

第6章 数据可视化

6.1 用Pillow操作图像

6.2 用Matplotlib绘图

6.3 调用Echarts

习题

第6章 数据可视化

- 借助于图形化手段，数据可视化可以清晰直观地表达信息。数据可视化实现了科学可视化领域与信息可视化领域的统一。
- 在数据分析工作中，尤其要重视数据可视化，因为错误或不充分的数据表示方法可能会毁掉原本很出色的数据分析工作。
- `pillow`、`matplotlib`等都是实现数据可视化的库，可以绘制的图形包括线图、直方图、饼图、散点图以及误差线图等；可以比较方便的定制图形的各种属性比如图线的类型、颜色、粗细、字体的大小等；它能够很好地支持一部分`TeX`排版命令，可以比较美观地显示图形中的数学公式。

第6章 数据可视化

□ 6.1 用Pillow操作图像

第6章 数据可视化



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0

第6章 数据可视化



Colour Image



Red



Green



Blue

31	32	33	34	35	6	77	78	79	G
41	42	43	44	45	16	87	88	89	
51	52	53	54	55					
61	62	63	64	65					B
71	72	73	74	75					
81	82	83	84	85					

第6章 数据可视化

用Pillow操作图像

图像处理是一门应用非常广的技术，PIL(Python Imaging Library)是Python常用的图像处理库，支持大量图像格式，并提供操作图像的强大功能，包括新建图像、裁剪图像、复制图像、粘贴图像、调整图像的大小、旋转和翻转图像、图像滤波、调色板、添加文字等功能，这些功能只需要简单的代码即可完成。PIL仅支持到Python 2.7，Python 3.x使用兼容PIL的版本Pillow，它在PIL的基础上加入了许多新特性。详细了解PIL的强大功能，请参考Pillow官方文档：<https://pillow.readthedocs.io/en/stable/>

第6章 数据可视化

图像的组成：都是由像素组成的。

像素点的描述：颜色值及其在图像中的位置。

1. 图像的属性

size属性：表示图像的分辨率，即图像的宽和高（单位为像素），是一个二元的tuple，如(300,200)，表示图像的宽为300，高为200。

mode属性：表示图像的模式，常用的模式为：L (luminance)表示灰度图，RGB表示真彩色图，CMYK表示出版图像。

format属性：表示图像格式或来源，如果图像不是从文件读取，值为None。

palette属性：表示调色板，返回一个ImagePalette类型

2. 图像空间坐标系：图像中的默认坐标系为：左上角是坐标原点(0, 0)，水平向右X轴，垂直向下Y轴。

3. 图像的颜色表示：计算机通常将图像中像素点的值用RGB值表示，或者再加上alpha值（通透度，透明度），称为RGBA值。在Pillow中，RGBA的值表示为由4个整数组成的元组，分别是R、G、B、A，整数的取值范围0~255，如(255, 0, 0, 255)代表红色，A为0表示透明，255表示不透明，当alpha值为0时，无论是什么颜色，该颜色都不可见。

第6章 数据可视化

图像处理中常用的模块和函数

Image模块中最重要的类就是**Image**，它代表一张图片，可以通过以下几种方式实例化：从文件中读取图片，处理其他图片得到，或者直接创建一个图片。

(1) 从文件中读取图片，创建**Image**对象

Image.open(filename)

(2) 处理其他图片得到**Image**对象

Image.crop(rect)

(3) 直接新建空白图像

Image.new(mode,size,color)

第6章 数据可视化

图像处理中常用的模块和函数

除了open函数外，其他方法需要通过Image 类的实例进行调用。

(1) 图像的读取和保存方法：

读取图像：

open(filename): filename 为文件完整的路径表示，Pillow库支持相当多的图像格式。例如：`Image.open("d:\图片1.jpg")`

保存图像：

save(filename)函数:Image模块中的save()函数可以保存图像，该函数还可以提供第二个参数，用于指定文件的保存格式，如果没有第二个参数，那么文件名中的扩展名用来指定文件格式。例如：`save("d:\图图.jpg")`

(2) 显示图像方法：

show(): 不需要参数，直接显示图像对象

(3) 从像中截取子图

crop(rect): 提供一个rect参数，表示截取子图在原始图像中的矩形区域。例如：`rect = (50,50, 200,150)`，`crop(rect)`表示新图为源图(50,50)到(199,149)这部分区域的子图

