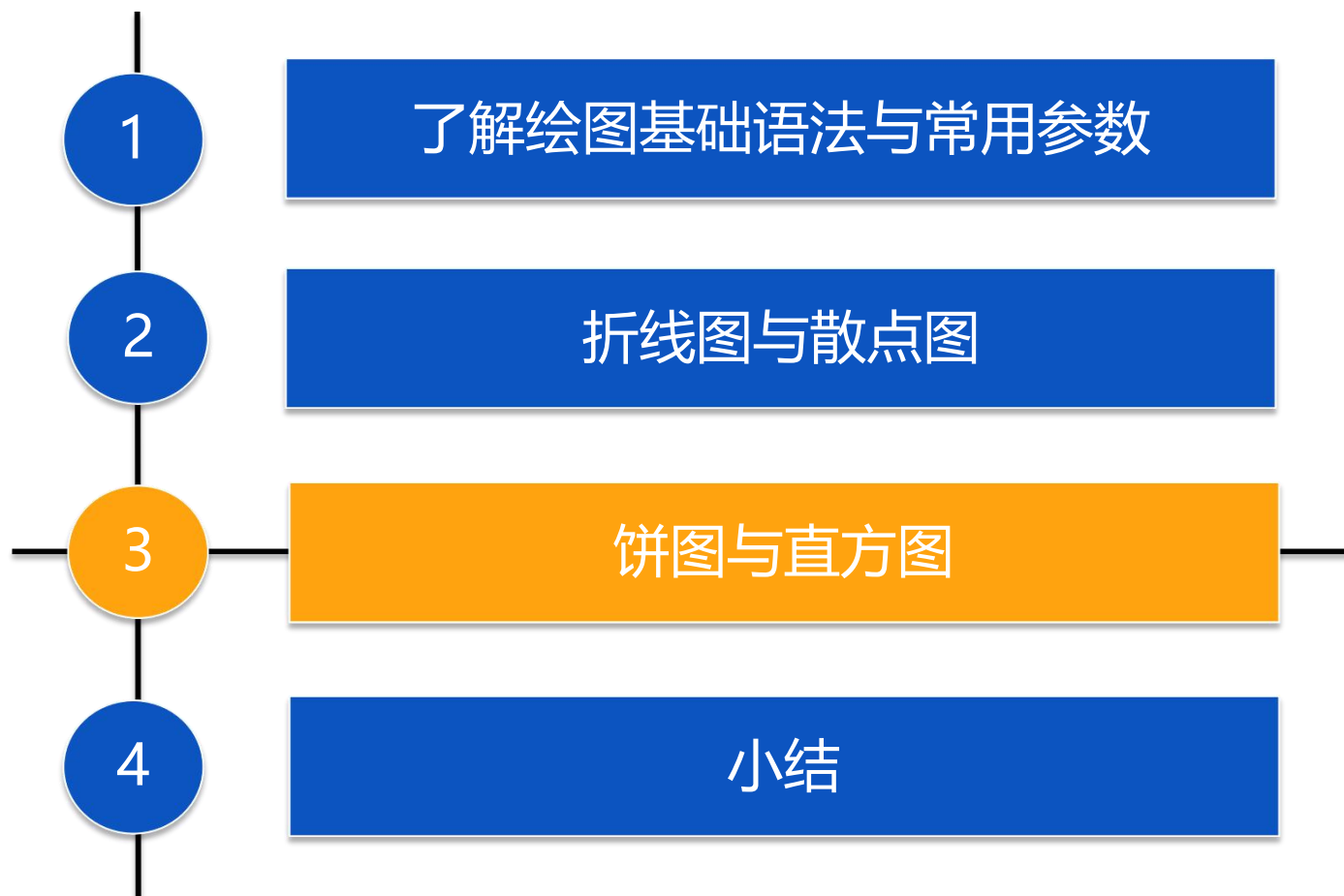


scatter可以单独控制每个点具有不同属性。大小 颜色 等

```
rng = np.random.RandomState(0)
x = rng.randn(100)
y = rng.randn(100)
colors = rng.rand(100)
sizes = 1000 * rng.rand(100)
plt.scatter(x, y, c=colors, s=sizes, alpha=0.3)
plt.colorbar(); # 显示颜色条
```



