# Python与金融数据挖掘(10)

文欣秀

wenxinxiu@ecust.edu.cn

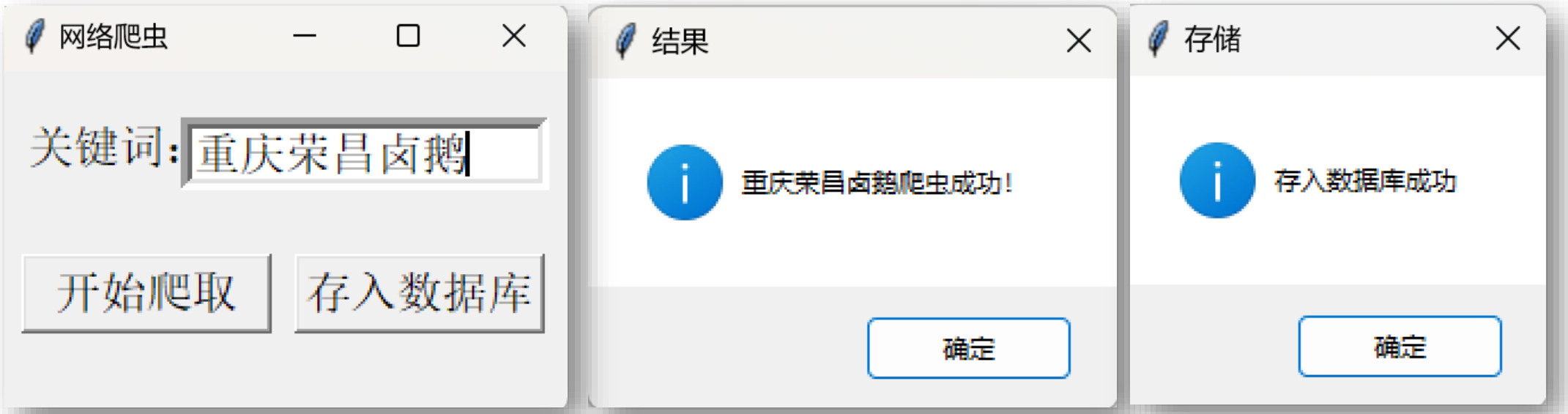
# 舆情数据评分系统搭建

- 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

# 重庆荣昌卤鹅事件

“卤鹅哥”林江是重庆荣昌的自媒体博主，也是荣昌卤鹅的传承者。2025 年 3 月 30 日，林江在个人抖音号发布视频称要请美国顶流自媒体博主 “甲亢哥” 吃荣昌非遗特色美食荣昌卤鹅。3 月 31 日，“卤鹅哥” 在成都街头首次向 “甲亢哥” 投喂荣昌卤鹅，此后他又辗转重庆、香港、深圳、长沙等城市，继续向 “甲亢哥” 投喂卤鹅。这一行为引发了众多网友关注，让荣昌卤鹅在网络上迅速走红，相关视频让荣昌卤鹅的网络曝光率暴增 4050%。

# 重庆荣昌卤鹅爬虫

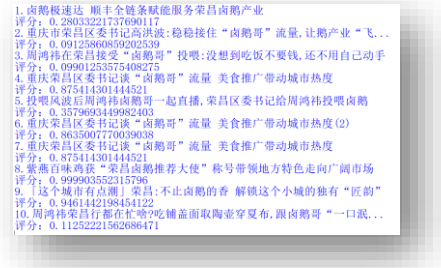


# 程序代码（一）

```
from tkinter import *
from tkinter.messagebox import *
import requests
import re
from snownlp import SnowNLP
import pymysql
title=[]
score=[]
```

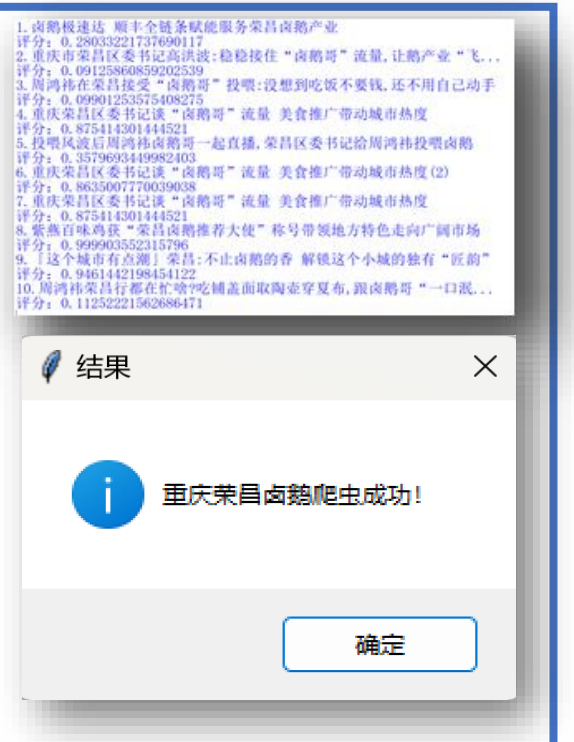
# 程序代码（二）

```
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 title,score
        company=E1.get()
        url = 'http://www.baidu.com/s?tn=news&rtt=1&wd=' + company
        res = requests.get(url, headers=headers).text
        p_title = '<h3 class="news-title_1YtI1 ">.*?>(.*?)</a>'
        title = re.findall(p_title, res, re.S)
        #下一页
```