第6章 数据可视化

(4) 粘贴图像

paste(Image, rect): 第一个参数为Image对象，第二个参数为矩形对象，表示把第一个参数的图像贴到源图像的矩形区域出，注意第一个参数Image对象的size必须和矩形对象保证尺寸一致，此外，矩形区域不能在图像外。

(5) 几何变换方法:

resize(size) : 缩放图像，提供一个tuple参数，表示新图像的大小。例如：
resize((640,640))，表示新生成一个图像，新图像是源图像经过变换后，尺寸为640*640。

rotate(angle): 提供一个int参数，表示逆时针旋转的角度，0-360之间。例如：
rotate(45)表示新图像是源图像逆时针旋转45度得来的。

transpose(sign): 提供一个符号常量，Pillow通过此函数对于一些常见的旋转作了专门的定义。例如：**transpose(Image.ROTATE_90)**，表示逆时针旋转90度
transpose(Image.FLIP_LEFT_RIGHT) 左右对换。

(6) 色彩空间变换

convert(string) : 提供一个字符串参数，表示图像的mode属性。该函数可以用来将图像转换为不同色彩模式，如将彩色图像转换为灰度图像等。例如：**convert("L")**。

第6章 数据可视化

(7) 图像滤波

图像滤波，即在尽量保留图像细节特征的条件下对目标图像的噪声进行抑制，是图像预处理中不可缺少的操作。

在ImageFilter 模块中，提供了图像滤波filter()函数，用于图像的滤波增强。

filter(ImageFilter.function)：提供一个参数，表示滤波增强的方式。在

ImageFilter模块中，预先定义了很多增强滤波器。表10-1 ImageFilter类的预定义过滤方法

(8) 图像增强

图像滤波增强处理实质上就是运用滤波技术来增强图像的某些空间频率特征，以改善地物目标与领域或背景之间的灰度反差，如调节图像的颜色、对比度、饱和度和锐化等等。

ImageEnhance.Contrast(Image)：提供一个图像对象，调整图像的对比度

ImageEnhance.Color(Image)：提供一个图像对象，调整图像的颜色平衡

ImageEnhance.enhance(factor)：对选择属性的数值增强factor倍

ImageEnhance.Brightness(Image)：提供一个图像对象，调整图像的亮度

ImageEnhance.Sharpness(Image)：提供一个图像对象，调整图像的锐度

第6章 数据可视化

图像处理中常用的模块和函数

ImageDraw模块中包含了**ImageDraw**类，它支持各种几何图形的绘制和文本的绘制，如直线、椭圆、弧、弦、多边形以及文字等。下面介绍**ImageDraw**类中的几个函数：

Draw(image)：创建一个可以在给定图像上绘图的绘图对象。例如：

```
im= Image.open('d:\图片1.jpg' )
```

```
draw =ImageDraw.Draw(im)
```

draw.arc(xy, start, end, options)：在给定的区域内，在开始和结束角度之间绘制一条弧（圆的一部分）。例如：**draw.arc((0,0,100,100),0, 180, fill = (0,255,0))**，表示在源图的（0,0）到（100,100）这个矩形内画一个圆，顺时针取0度到180度之间的半圆弧，半圆弧颜色为绿色

draw.text(position,string, options)：在给定的位置绘制一个字符。参数**position**给出了文本的左上角的位置，参数**options**的**font**用于指定所用字体。例如：**draw.text((0,0),"Hello", fill = (0,255,0))**，表示在源图的(0,0)位置绘制绿色的Hello。

第6章 数据可视化

对图片进行灰度处理并进行边缘检测 demo

```
from PIL import Image, ImageFilter
```

```
im = Image.open("d:\图片1.jpg")
```

```
print(im.mode," ",im.size)
```

```
out = im.convert("L")
```

```
out=out.filter(ImageFilter.FIND_EDGES)
```

```
out.show()
```

```
out.save("d:\图图.jpg")
```

>>>运行结果如图

RGB (466, 174)



常用滤波算法: ImageFilter.BLUR、
ImageFilter.CONTOUR、
ImageFilter.DETAIL、
ImageFilter.EDGE_ENHANCE、
ImageFilter.EDGE_ENHANCE_MORE、
ImageFilter.EMBOSS、
ImageFilter.SMOOTH、
ImageFilter.SHARPEN.....