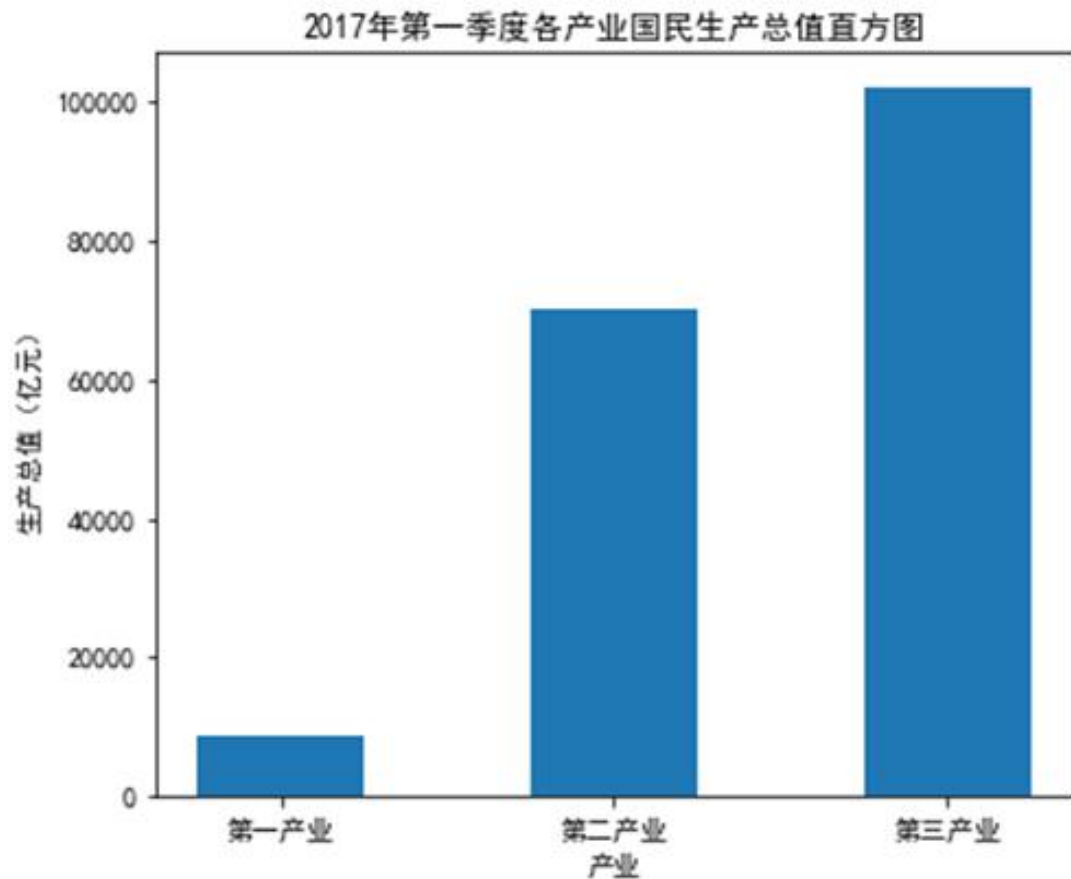
目录



绘制直方图

直方图

- 直方图 (Histogram) 又称质量分布图，是统计报告图的一种，由一系列高度不等的纵向条纹或线段表示数据分布的情况，一般用横轴表示数据所属类别，纵轴表示数量或者占比。
- 用直方图可以比较直观地看出产品质量特性的分布状态，便于判断其总体质量分布情况。直方图可以发现分布表无法发现的数据模式、样本的频率分布和总体的分布。



绘制直方图

bar函数

*plt.bar (left, height, width = 0.8, bottom = None, hold = None, data = None, **kwargs)*

