Python与金融数据挖掘(11)

文欣秀

wenxinxiu@ecust.edu.cn

案例分析

英为财情数据显示，4月20日当天，德国宝马汽车公司(BMWG)收盘大跌3.62%，报100.02欧元/股，流通市值蒸发24.21亿欧元，折合人民币约183亿元，有网友戏称这是史上最贵的一杯冰淇淋。

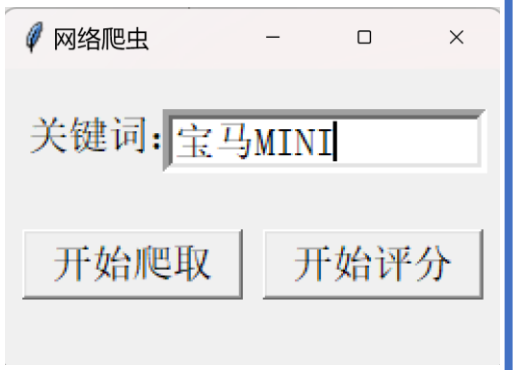
据了解，事发当日网友视频中的冰淇淋分为单球和双球，售价分别为35元和50元。而据相关预测，上海车展总访客量约为100万人，即便是访客全体每个人都拿一份冰淇淋，总价约为3500万元，这和宝马蒸发的183亿元相比，宝马明显是亏大了。有媒体尖锐的指出，这本身是次不错的营销，却因一杯冰淇淋搞砸了。

舆情数据按标题评分

```
score = []
title=["XX饼干成分不合格", "XX研发新产品","XX有偷税漏税行为"]
keywords = ['违约','不合格','偷税']
for i in range(len(title)):
    num = 0
    for k in keywords:
        if k in title[i]:
            num-=10
    score.append(num)
for i in range(len(title)):
    print("{}评分为{}分".format(title[i],score[i])
```

舆情数据评分系统（1）

```
from tkinter import *
from tkinter.messagebox import *
import requests
import re
#定义三个全局变量在函数之间共享数据
title=[]
href=[]
company=""
```



舆情数据评分系统（2）

```
def crawler():
    try:
        headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/69.0.3497.100 Safari/537.36'}
        global company,title,href
        company=E1.get()
        url = 'http://www.baidu.com/s?tn=news&rtt=1&wd=' + company
        res = requests.get(url, headers=headers).text
        p_href = '<h3 class="news-title_1YtI1 "><a href="(.*?)"'
        href = re.findall(p_href, res, re.S)
        p_title = '<h3 class="news-title_1YtI1 ">.*?>(.*?)</a>'
        title = re.findall(p_title, res, re.S)
        for i in range(len(title)):
            title[i] = title[i].strip()
            title[i] = re.sub('<.*?>', '', title[i])
            print(str(i + 1) + '.' + title[i])
            print(href[i])
        showinfo("结果","{}".format(company+'爬虫成功！'))
    except:
        showinfo("结果",{}.format(company+'爬虫失败！'))
```

```
1.争议中的宝马MINI该走向何方?
https://www.thepaper.cn/newsDetail_forward_22860022
2.从宝马mini冰淇淋事件谈互联网时代舆情应对
https://baijiahao.baidu.com/s?id=1764205007101848172&a
3.清醒点!宝马MINI的双标,可不光针对“中国的人”
https://baijiahao.baidu.com/s?id=1764201789610609306&a
4.宝马mini几只冰淇淋引发的狗血事件
```

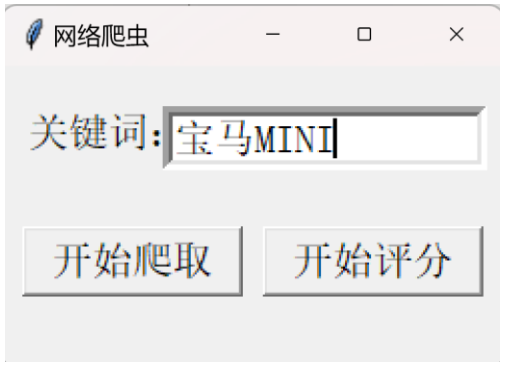
舆情数据评分系统（3）

```
def grade():
    global company,title,href
    score = []
    keywords = ['双标', '狗血'] # 这个关键词列表可以自己定义，这里只是为了演示
    for i in range(len(title)):
        num = 10
        # 获取新闻正文
        try:
            article = requests.get(href[i], headers=headers, timeout=10).text
        except:
            article = '爬取失败'
        # 只筛选真正的正文内容，旁边的滚动新闻之类的内容忽略
        p_article = '<p.*?>(.*?)</p>' # 有的时候p标签里还有class等无关内容
        article_main = re.findall(p_article, article) # 获取<p>标签里的正文信息
        article = ''.join(article_main) # 将列表转换成为字符串
        for k in keywords:
            if (k in article) or (k in title[i]):
                num -= 5
        score.append(num)
    for i in range(len(title)):
        print(title[i],score[i])
```

```
争议中的宝马MINI该走向何方？ 10
从宝马mini冰淇淋事件谈互联网时代舆情应对 10
清醒点!宝马MINI的双标,可不光针对“中国的人” 5
宝马mini几只冰淇淋引发的狗血事件 5
宝马mini事件中关于品牌舆情危机的22点思考 10
```

舆情数据评分系统（4）

```
root = Tk()
root.title("网络爬虫")
root.geometry("250x150")
L1 = Label(root, text="关键词：",font=20)
L1.place(x=10,y=20)
E1 = Entry(root, bd =5,font=20,width=15)
E1.place(x=80,y=20)
B1 = Button(root, text="开始爬取",font=20,width=10,command=crawler)
B1.place(x=10,y=80)
B2 = Button(root, text="开始评分",font=20,width=10,command=grade)
B2.place(x=130,y=80)
root.mainloop()
```



舆情数据评分系统搭建

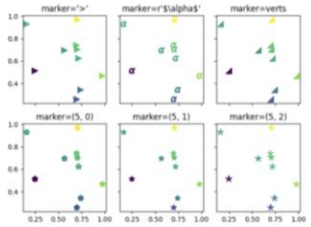
- 创建窗体和控件，用于输入新闻主题
- 编写爬虫模块，用于数据采集和清洗
- 编写舆情分析模块，用于数据的评分
- 编写数据库模块，用于存储统计数据
- 编写绘图模块，用于展示及相关性分析
- 编写机器学习算法模块，用于结果预测

Matplotlib

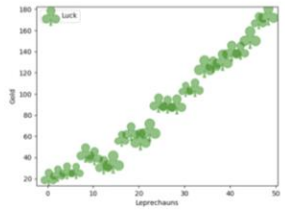
Matplotlib： 一个用来绘制二维图形的Python 模块。它可以绘制多种图形，如**直方图**、**散点图**以及**误差线图**等；可以方便地定制图形的各种属性，如类型、颜色、粗细、字体等，还可以美观地显示图中数学公式。