第6章 数据可视化

创建一个新的200*100的画布，并绘制两条对角线 demo1

```
from PIL import Image, ImageDraw
im= Image.new('RGB', (200, 100), '#FF0000' )
draw =ImageDraw.Draw(im)
draw.line((0,0,im.size[0],im.size[1]), fill=128,width=10)
draw.line((0,im.size[1], im.size[0], 0), fill = 128,width=10)
im.show()
del draw
```



第6章 数据可视化

用pillow生成字母验证码图片 demo2

生成字母验证码图片的算法为：首先随机几个验证码字符和颜色,再将字符按随机颜色draw到画布上,最后添加干扰和保存图像

```
import random
```

```
import string
```

```
from PIL import Image, ImageFilter, ImageFont, ImageDraw
```

```
def getcolor1():
```

```
    red = random.randrange(64, 255)
```

```
    green = random.randrange(64, 255)
```

```
    blue = random.randrange(64, 255)
```

```
    return (red,green,blue)
```

```
def getcolor2():
```

```
    red = random.randrange(32, 127)
```

```
    green = random.randrange(32, 127)
```

```
    blue = random.randrange(32, 127)
```

```
    return (red,green,blue)
```

第6章 数据可视化

用pillow生成字母验证码图片

```
def getcode():  
    source =  
'1234567890qwertyuiopasdfghjklzxcvbnmQWERTYUIOPAS  
DFGHJKLZXCVBNM'  
    resultt = "  
    for i in range(4):  
        resultt += source[random.randrange(len(source))]  
    return resultt
```


第6章 数据可视化

```
def Verification_code():
    size = (200,100)
    #随机的颜色，只能稍稍提高安全
    color = getcolor1()
    #获取验证码中的四个字符
    code = getcode()
    #创建画布
    image = Image.new('RGB',size,color)
    imagedraw = ImageDraw.Draw(image,'RGB')
    #选择字体
    imagefont=ImageFont.truetype("C:\\WINDOWS\\Fonts\\SIMLI.TTF", 80)
    for i in range(4):
        position = (20+40*i,10)
        #写入验证码
        imagedraw.text(position,code[i],font=imagefont,fill=getcolor1())
    #加点进行干扰
    for j in range(500):
        imagedraw.point((random.randrange(200),random.randrange(100)), fill=getcolor2())
    #对图像进行模糊滤镜处理
    image = image.filter(ImageFilter.BLUR)
    image.save("d:\\动态图片.jpg",dpi=(300.0, 300.0))
Verification_code()
```



第6章 数据可视化

□ 6.2 用Matplotlib绘图

第6章 数据可视化

Matplotlib常用函数介绍

Matplotlib库中提供的Matplotlib.pyplot模块，能够以各种硬拷贝格式和跨平台的交互式环境快速地创建出版质量级别的多种类型的图形，如折线图，散点图，条形图，饼图，堆叠图，3D 图和地图图形等等。在一个图形窗口上，可以绘制多个图例，多个子图，也可以放大局部区域等。程序员利用Matplotlib进行图形开发，仅需要几行代码，便可达到目的，但程序员应该根据数据的特点，进行图表类型的选择，便于用户很容易通过图形理解分析数据。

第6章 数据可视化

Matplotlib常用函数介绍

matplotlib.pyplot.figure(): 创建新的图形窗口，如果不显式建立图形窗口，系统会自动建立图形窗口

matplotlib.pyplot.close(): 关闭图形窗口

matplotlib.pyplot.show(): 显示图形

matplotlib.pyplot.axis(rect): 用来指定坐标轴的视窗，例如：

matplotlib.pyplot.axis([0, 6, 0, 20])，表示x轴的长度为0到6，y轴的长度为0到20。如果画图时，不指定x轴的长度和y轴的长度，系统会按要处理的数据特性，自动定义轴的长度。

matplotlib.pyplot.subplot(numrows[, numcols[, fignum]): 该函数相当于把原图形窗口分割成numrows*numcols个子窗口，目前的子窗口是第fignum个子窗口。子窗口的编号：从左向右、从上向下，顺序编号。例如：**subplot(211)**等同于**subplot(2, 1, 1)**，

matplotlib.pyplot.xlabel(string) : 设置x轴标签

matplotlib.pyplot.ylabel(string) : 设置y轴标签

matplotlib.pyplot.title(string) : 设置图形的标题

matplotlib.pyplot.legend() : 按缺省样式生成默认图例

matplotlib.pyplot.plot(*args, **kwargs): 绘制折线图

matplotlib.pyplot.pie(*args, **kwargs): 绘制饼图

matplotlib.pyplot.hist (*args, **kwargs): 绘制直方图

matplotlib.pyplot.bar(*args, **kwargs): 绘制条形图

第6章 数据可视化

折线图的函数定义及属性说明

定义：折线图，就是将多个(x,y)点，连接起来，生成一个折线图。

下面给出该函数的常见调用形式：

plot([x], y, [fmt], data=None, **kwargs)