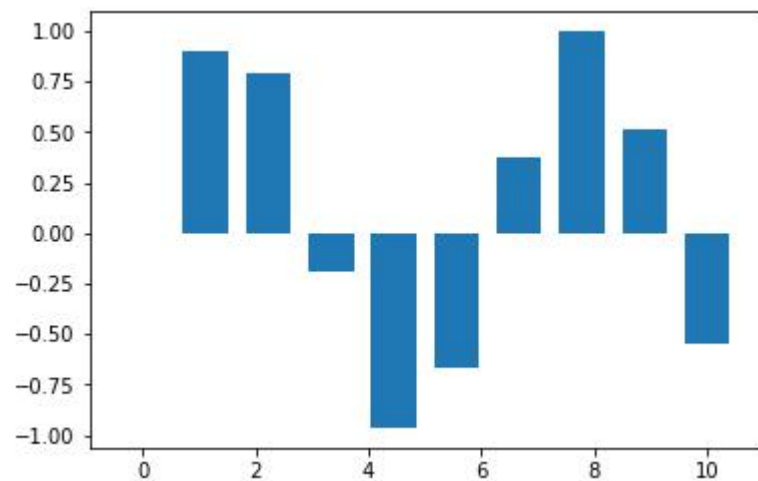
➤ 常用参数及说明如下表所示。

参数名称	说明
left	接收array。表示x轴数据。无默认。
height	接收array。表示x轴所代表数据的数量。无默认。
width	接收0-1之间的float。指定直方图宽度。默认为0.8。
color	接收特定string或者包含颜色字符串的array。表示直方图颜色。默认为None。

绘制直方图

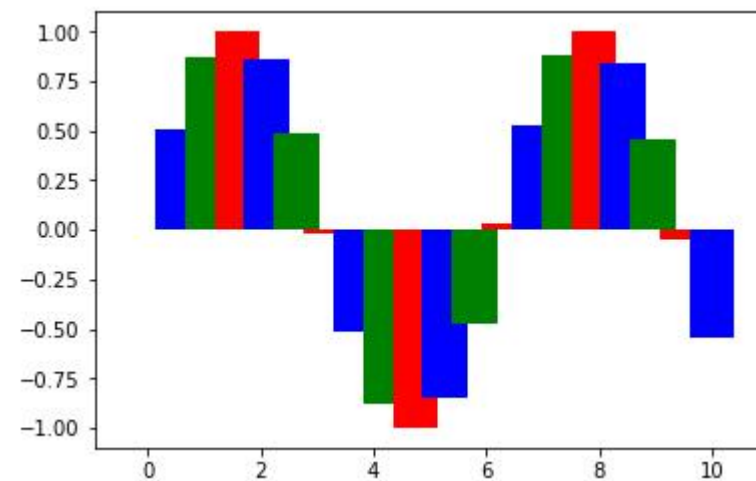
```
x=np.linspace(0,10,10)
```

```
plt.bar(x,np.sin(x))
```



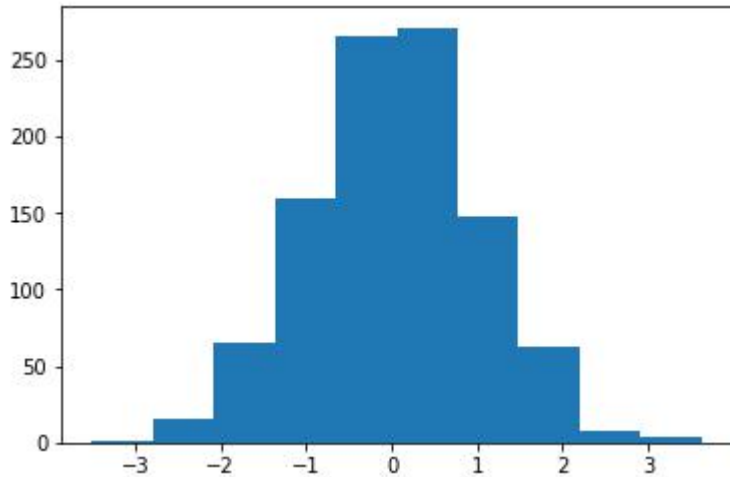
```
x=np.linspace(0,10,20)
```

```
plt.bar(x,np.sin(x),color=['red','blue','green'])
```



绘制频次直方图

```
import numpy as np
import matplotlib.pyplot as plt
data = np.random.randn(1000)
data.mean() #0.04366974487426336均值
data.std() #0.9860214413153232 方差
plt.hist(data)
```