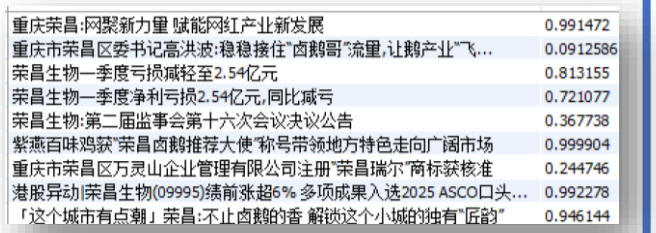
# 程序代码（三）

```
        #...
        for i in range(len(title)):
            title[i] = title[i].strip()
            title[i] = re.sub('<.*?>', '', title[i])
            print(str(i + 1) + '.' + title[i])
            s = SnowNLP(title[i])
            score. append(s. sentiments)
            print(f"评分：{score}")
        showinfo("结果","{}".format(company+'爬虫成功！'))
    except:
        showinfo("结果",{}.format(company+'爬虫失败！'))
```



# 程序代码（四）

```
def save():
    global title,href
    try:
        conn = pymysql.connect(host="localhost", user="root",
        password="123456", database="test")
        cur = conn.cursor()
        cur.execute("""DROP TABLE IF EXISTS result""")
        sql = """CREATE TABLE result (title  CHAR(100),score float)"""
        cur.execute(sql)
        conn.commit()
        conn.close()
        #...
```



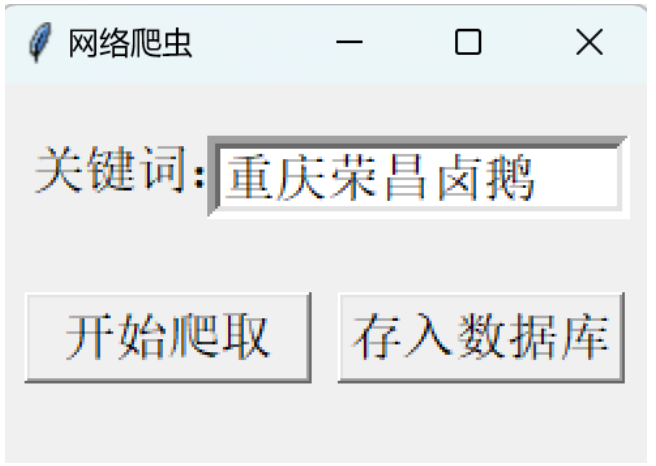
| | |
|---|---|
| 重庆荣昌:网聚新力量 赋能网红产业新发展 | 0.991472 |
| 重庆市荣昌区委书记高洪波:稳稳接住"卤鹅哥"流量,让鹅产业"飞... | 0.0912586 |
| 荣昌生物一季度亏损减轻至2.54亿元 | 0.813155 |
| 荣昌生物一季度净利亏损2.54亿元,同比减亏 | 0.721077 |
| 荣昌生物:第二届监事会第十六次会议决议公告 | 0.367738 |
| 紫燕百味鸡获"荣昌卤鹅推荐大使"称号带领地方特色走向广阔市场 | 0.999904 |
| 重庆市荣昌区万灵山企业管理有限公司注册"荣昌瑞尔"商标获核准 | 0.244746 |
| 港股异动|荣昌生物(09995)绩前涨超6% 多项成果入选2025 ASCO口头... | 0.992278 |
| 「这个城市有点潮」荣昌:不止卤鹅的香 解锁这个小城的独有"匠韵" | 0.946144 |

# 程序代码（五）

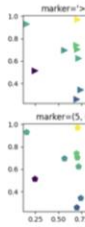
```
        # 将数据存入数据库
        conn = pymysql.connect(host='localhost', port=3306, user='root',
password='123456', database='test')
        cur = conn. cursor()
        for i in range(len(title)):
            sql = '''INSERT INTO result(title,score) VALUES (%s,%s) '''
        cur. execute(sql, (title[i],score[i]))
        conn. commit()
        cur. close()
        conn. close()
        showinfo("存储","存入数据库成功")
    except:
        showinfo("存储","存入数据库失败")
```