函数用于绘制一条折线图，**x**若省略，则**plot**函数自动创建从0开始的 **x**坐标；**fmt**是字符串类型，用于描述颜色标志线型属性的值，格式为：'**[color][marker][line]**';
kwargs用于设定线型、线宽、坐标点的标志等等图形的其他属性。

plot([x], y, [fmt], [x2], y2, [fmt2], ..., **kwargs) #用于绘制多条折线图

折线图的常见调用形式：

```
plot(x,y,label='First Line',color='r',linewidth=2,linestyle='-',  
,marker='*',markersize=12)
```

或者写成：

```
matplotlib.pyplot.plot(x, y, 'r*-', label='First Line',linewidth=2,  
markersize=12)
```

第6章 数据可视化

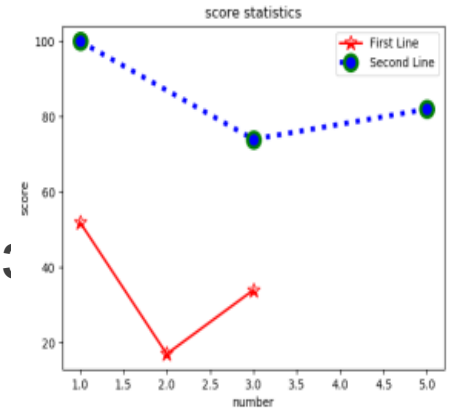
折线图的函数定义及属性说明

color	图形颜色
linestyle	线型: '-' (实线) , '--' (虚线) , '-.' (点虚线) , ':' (点线) , 'None', '' (空)
label	图形的图例名
Marker	坐标点的标志样式, ".", ",", "o", "v", "^", "<", "1", "2", "3", "4", "8", "s", "p", "*", "_"
linewidth	线宽
markeredgewidth	标志的边宽
markeredgecolor	标志的边颜色
markersize	标志的大小
fillstyle	标志填充的样式: 'full' , 'left' , 'right' , 'bottom' , 'top' , 'none' , 表示填满, 不填, 还是填一半颜色

第6章 数据可视化

绘制两条折线图 demo3

```
import matplotlib.pyplot as plt
x = [1,2,3]
y = [52,17,34] #x和y合成三个点: (1,52), (2,17), (3,34)
x2 = [1,3,5]
y2 = [100,74,82] #x2和y2合成三个点: (1,100), (3,74), (5,82)
plt.plot(x,y,label='First
Line',fillstyle='bottom',color='r',linewidth=2,linestyle='-',
'marker='*',markersize=12)
plt.plot(x2,y2,label='Second
Line',color='b',linewidth=4,linestyle=':',marker='o',markersize=12,marker
edgewidth=3,markeredgecolor='g')
plt.xlabel('number') #设置x轴标签
plt.ylabel('score') #设置y轴标签
plt.title('score statistics') #设置图形的标题
plt.legend() #按缺省样式生成默认图例
plt.show()
```