官网：https://matplotlib.org/

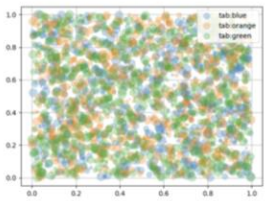
Matplotlib



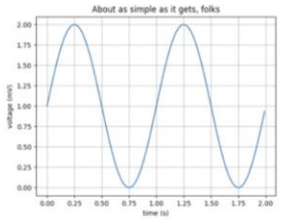
Marker examples



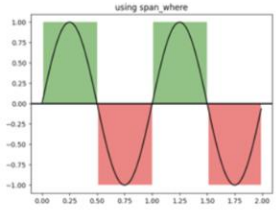
Scatter Symbol



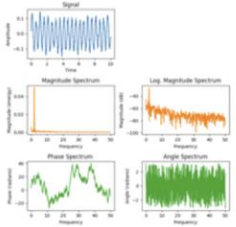
Scatter plots with a legend



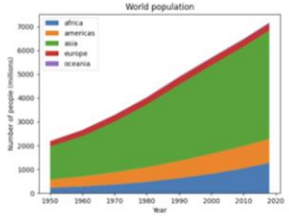
Simple Plot



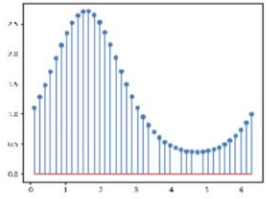
Using span_where



Spectrum Representations



Stackplots and streamgraphs



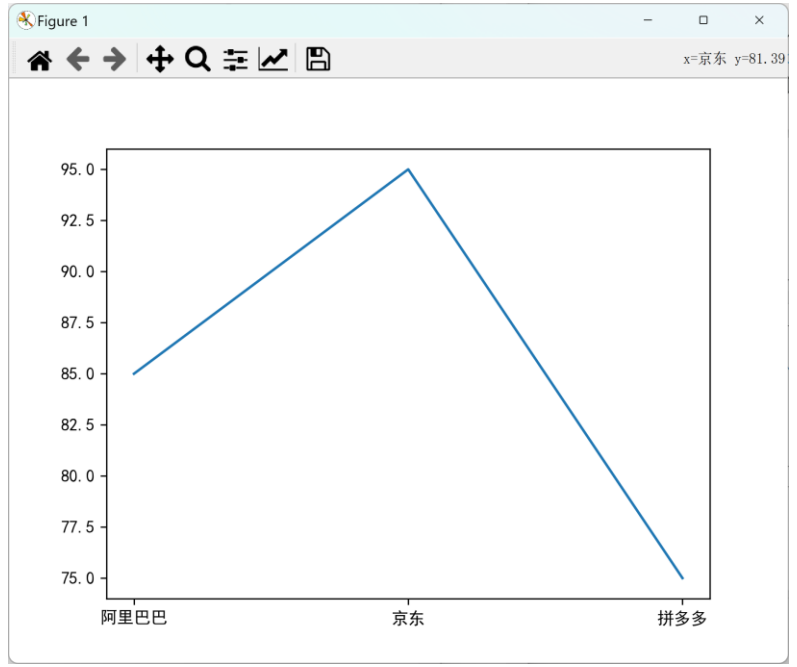
Stem Plot

Matplotlib常用函数

函数名称	函数作用
plot()	绘图折线图
show()	在本机显示图形

绘制折线图

```
import matplotlib.pyplot as plt
name=["阿里巴巴","京东","拼多多"]
grade=[85, 95, 75] #虚构数据仅为举例
plt.rcParams['font.sans-serif']=['SimHei']
plt.plot(name, grade)
plt.show()
```



常用函数及其属性

plt.figure(figsize=(w, h))：创建绘图对象，并设置宽度w和高度h

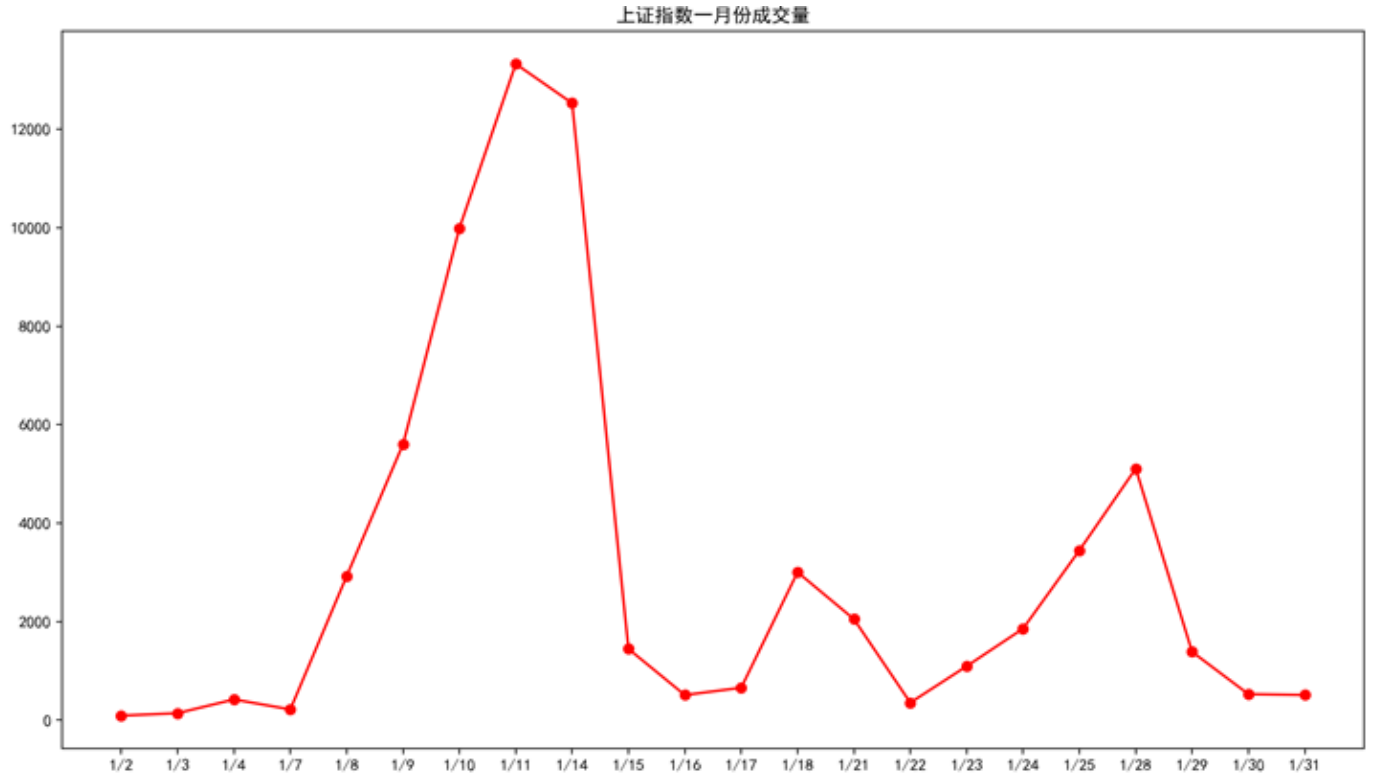
plt.title()：为图表添加标题

plt.plot()参数主要包括：

- 常见的颜色字符：'r'、'g'、'b'、'y'、'w'等
- 常见的线型字符：'-'（直线）、'--'（虚线）、':'（点线）等
- 常用的描点标记：'o'（圆圈）、's'（方块）、'^'（三角形）等