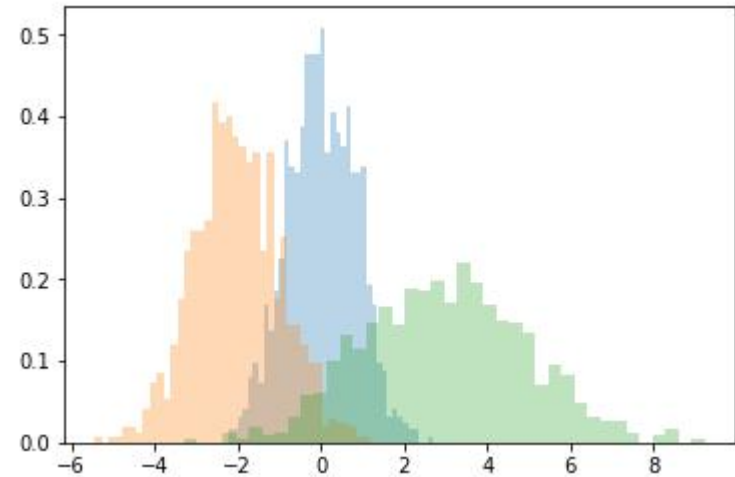


```
x1 = np.random.normal(0, 0.8, 1000)
x2 = np.random.normal(-2, 1, 1000)
x3 = np.random.normal(3, 2, 1000)
kwargs = dict(histtype='stepfilled', alpha=0.3, normed=True,
bins=40)
plt.hist(x1, **kwargs)
plt.hist(x2, **kwargs)
plt.hist(x3, **kwargs);
```

normed: 是否将得到的直方图向量归一化。默认为0

histtype: 直方图类型，
'bar' ,
'barstacked' ,
'step' , 'stepfilled'