# 程序代码（六）

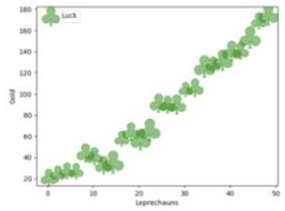
```
root = Tk()
root.title("网络爬虫")
root.geometry("250x150")
L1 = Label(root, text="关键词：",font=20)
L1.place(x=10,y=20)
E1 = Entry(root, bd =5,font=20,width=15)
E1.place(x=80,y=20)
B1 = Button(root, text="开始爬取
",font=20,width=10,command=crawler)
B1.place(x=10,y=80)
B2 = Button(root, text="存入数据库
",font=20,width=10,command=save)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



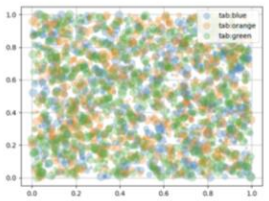
# Matplotlib



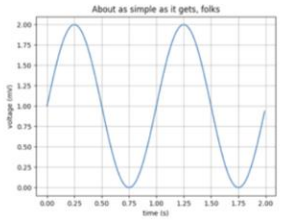

Marker examples



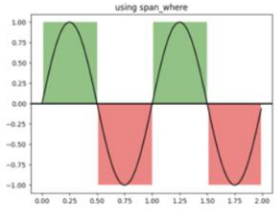

Scatter Symbol



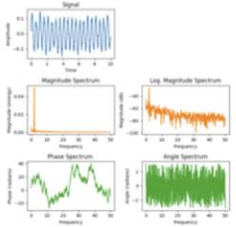

Scatter plots with a legend



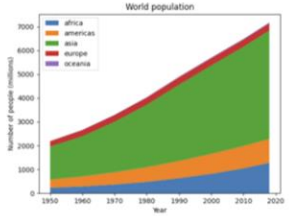

Simple Plot



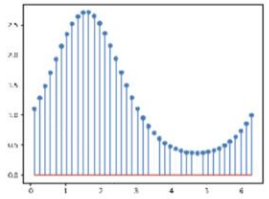

Using span_where




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 |
| | |
| | |
| | |
| | |
| | |

# 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

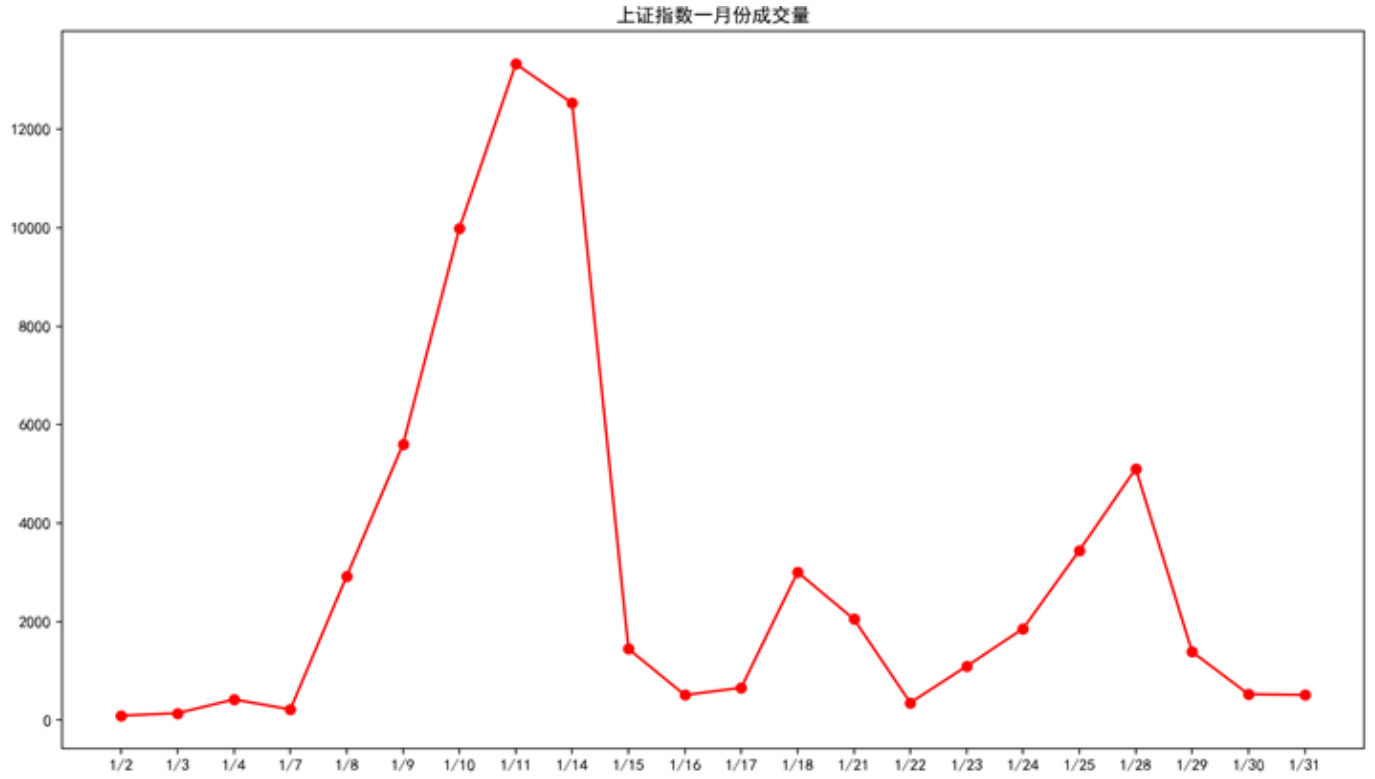
**plt.title()：**为图表添加标题

**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

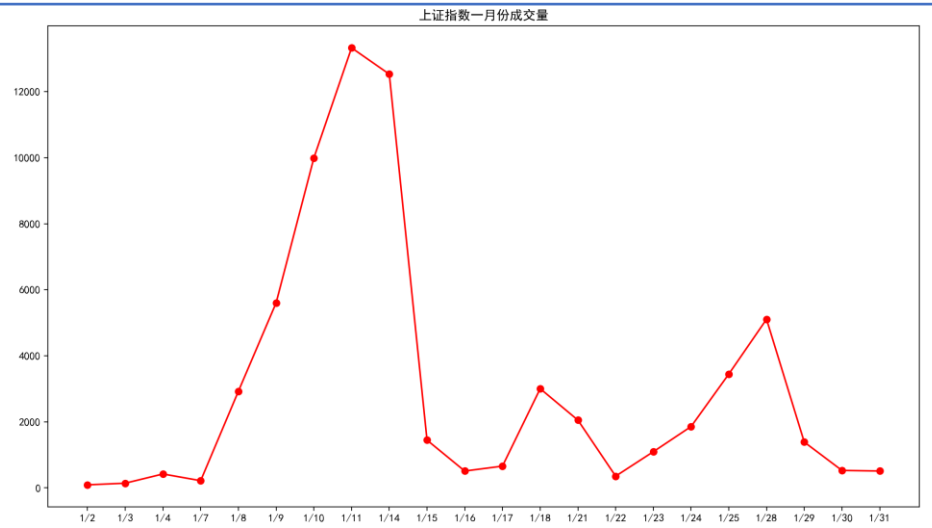
# Matplotlib应用案例

**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()
```