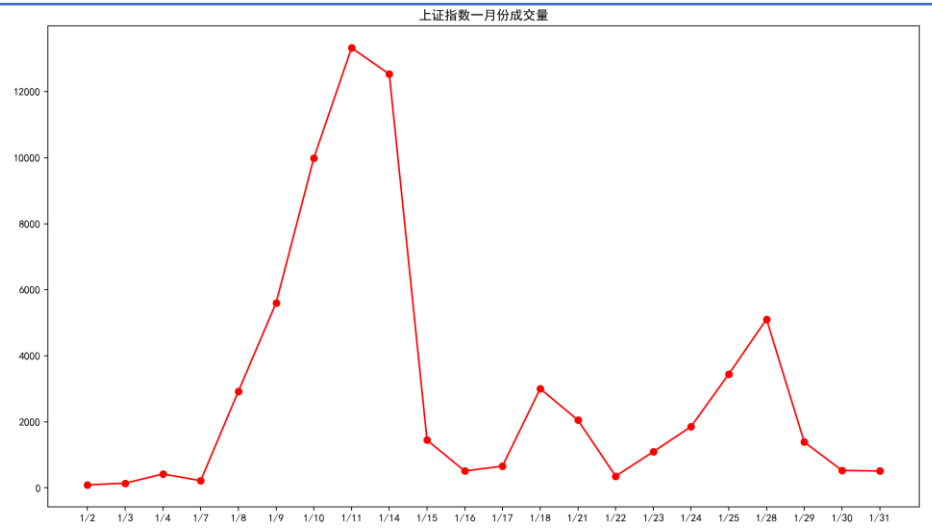
Matplotlib应用案例

编写程序：从文件中读入某股票的日期和成交量，使用matplotlib绘制出价格折线图。



Matplotlib应用案例

```
import matplotlib.pyplot as plt
date,num=[],[]
with open("上证指数1.txt","r") as fobj:
    for i in fobj:
        if i[:2]=="日期":
            continue
        i=i.strip(); info=i.split(",")
        date.append(info[0][5:]);  num.append(float(info[6]))
plt.rcParams['font.sans-serif']=['SimHei']
plt.title("上证指数一月份成交量")
plt.plot(date,num,"or-")
plt.show()
```

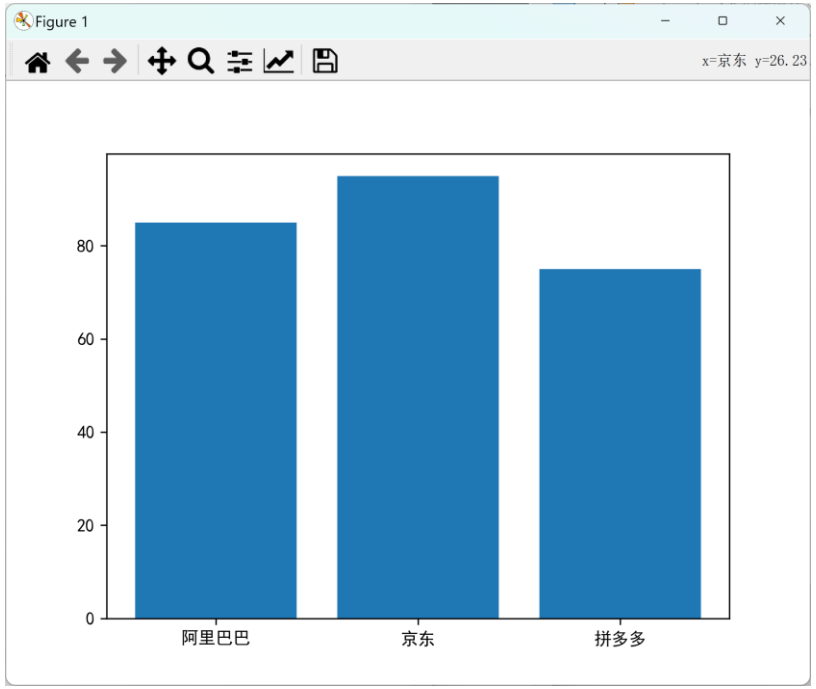


Matplotlib常用函数

函数名称	函数作用
plot()	绘图折线图
show()	在本机显示图形
bar()	绘制垂直条形图

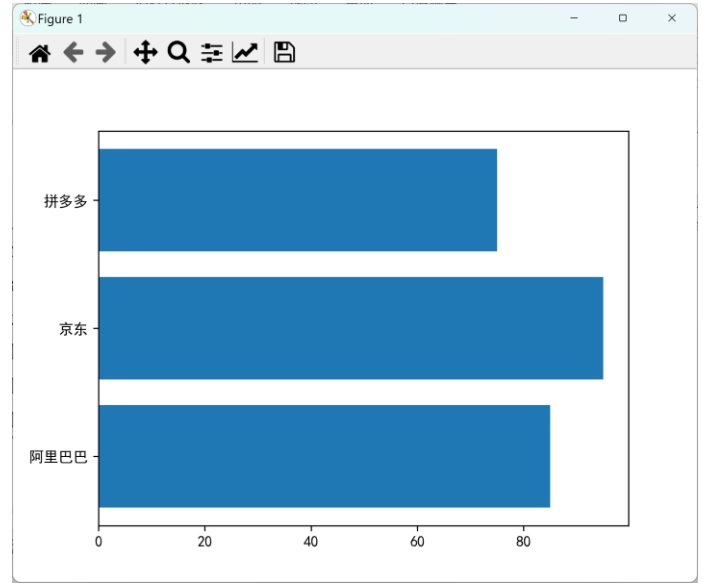
绘制垂直条形图

```
import matplotlib.pyplot as plt
name=["阿里巴巴","京东","拼多多"]
grade=[85, 95, 75]  #虚构数据仅为举例
plt.rcParams['font.sans-serif']=['SimHei']
plt.bar(name, grade)
plt.show()
```



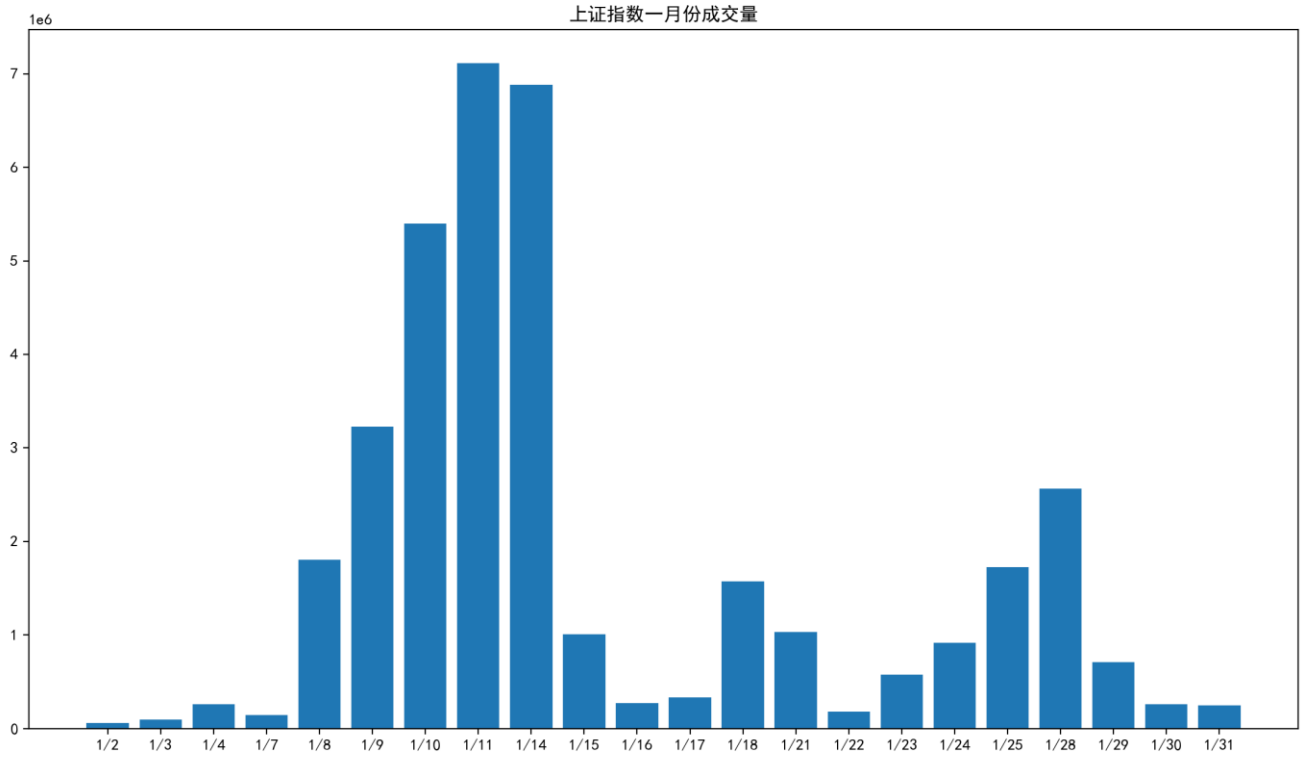
绘制水平条形图

```
import matplotlib.pyplot as plt
name=["阿里巴巴","京东","拼多多"]
grade=[85, 95, 75]  #虚构数据仅为举例
plt.rcParams['font.sans-serif']=['SimHei']
plt.barh(name, grade)
plt.show()
```