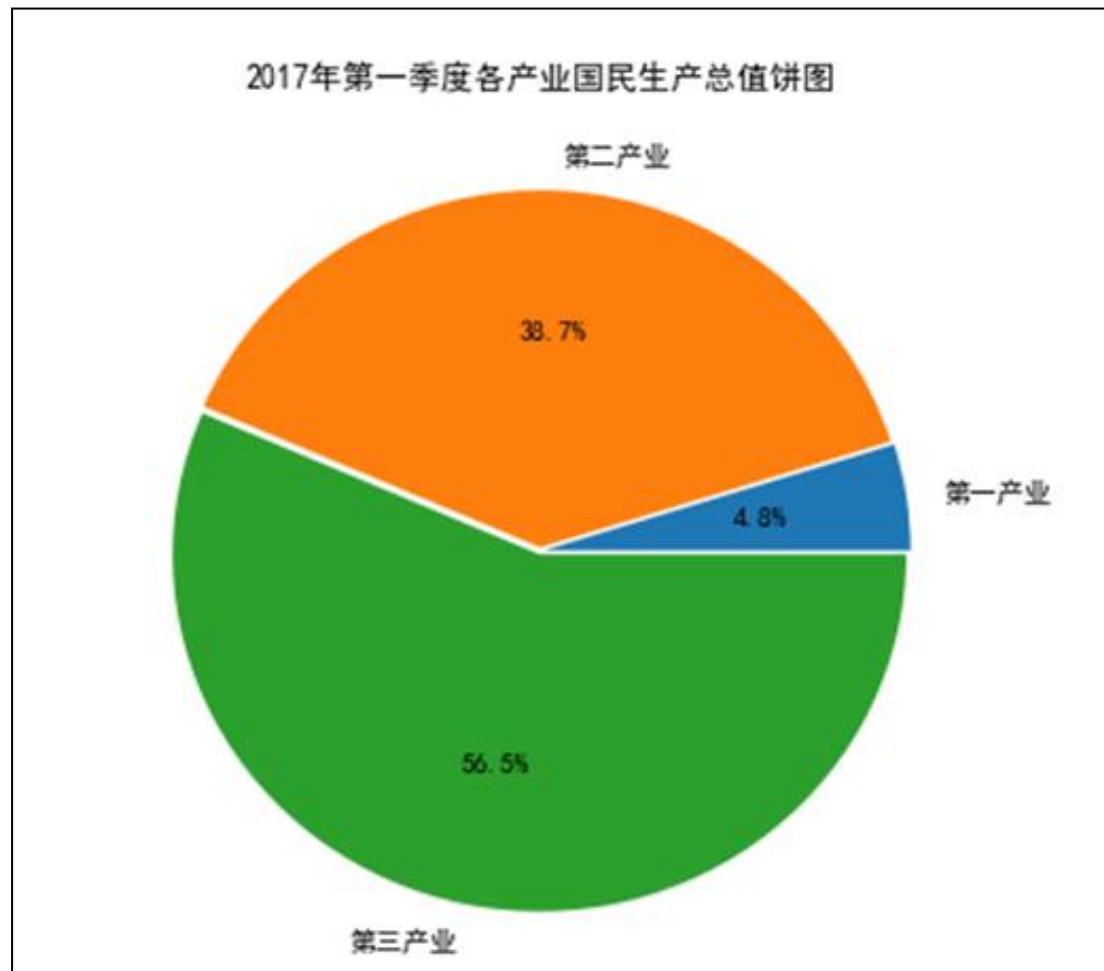
bins 直方图柱形数量



绘制饼图

饼图

- 饼图 (Pie Graph) 是将各项的大小与各项总和的比例显示在一张“饼”中，以“饼”的大小来确定每一项的占比。
- 饼图可以比较清楚地反映出部分与部分、部分与整体之间的比例关系，易于显示每组数据相对于总数的大小，而且显现方式直观。



绘制饼图

pie函数

matplotlib.pyplot.pie(x, explode=None, labels=None, colors=None, autopct=None, pctdistance=0.6, shadow=False, labeldistance=1.1, startangle=None, radius=None, ...)

➤ 常用参数及说明如下表所示。

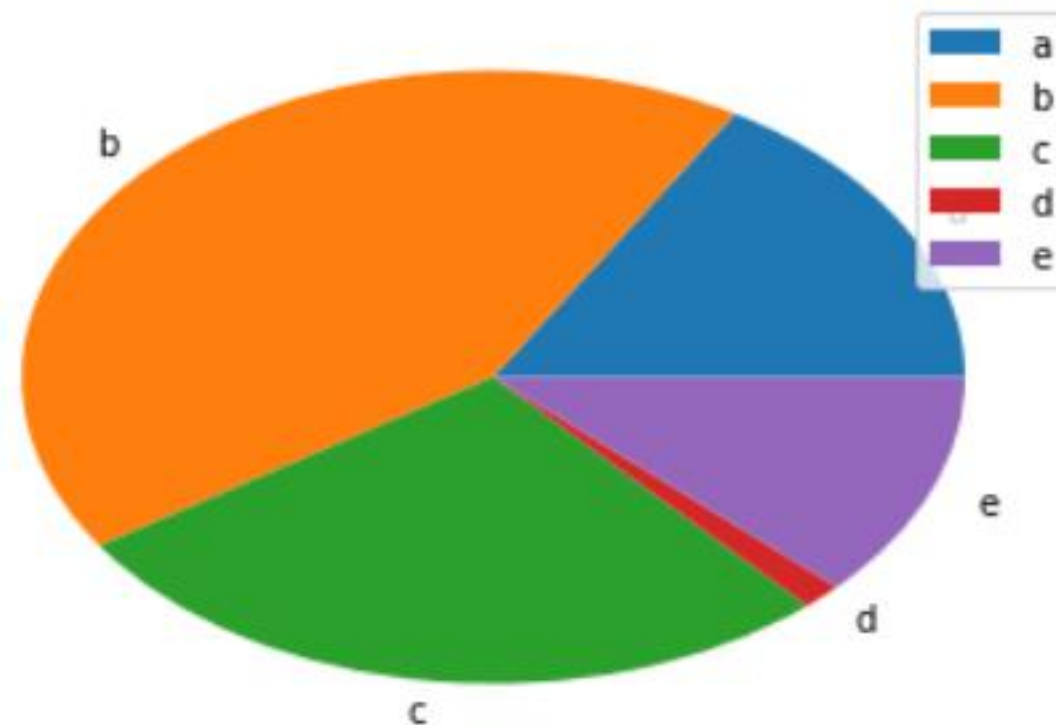
参数名称	说明	参数名称	说明
x	接收array。表示用于绘制撇的数据。无默认。	autopct	接收特定string。指定数值的显示方式。默认为None。
explode	接收array。表示指定项离饼图圆心为n个半径。默认为None。	pctdistance	接收float。指定每一项的比例和距离饼图圆心n个半径。默认为0.6。
labels	接收array。指定每一项的名称。默认为None。	labeldistance	接收float。指定每一项的名称和距离饼图圆心多少个半径。默认为1.1。
color	接收特定string或者包含颜色字符串的array。表示饼图颜色。默认为None。	radius	接收float。表示饼图的半径。默认为1。