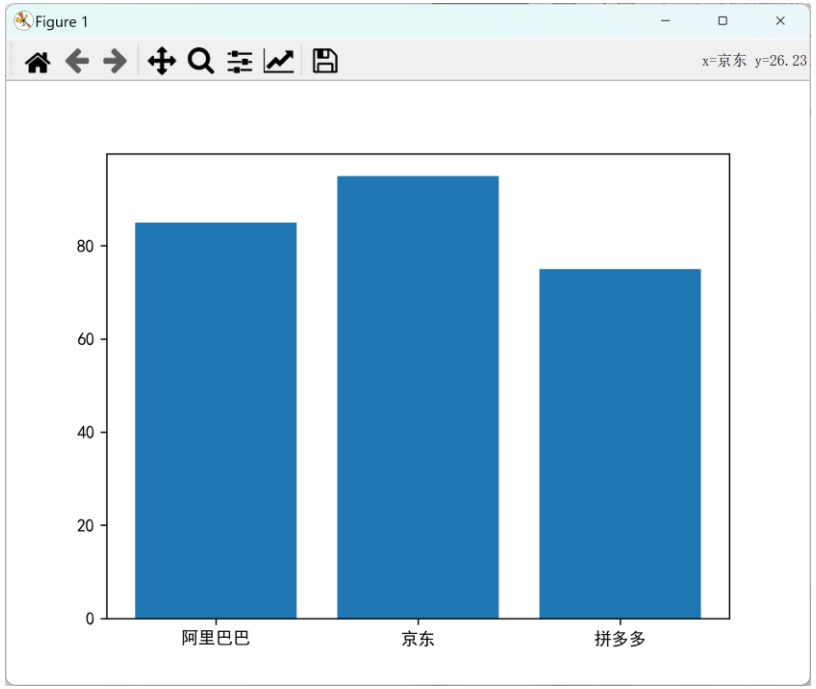


# Matplotlib常用函数

| 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 |
| | |
| | |
| | |
| | |

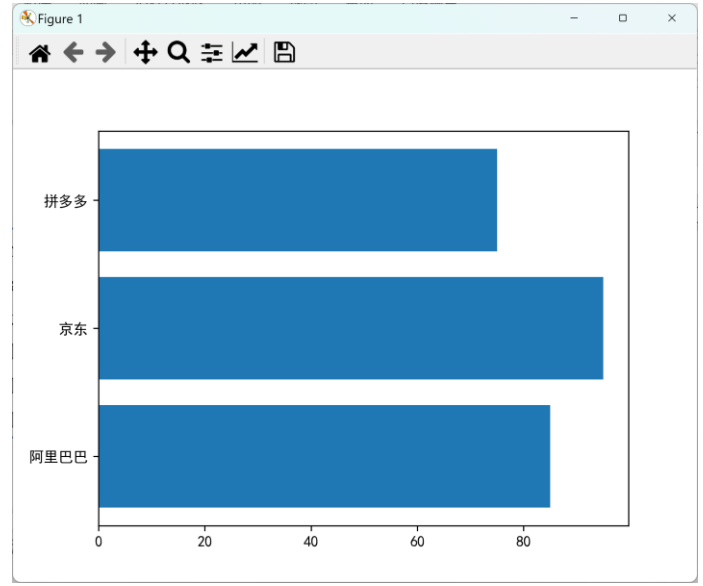
# 绘制垂直条形图

```
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()
```