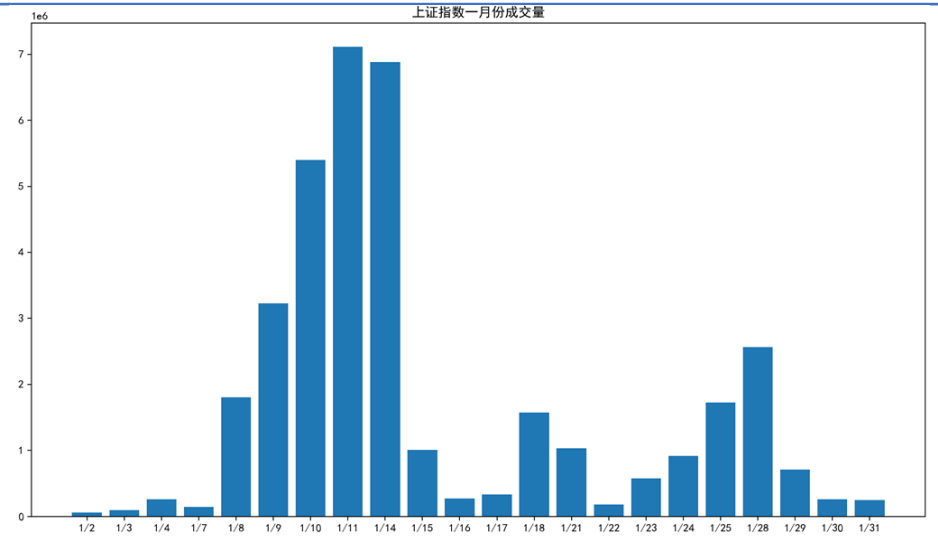
思考题

编写程序：从文件中读入某股票的日期和成交量，使用matplotlib绘制出价格条形图。



Matplotlib应用案例

```
import matplotlib.pyplot as plt
date,num=[],[]
with open("上证指数1.txt","r") as fobj:
    for i in fobj:
        if i[:2]=="日期":
            continue
        i=i.strip(); info=i.split(",")
        date.append(info[0][5:]);  num.append(float(info[7]))
plt.rcParams['font.sans-serif']=['SimHei']
plt.title("上证指数一月份成交量")
plt.bar(date,num)
plt.show()
```

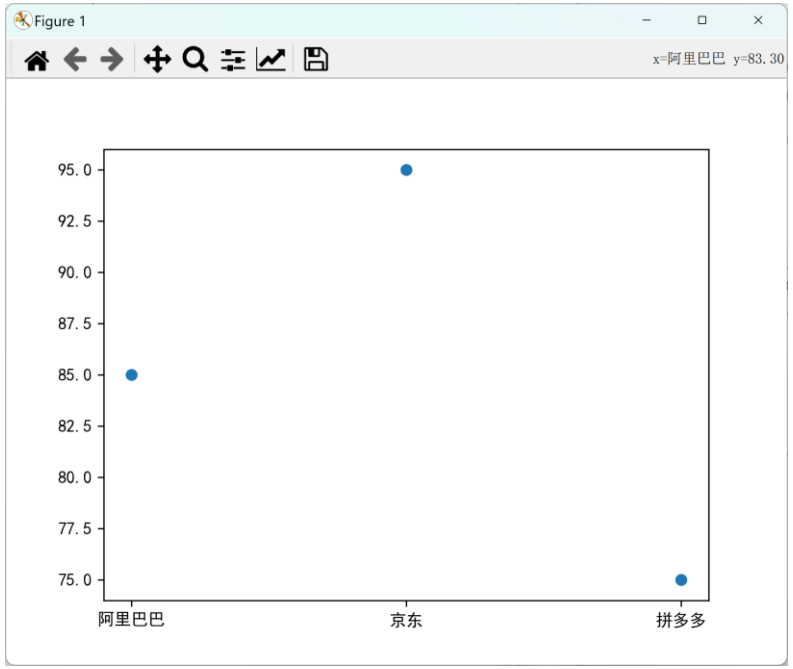


Matplotlib常用函数

函数名称	函数作用
plot()	绘图折线图
show()	在本机显示图形
bar()	绘制垂直条形图
scatter()	绘制散点图

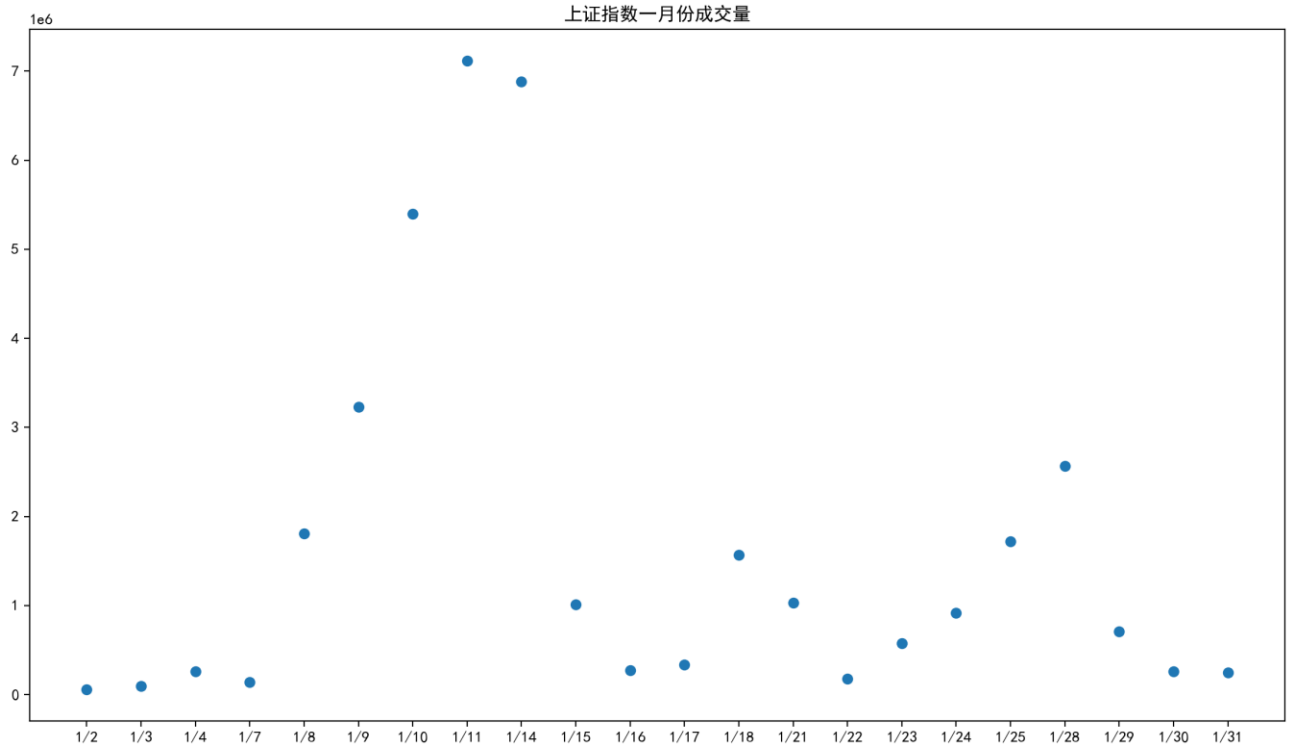
绘制散点图

```
import matplotlib.pyplot as plt
name=["阿里巴巴","京东","拼多多"]
grade=[85, 95, 75] #虚构数据仅为举例
plt.rcParams['font.sans-serif']=['SimHei']
plt.scatter(name, grade)
plt.show()
```