第6章 数据可视化

条形图案例 demo4

```
import matplotlib.pyplot as plt

plt.bar([1,3,5,7,9],[5,2,7,8,2], label="Example one")

plt.bar([2,4,6,8,10],[8,6,2,5,6], label="Example two", color='g')

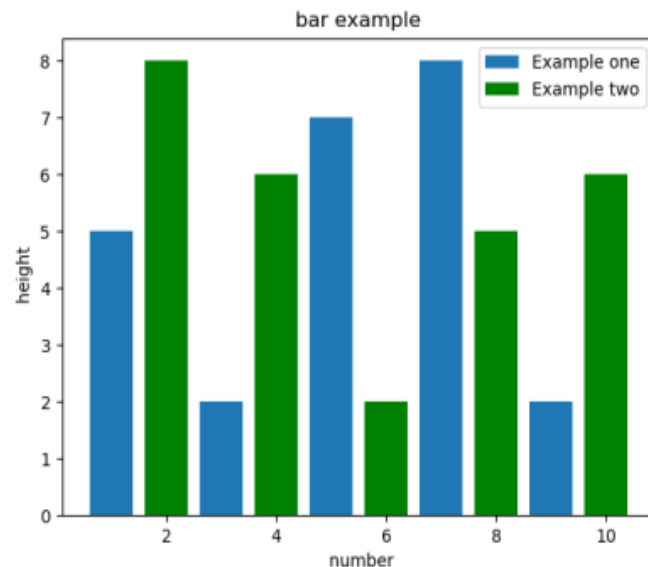
plt.xlabel(' number')

plt.ylabel(' height')

plt.title('bar example')

plt.legend()

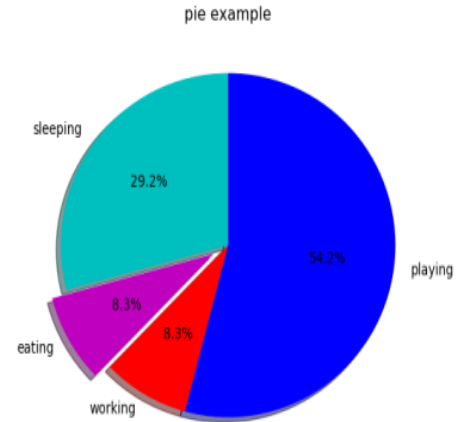
plt.show()
```



第6章 数据可视化

饼图案例 demo5

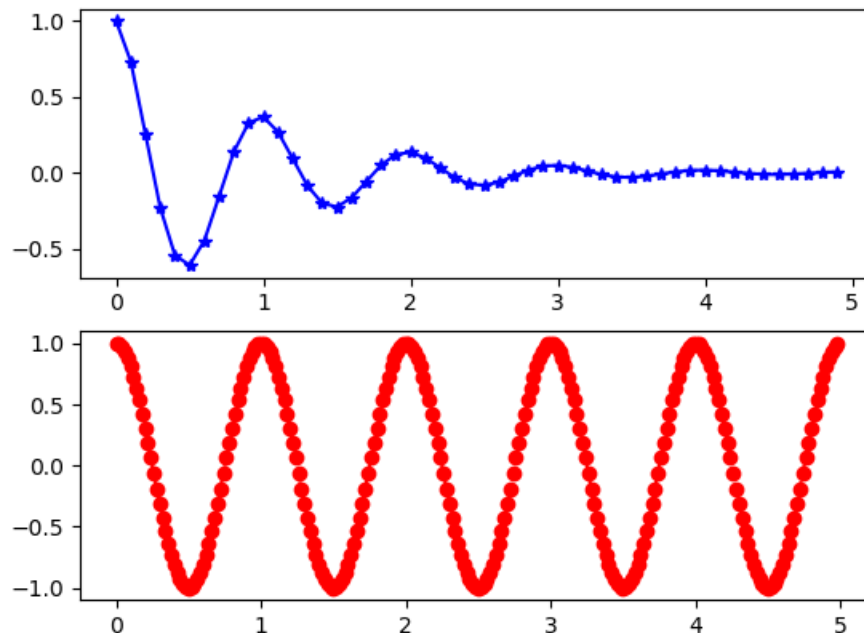
```
import matplotlib.pyplot as plt
slices = [7,2,2,13]
activities = ['sleeping','eating','working','playing']
cols = ['c','m','r','b']
plt.pie(slices, labels=activities,colors=cols, startangle=90, shadow=
True,      explode=(0,0.1,0,0), autopct='%1.1f%%')
plt.title('pie example')
plt.show()
```