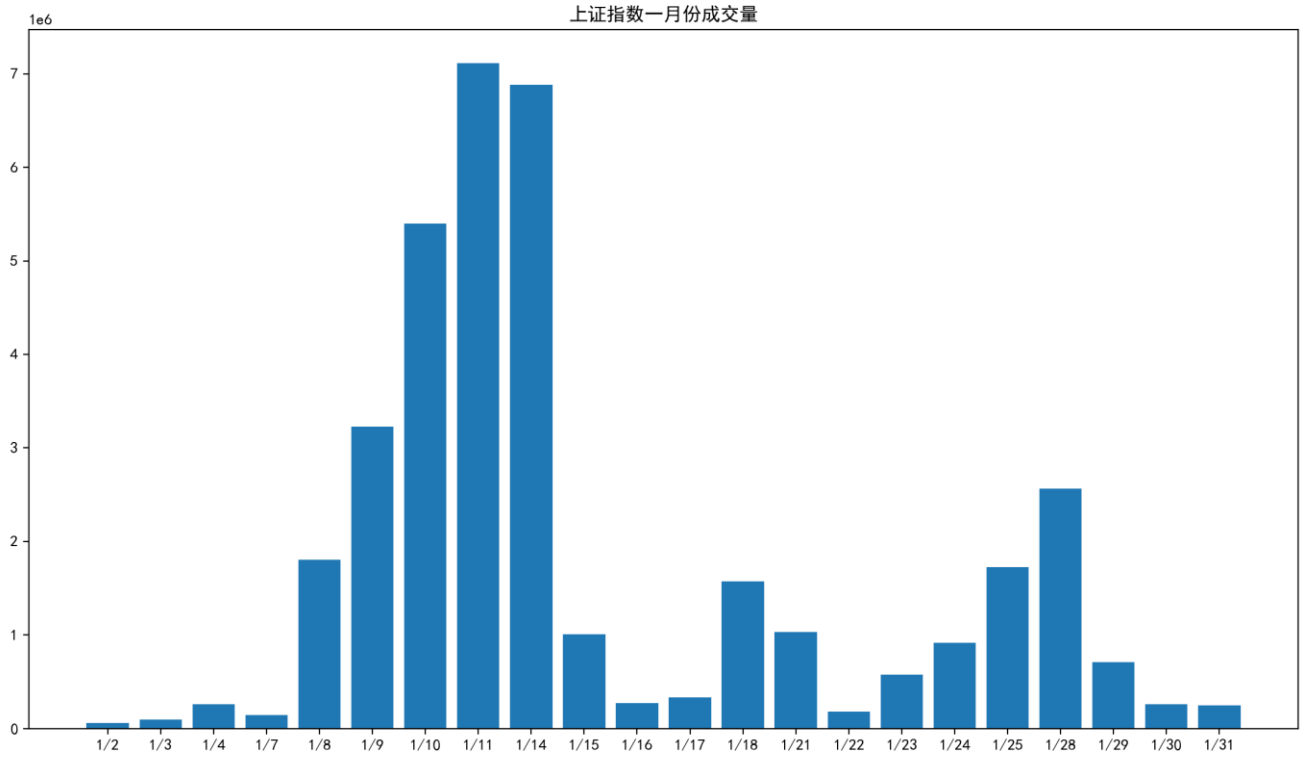
# 绘制水平条形图

```
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()
```