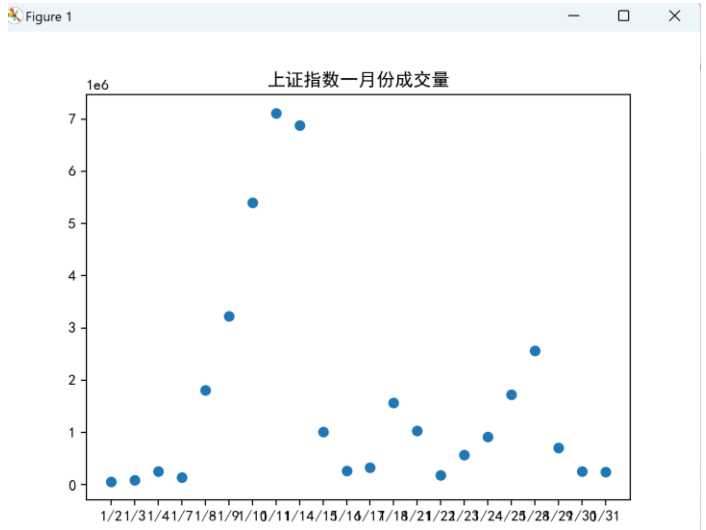
思考题

编写程序：从文件中读入某股票的日期和成交量，使用matplotlib绘制出价格散点图。



Matplotlib应用案例

```
import matplotlib.pyplot as plt
date,num=[],[]
with open("上证指数1.txt","r") as fobj:
    for i in fobj:
        if i[:2]=="日期":
            continue
        i=i.strip(); info=i.split(",")
        date.append(info[0][5:]);  num.append(float(info[7]))
plt.rcParams['font.sans-serif']=['SimHei']
plt.title("上证指数一月份成交量")
plt.scatter(date,num)
plt.show()
```

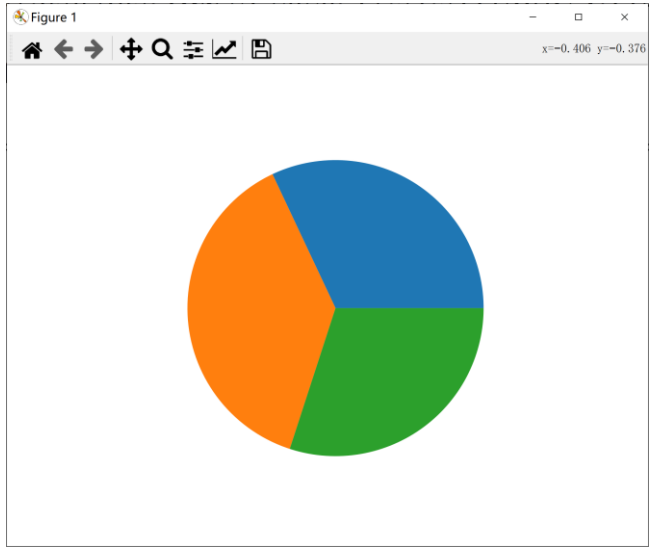


Matplotlib常用函数

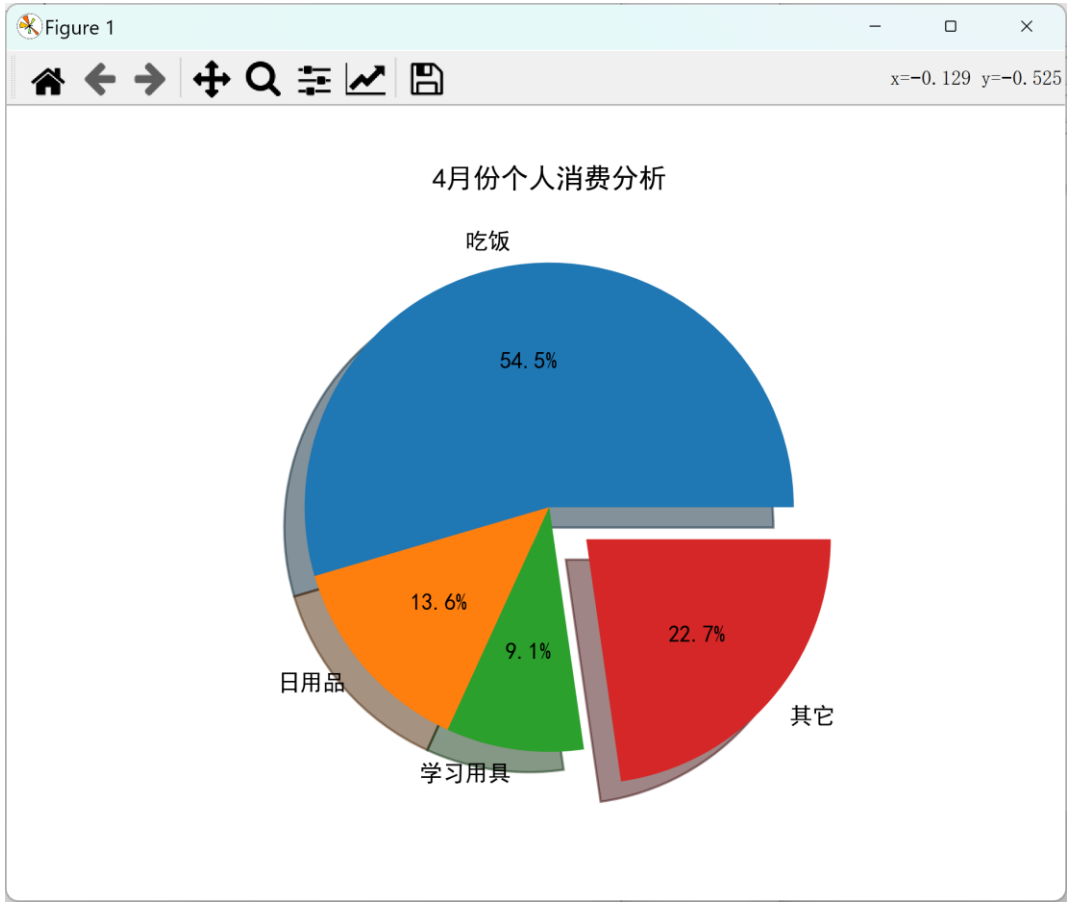
函数名称	函数作用
plot()	绘图折线图
show()	在本机显示图形
bar()	绘制垂直条形图
scatter()	绘制散点图
pie()	绘制饼图