第6章 数据可视化

创建2个子图，并绘制图形 demo6

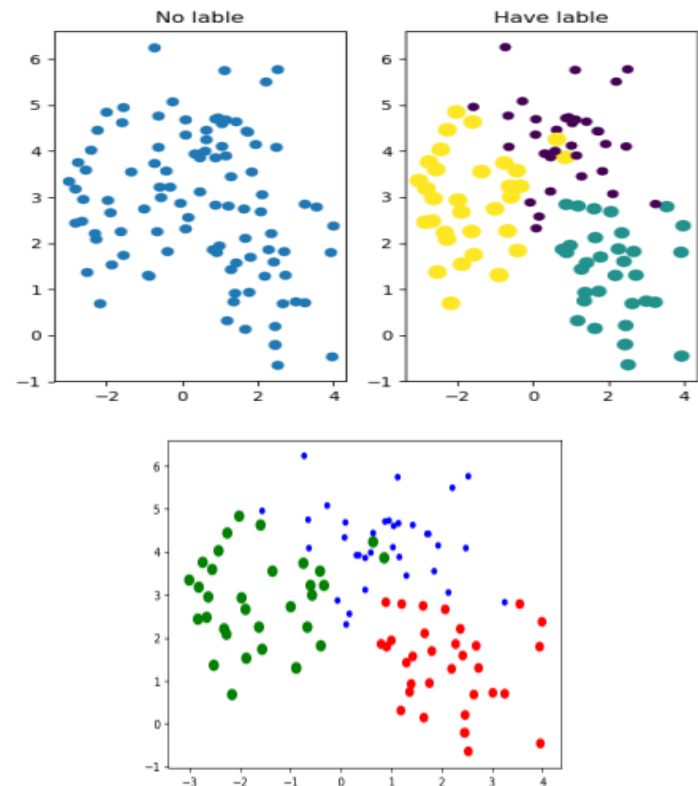
```
import matplotlib.pyplot as plt
import numpy as np
plt.figure()
plt.subplot(211)
x1 = np.arange(0.0, 5.0, 0.1)
y1 = np.exp(-x1) * np.cos(2*np.pi*x1)
plt.plot(x1, y1, 'b-*)
x2 = np.arange(0.0, 5.0, 0.02)
y2 = np.cos(2*np.pi*x2)
plt.subplot(212)
plt.plot(x2, y2, 'r--o')
plt.show()
plt.close()
```



第6章 数据可视化

散点图案例 demo9

```
from sklearn.datasets.samples_generator import make_blobs
import matplotlib.pyplot as plt
X, y = make_blobs(n_samples=100, centers=3, n_features=2, random_state=0)
y=y+1;
plt.figure(1) #新建一个名叫1的画图窗口
ax=plt.subplot(121)
plt.scatter(X[:,0],X[:,1])
ax.set_title('No lable')
ax=plt.subplot(122)
plt.scatter(X[:,0],X[:,1],y*30,y*30)
ax.set_title('Have lable')
plt.figure(2)
ax=plt.subplot(111)
id=(y==1)
plt.scatter(X[id,0],X[id,1],s=20,color='b')
id=(y==2)
plt.scatter(X[id,0],X[id,1],s=50,color='r')
id=(y==3)
plt.scatter(X[id,0],X[id,1],s=70,color='g')
plt.show()
```



第6章 数据可视化

□ 6.3 调用Echarts

第6章 数据可视化

ECharts，是百度提供的基于JavaScript实现的开源可视化库，可以流畅的运行在PC和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari等），其底层依赖轻量级的矢量图形库ZRender，并提供了组件解决多维度数据筛选、视图缩放平移、展示细节等交互功能，可高度个性化定制的数据可视化图表。

ECharts 提供了常规的折线图、柱状图、散点图、饼图、K线图，用于统计的盒形图，用于地理数据可视化的地图、热力图、线图，用于关系数据可视化的关系图等，**ECharts**支持包括二维表，key-value等多种格式的数据源，通过简单的设置encode属性就可以完成从数据到图形的映射。详细介绍见官网：

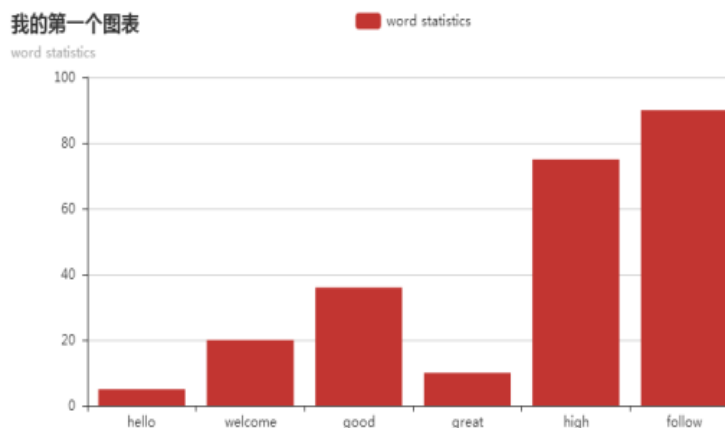
<https://pyecharts.org/#/zh-cn/intro>。

Python中提供了第三方库pyecharts，用于生成Echarts图表。

第6章 数据可视化

用柱状图显示每个单词的词频 demo10

```
from pyecharts import Bar
bar = Bar("我的第一个图表", "word statistics")
datax=["hello", "welcome", "good", "great", "high", "follow"]
datay=[5, 20, 36, 10, 75, 90]
bar.add("word statistics",datax ,datay ,is_more_utils=True)
bar.show_config()
bar.render(r"d:myfirstchart.html")
#生成html文件
```