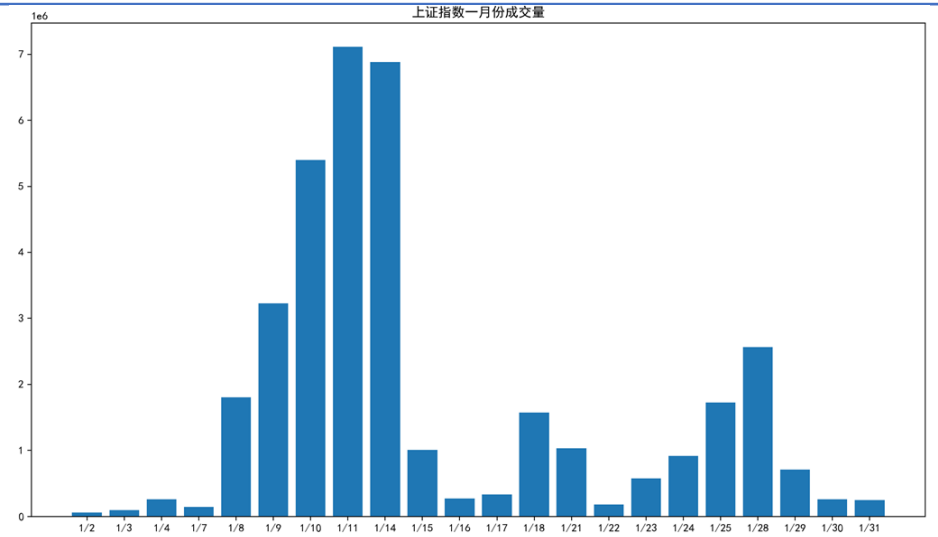
# 思考题

**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()
```

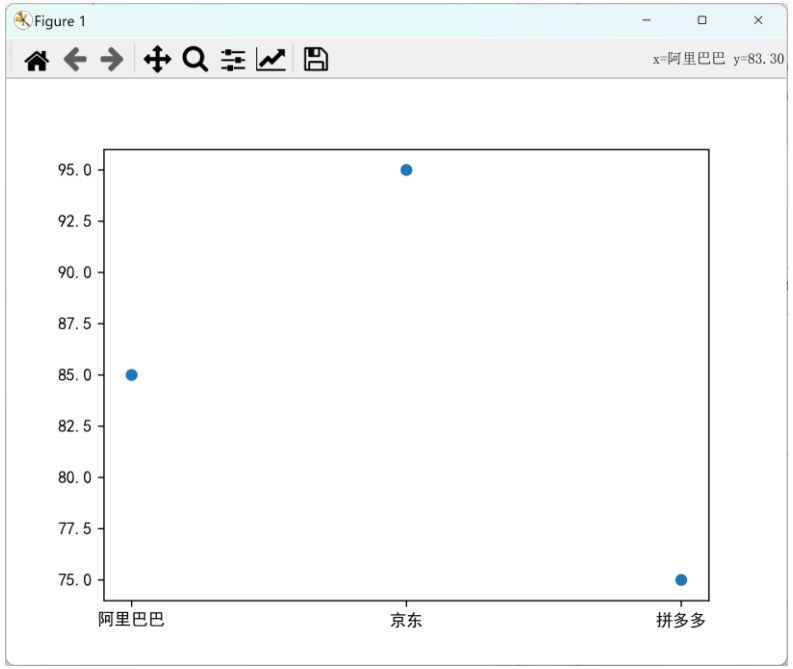


# Matplotlib常用函数

| 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 |
| | |
| | |
| | |

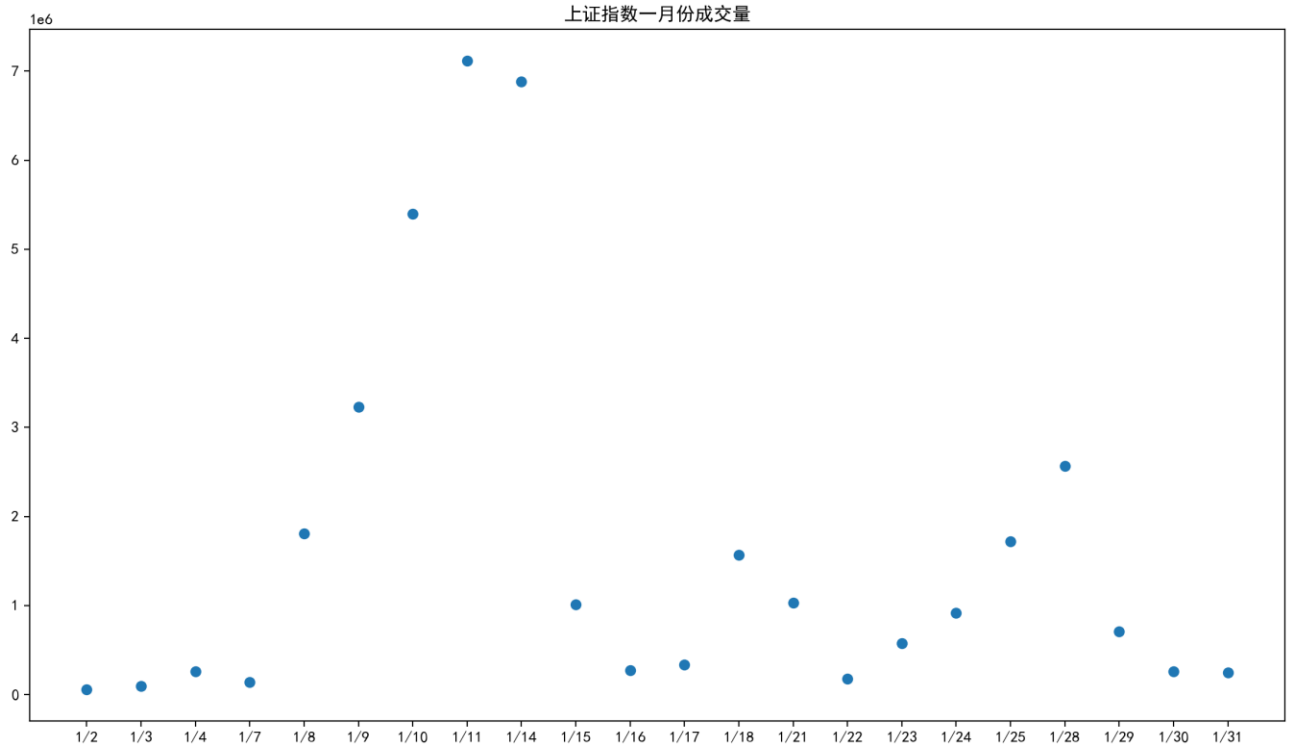
# 绘制散点图

```
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()
```



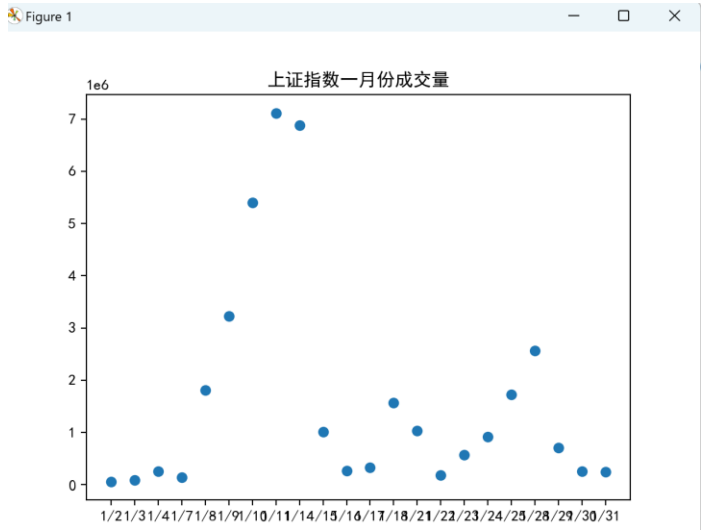
# 思考题

**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()
```