绘制饼图

```
import matplotlib.pyplot as plt
score=[85, 95, 75]
plt.pie(score)
plt.show()
```

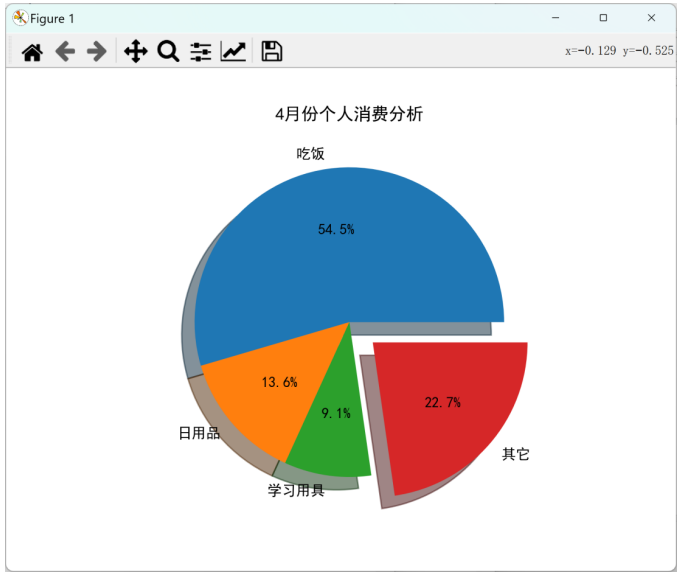


制作个人消费饼图



制作个人消费饼图

```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei']
labels = ['吃饭','日用品','学习用具','其它']
sizes = [1200,300,200,500]
explodes = (0,0,0,0.2)
plt.pie(sizes,explode=explodes,labels=labels,
 autopct='%.1f%%', shadow=True)
plt.title("4月份个人消费分析")
plt.show()
```



Matplotlib常用函数

函数名称	函数作用
plot()	绘图折线图
show()	在本机显示图形
bar()	绘制垂直条形图
scatter()	绘制散点图
pie()	绘制饼图
subplot()	绘制子图

个人消费对比分析

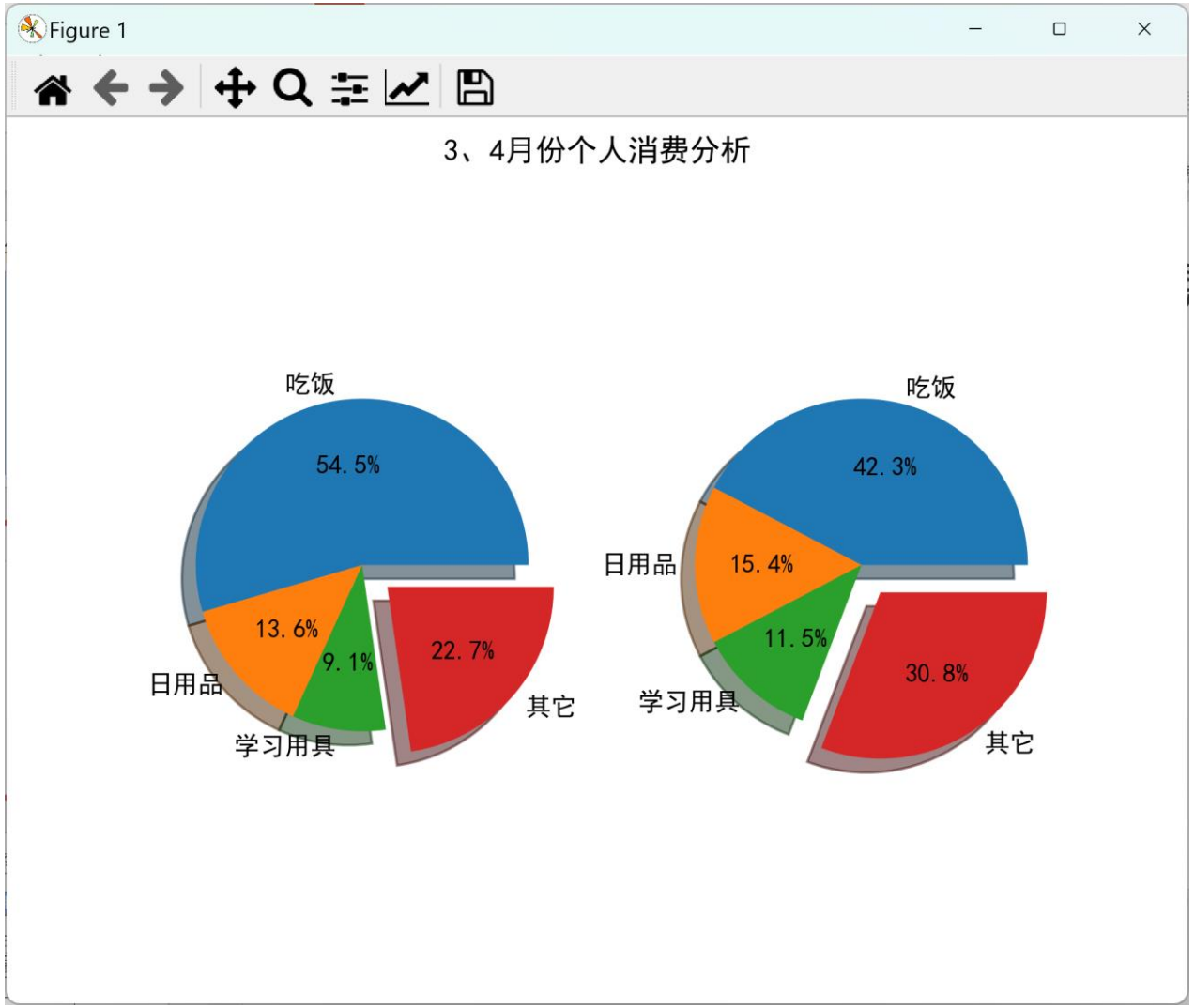
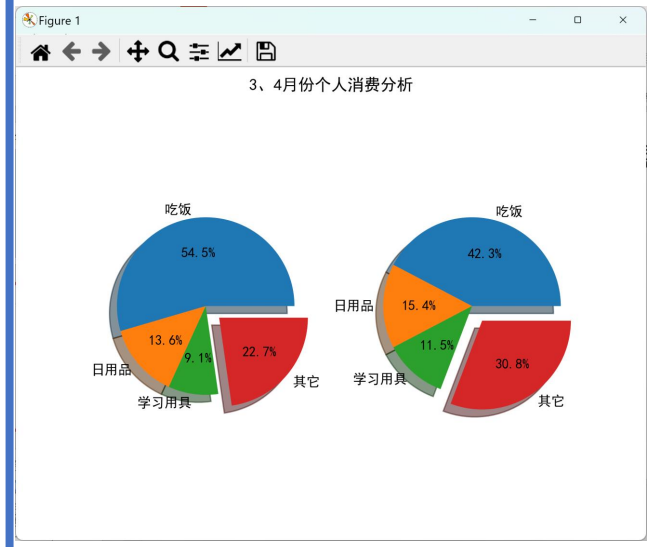