绘制饼图

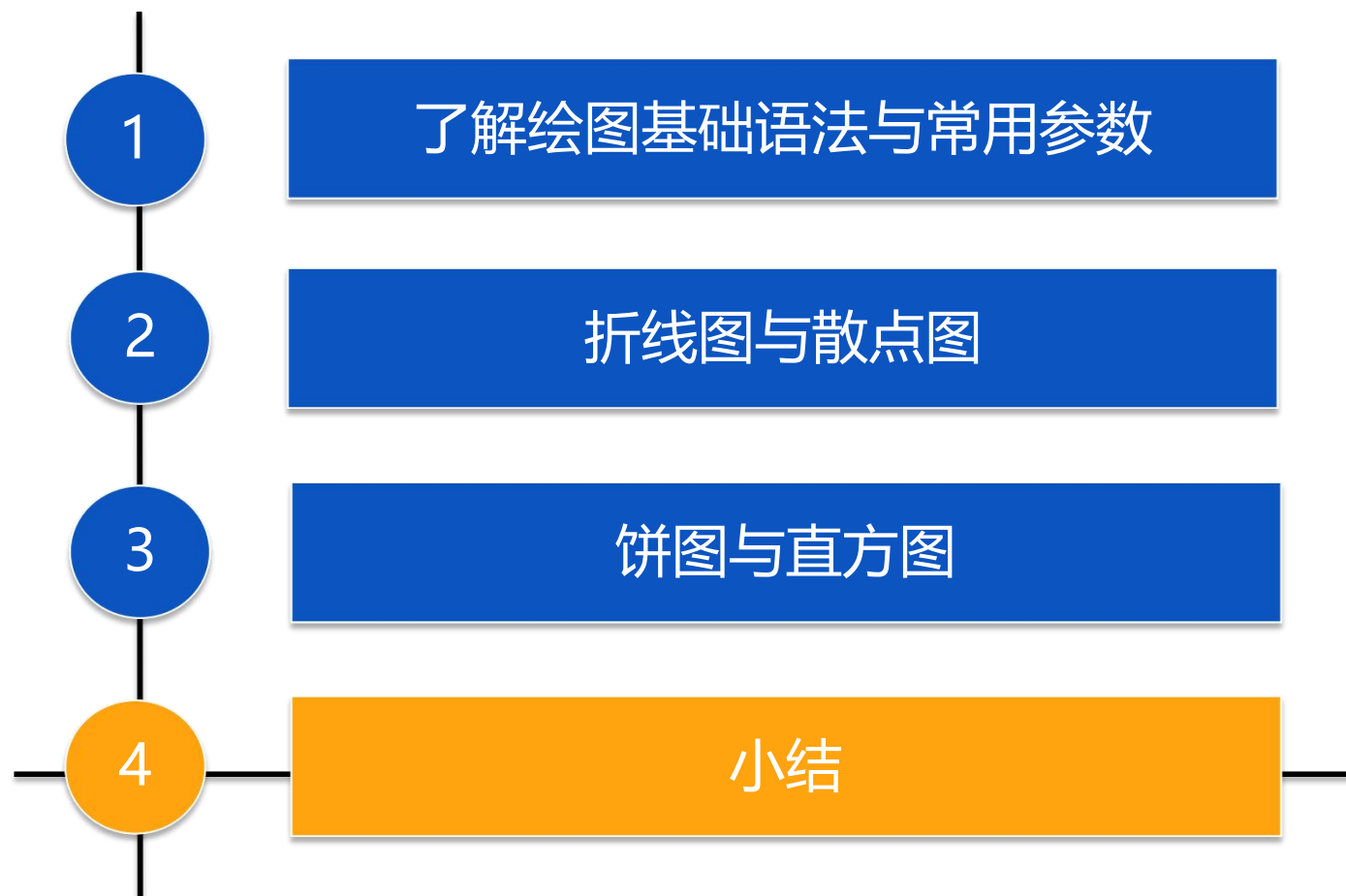
饼图

```
plt.pie(x,labels=['a','b','c','d','e'])
```

```
plt.legend()
```