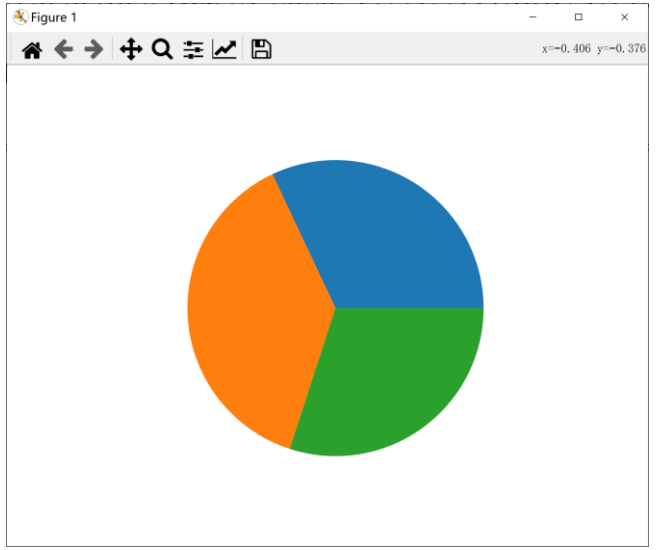


# Matplotlib常用函数

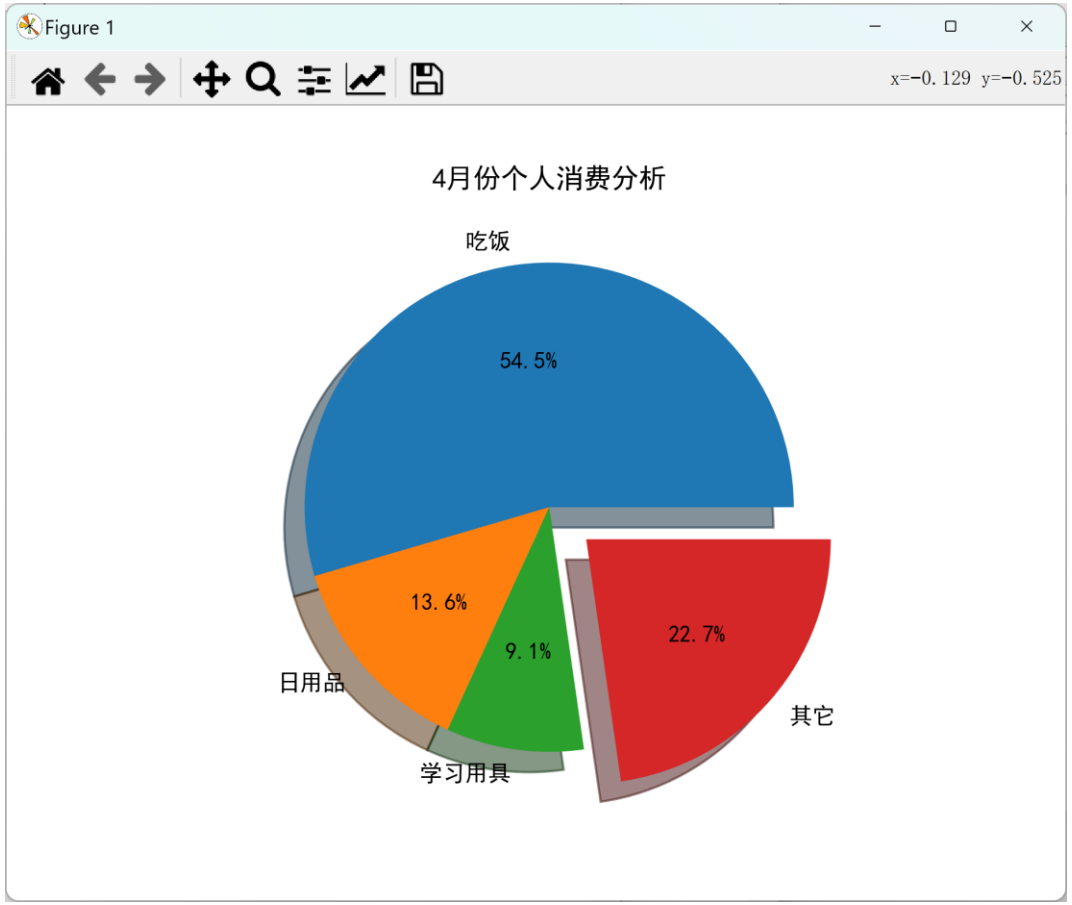
| 函数名称 | 函数作用 |
| --- | --- |
| plot() | 绘图折线图 |
| show() | 在本机显示图形 |
| bar() | 绘制垂直条形图 |
| scatter() | 绘制散点图 |
| **pie()** | 绘制饼图 |
| | |
| | |