Figure 1
3、4月份个人消费分析
吃饭
54. 5%
13. 6%
9. 1%
22. 7%
日用品
学习用具
其它
吃饭
42. 3%
日用品
15. 4%
11. 5%
30. 8%
学习用具
其它

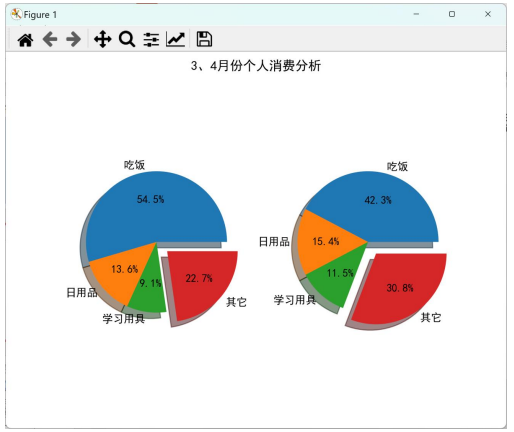
个人消费对比分析

```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei']
p1=plt.subplot(121)
p2=plt.subplot(122)
labels = ['吃饭','日用品','学习用具','其它']
sizes1 = [1200,300,200,500]
sizes2 = [1100,400,300,800]
```



个人消费对比分析

```
explodes = (0,0,0,0.2)
p1.pie(sizes1,explode=explodes,labels=labels,
 autopct='%1.1f%%', shadow=True)
p2.pie(sizes2,explode=explodes,labels=labels,
 autopct='%1.1f%%', shadow=True)
plt.suptitle("3、4月份个人消费分析")
plt.show()
```

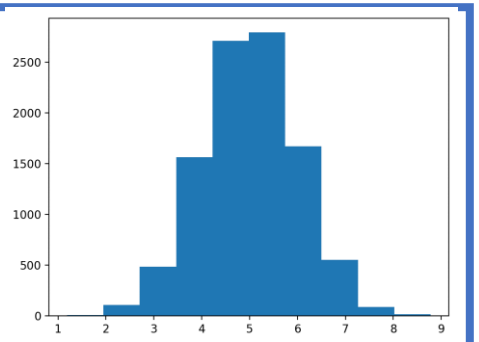


Matplotlib常用函数

函数名称	函数作用
plot()	绘图折线图
show()	在本机显示图形
bar()	绘制垂直条形图
scatter()	绘制散点图
pie()	绘制饼图
subplot()	绘制子图
hist()	绘制直方图

绘制直方图

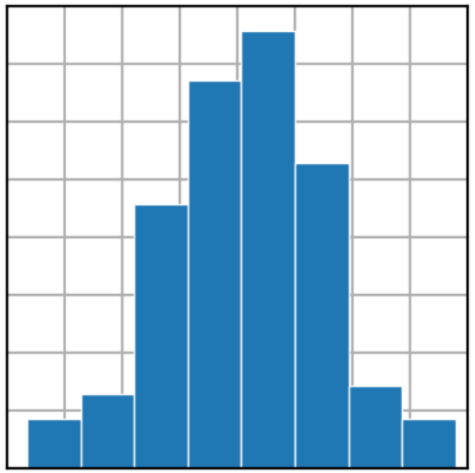
```
import matplotlib.pyplot as plt

import numpy as np

#生成10000个高斯分布随机数，均值为5，标准差为1

x=np. random. normal(loc=5, scale=1, size=10000)

plt. hist(x)

plt. show()
```



35

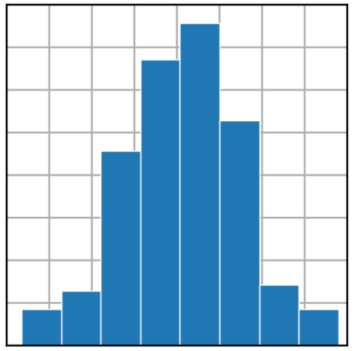
直方图

直方图(Histogram)：又称质量分布图，是一种统计报告图，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类型，纵轴表示分布情况。



直方图

构建直方图：第一步是将值的范围分段，即将整个值的范围分成一系列间隔，然后计算每个间隔中有多少值。直方图是用面积表示各组频数的多少，矩形的高度表示每一组的频数或频率，宽度则表示各组的组距。



直方图

plt.hist(x, bins=10, range=None, normed=False, …)

x： 指定要绘制直方图的数据

bins： 指定直方图条形的个数

range： 指定直方图数据的上下界

normed： 是否将直方图的频数转换成频率

Numpy

NumPy(Numerical Python的缩写)：是一个开源的Python科学计算库，NumPy数组在数值运算方面的效率优于列表。它是数据分析、机器学习和科学计算的主力军。

官网：**https://numpy.org/doc/stable/**

创建Numpy数组

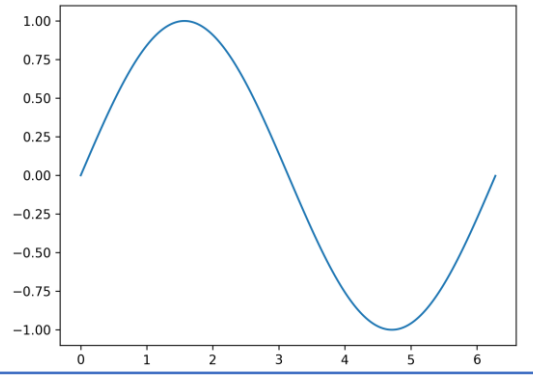
```
>>> import numpy as np          #一般以np作为别名
>>> score=np.array([80,91,78])      # 创建一维数组
>>> print(score+5)
>>> b = np.array([[10,5],[30,6]])    # 创建二维数组
>>> print(b*b)
```

Numpy重要函数

```
>>> import numpy as np
>>> a = np. arange(0,10, 0.1)          #[0, 10)，步长为0.1
>>> b = np. linspace(0,10,100)         #[0,10]，分成100份
>>> c=a. reshape(20,5)                 #变为20行5列
>>> result=a. reshape(-1,1)            #变成1列
>>> test=result. flatten()  #返回一个折叠成一维的数组
```

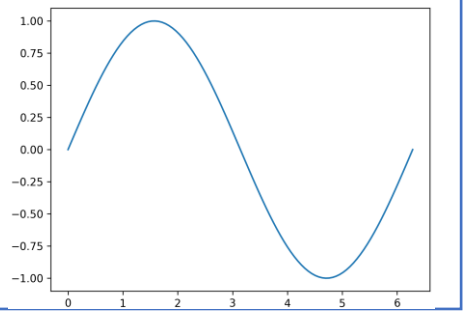
Numpy绘制函数图

```
import matplotlib.pyplot as plt
import numpy as np
x=np.arange(0,2*np.pi,0.01)  #x从0到2π，步长0.01
y=np.sin(x)
plt.plot(x,y)
plt.show()
```