目录



美国出生人口数据分析

要求分别统计不同年和月男女出生人口信息。

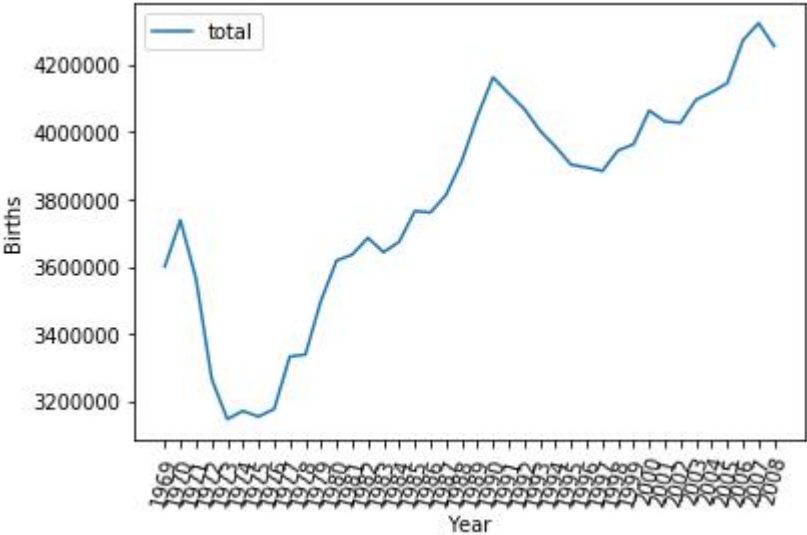
```
import numpy as np
import matplotlib.pyplot as plt

birth = np.loadtxt("d:/births.csv",dtype=np.int,usecols=(0,1,4),delimiter=" ",skiprows=1)
gender = np.loadtxt("d:/births.csv",dtype=np.str,usecols=3,delimiter=" ",skiprows=1)

#统计不同年份的出生人数
y=birth[:,0]
m=birth[:,1]
s=birth[:,2]
x=np.arange(year.size)
year=np.sort(np.unique(y)) #去重后排序
count1=np.zeros_like(year)

for i in x:
    j=np.sum(s[y==year[i]])
    count1[i]=j

plt.plot(x,count1,label='total')
plt.xlabel("Year")
plt.ylabel("Births")
plt.xticks(x,year,rotation=75)
plt.legend()
plt.show()
```



	A	B	C	D	E
1	year	month	day	gender	births
2	1969	1	1	F	4046
3	1969	1	1	M	4440
4	1969	1	2	F	4454
5	1969	1	2	M	4548
6	1969	1	3	F	4548
7	1969	1	3	M	4994
8	1969	1	4	F	4440
9	1969	1	4	M	4520
10	1969	1	5	F	4192
11	1969	1	5	M	4198
12	1969	1	6	F	4710
13	1969	1	6	M	4850
14	1969	1	7	F	4646
15	1969	1	7	M	5092
16	1969	1	8	F	4800
17	1969	1	8	M	4934
18	1969	1	9	F	4592
19	1969	1	9	M	4842

鸢尾花数据集可视化

Python的Sklearn机器学习库中自带的数据集——鸢尾花数据集。

数据集共包含花瓣长度、花瓣宽度、花萼长度、花萼宽度四个特征4个特征变量、1个类别变量，共有150个样本。类别变量分别对应鸢尾花的三个类别，分别是山鸢尾 (Iris-setosa)、变色鸢尾 (Iris-versicolor)和维吉尼亚鸢尾(Iris-virginica)。