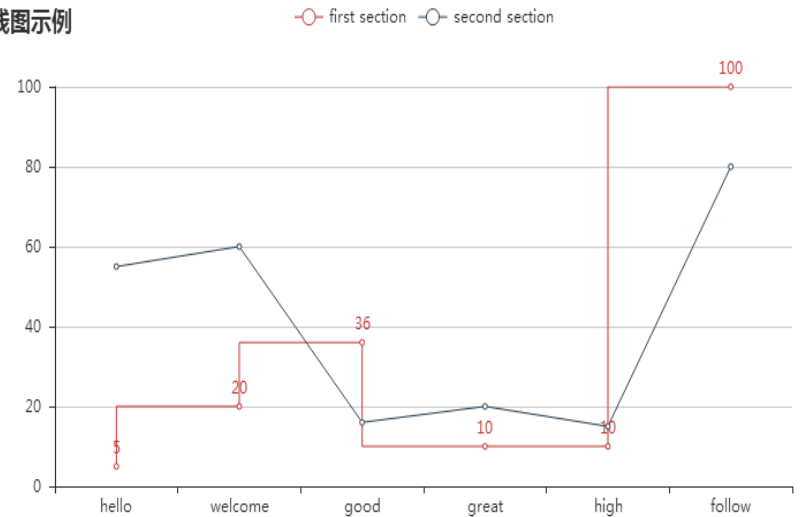


第6章 数据可视化

用折线图显示2个section的单词统计 demo8

```
from pyecharts import Line
attr=["hello", "welcome", "good", "great", "high", "follow"]
v1 =[5, 20, 36, 10, 10, 100]
v2 =[55, 60, 16, 20, 15, 80]
line =Line("折线图示例")
line.add("first section", attr, v1, is_step=True)
line.add("second section", attr, v2)
line.show_config()
line.render(r"d:mysecondchart.html")
```

折线图示例



第6章 数据可视化

在地图上显示各省的学生人数 demo7

```
import pandas as pd
from pyecharts import Map

data=[("广东",1000),("山东",3500),("河南",300),("四川",2100),
      ("江苏",1500),("安徽",5000),("浙江",800)]
data=pd.DataFrame(data)
data.columns=['province','number']
map=Map("各省学生数", "单位： 人", title_color="#fff", title_pos="center",
width=1200, height=600, background_color='#404a59')
attr=data['province']
value=data['number']
map.add("", attr, value, visual_range=[0, 5000], visual_text_color="#fff",
symbol_size=15, is_visualmap=True,is_label_show=True)
map.render(r"e:school.html")
```

```
pip install pyecharts==0.5.0
pip install echarts-countries-pypkg
pip install echarts-china-provinces-pypkg
pip install echarts-china-cities-pypkg
```


第6章 数据可视化

作业题：

1. 请从网上下载鸢尾花数据文件，用Echarts和Matplotlib的散点图绘制萼片（sepal）和花瓣（petal）的大小关系。

2. 用Echarts和Matplotlib绘制萼片和花瓣的大小关系。

3. 用Echarts和Matplotlib绘制萼片和花瓣的分布情况。

4. 用Echarts和Matplotlib绘制萼片和花瓣的分布情况。

5. 用Echarts实现你所在位置的标注。

列名	说明	类型
SepalLength	花萼长度	float
SepalWidth	花萼宽度	float
PetalLength	花瓣长度	float
PetalWidth	花瓣宽度	float
Class	类别变量。0 表示山鸢尾，1 表示变色鸢尾，2 表示维吉尼亚鸢尾。	int

萼片和花

分布情况。

鸢尾花数据集一共包含4个特征变量，1个类别变量。共有150个样本，iris是鸢尾植物，这里存储了其萼片和花瓣的长宽，共4个属性，鸢尾植物分三类。

第6章 数据可视化

THE END