Numpy绘制函数图

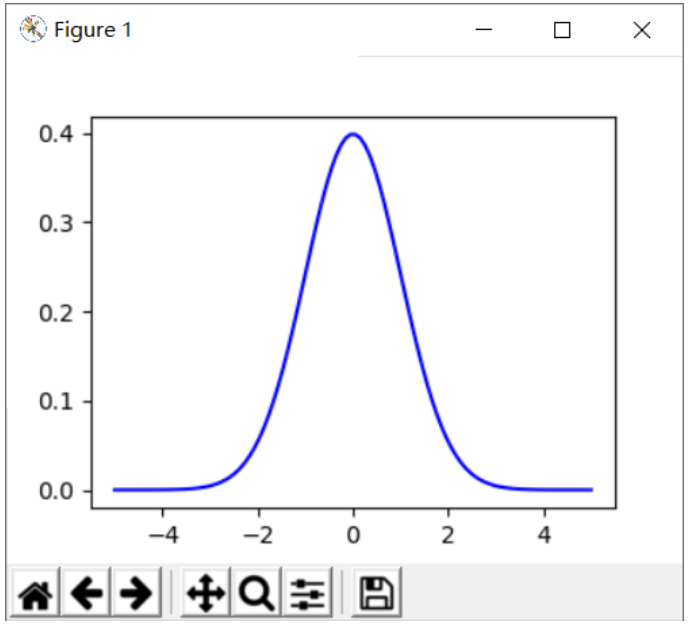
```
import matplotlib.pyplot as plt
import numpy as np
x=np.linspace(0,2*np.pi,100)  #x从0到2π分成100份
y=np.sin(x)
plt.plot(x,y)
plt.show()
```



思考题

编写程序，绘制正态分布的密度函数：$f(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

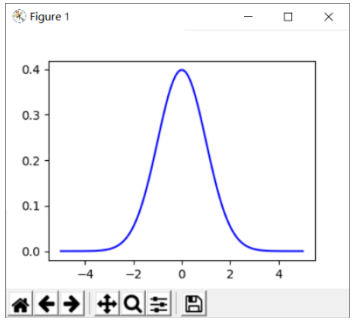
其中： $\mu=0,\sigma=1$ $x\in[-5,5]$



44

正态分布密度函数

```
import matplotlib.pyplot as plt
from numpy import *
plt.figure(figsize=(4,3))
x=linspace(-5,5,100)   #x从-5到5分成100份
y=(1/(sqrt(2*pi)))*exp(-(x*x)/2)
plt. plot(x,y,'-b')
plt. show()
```



Numpy元素取值

```
>>> import numpy as np
>>> a = np. arange(10). reshape(2,5)
>>> a[0]                                  #打印第1行
>>> a[1][2]或者a[1, 2]                    #打印第2行第3列
>>> a[: , 1]                              #打印第2列
>>> a[: , [1,3]]                          #打印第2、4列
```

随机整数

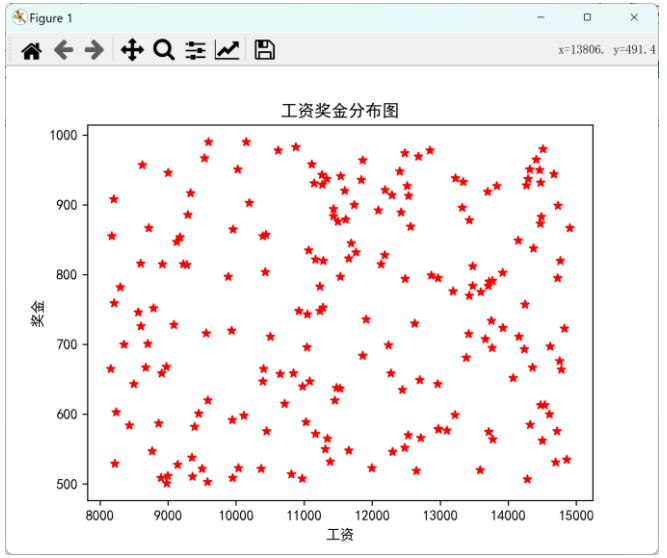
numpy.random. randint(low, high, size, dtype=int)：返回范围为[low, high)随机整数，size为数组尺寸

```
>>> import numpy as np
>>> one=np. random. randint(2) # 产生1个[0,2)之间随机整数
>>> grade=np. random. randint(1,5,size=10) # 产生10个[1,5)之间随机整数
>>> salary=np. random. randint(2000,3000,size=(2,4)) #2行4列
```

工资奖金散点图

```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.family']=['SimHei']
salary=np. random. randint(8000,15000,size=200)
bonus=np. random. randint(500,1000,size=200)
plt.scatter(salary,bonus,c="r",marker="*")
plt.xlabel("工资")
plt.ylabel("奖金")
plt.title('工资奖金分布图')
plt.show()
```

如何产生浮点数工资及奖金?



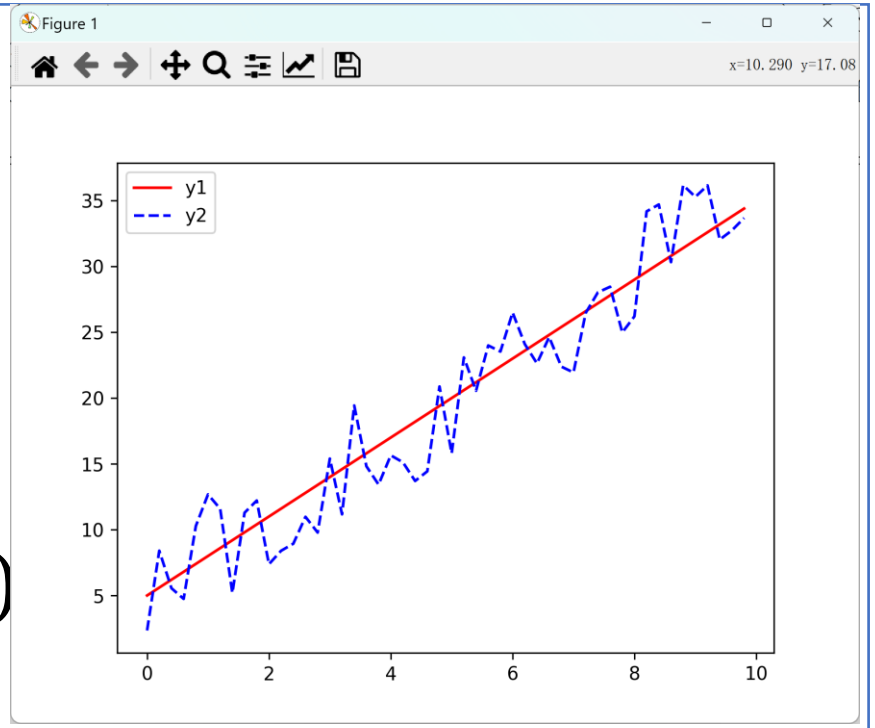
随机浮点数

numpy.random.uniform(low,high,size)： 从一个均匀分布[low,high)中随机采样，size为样本数目

```
>>> import numpy as np
>>> test=np. random. uniform()   # 产生1个[0,1)之间随机浮点数
>>> score= np. random. uniform(0, 100, size=3) #产生 3个0-99的随机浮点数
>>> s= np. random. uniform(200,300,size=(2 ,4)) #产生2行4列200-299的浮点数
```

案例分析

```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(0,10,0.2)
y1=3*x+5; y2=[]
for i in y1:
    y2.append(i+np.random.uniform(-5,5))
plt.plot(x,y1,"r-",label='y1')
plt.plot(x,y2,"b--",label='y2')
plt.legend(loc='upper left')
plt.show()
```



如何将数据存入文件中？

谢　谢