可以通过seaborn导入到pandas对象

```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head()
```

#iris为pandas dataframe对象

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

3、鸢尾花数据集介绍

通过sklearn.datasets扩展包中的load_iris()函数导入鸢尾花数据集，该Iris中有两个属性，分别是：iris.data和iris.target。

data里是一个二维数组，每一列代表了萼片或花瓣的长宽，一共4列，每一列代表某个被测量的鸢尾植物，一共采样了150条记录。

target是一个一维数组，存储了data中每条记录属于哪一类鸢尾植物，数组长度是150，数组元素的值因为共有3类鸢尾植物，所以不同值只有3个。

Iris Setosa（山鸢尾）

Iris Versicolour（杂色鸢尾）

Iris Virginica（维吉尼亚鸢尾）

```
from sklearn.datasets import load_iris
#载入数据集
iris = load_iris()
#输出数据集
print(iris.data[0:10]) #为numpy数组， 150,4
```

```
print(iris.target[0:10])
```

```
iris.data[0:10]
```

```
Out[8]:
```

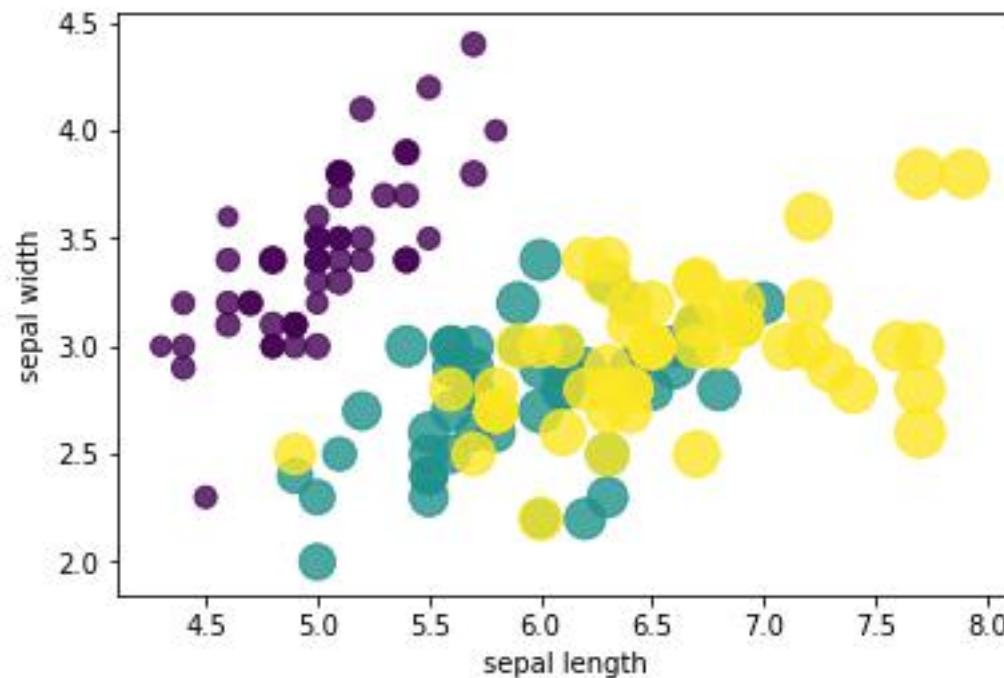
```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1]])
```

```
[0 0 0 0 0 0 0 0 0 0]
```

Scikit-Learn自带鸢尾花数据集，包含三种花。采集了花瓣和花萼的长度和宽度4个特征。

```
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
x=load_iris().data
y=load_iris().target
plt.scatter(x[:,0],x[:,1],s=10*x[:,2]+x[:,3],c=y)
plt.xlabel("sepal length")
plt.ylabel("sepal width")
```

散点图可以让我们看到不同维度的数据，每个点的坐标值(x,y)分别表示花萼的长度和宽度，点的大小表示花瓣的长度和宽度，颜色对应不同种类的花型。





Thank you!