# 绘制饼图

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

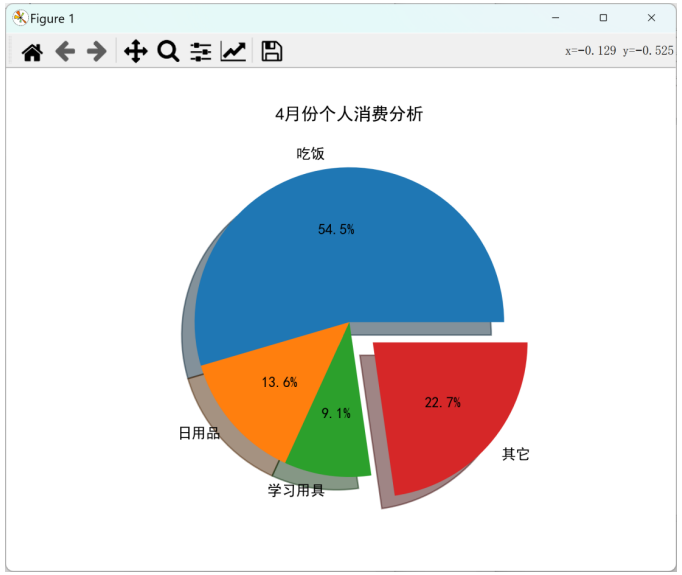


# 制作个人消费饼图



# 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

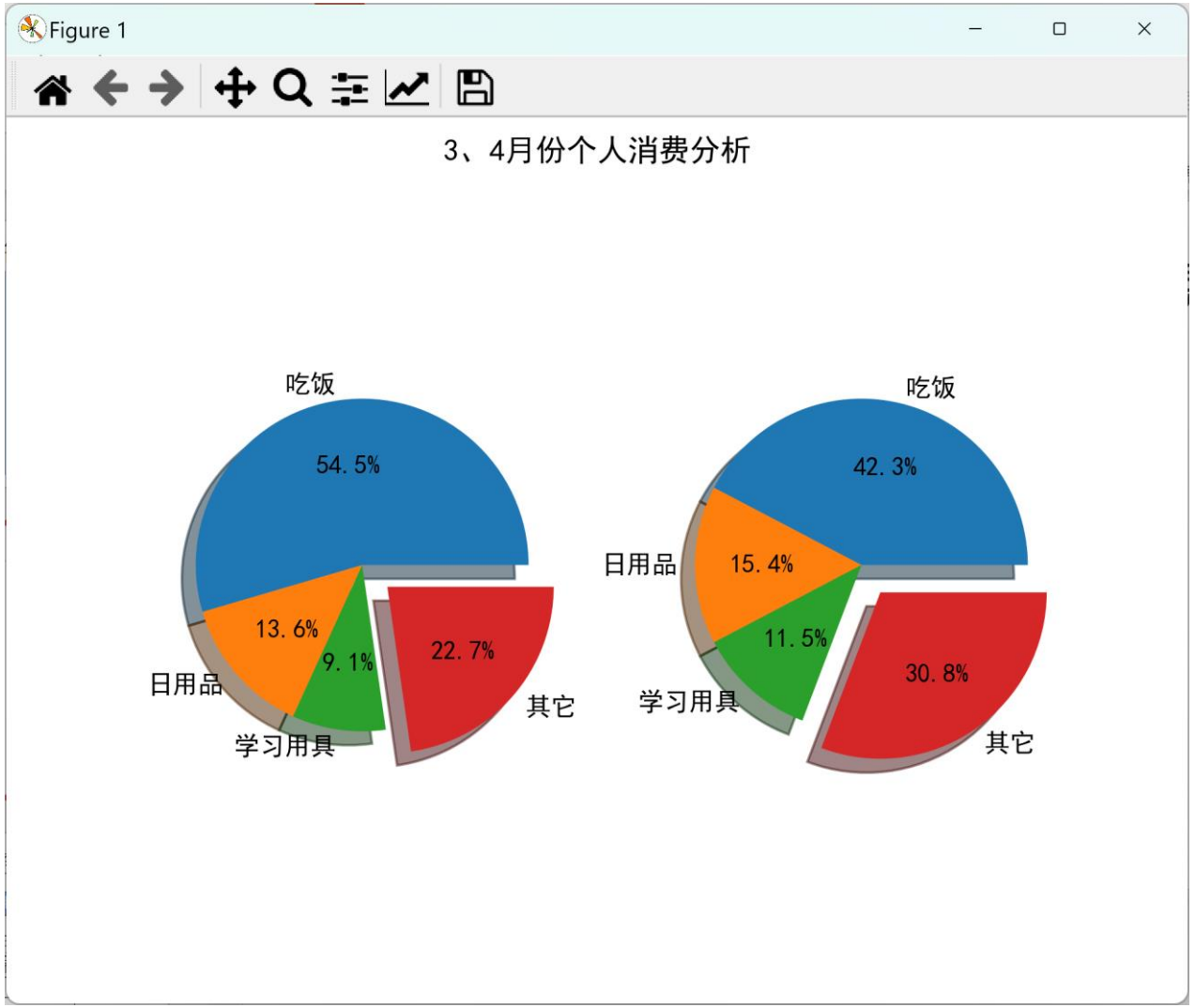
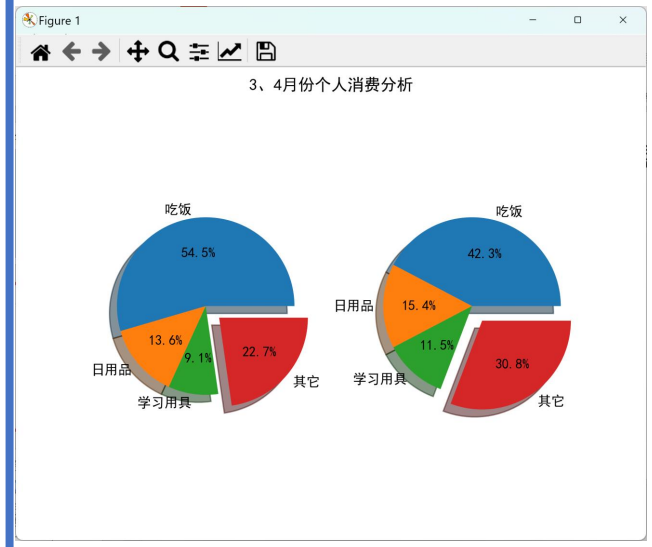


Figure 1
3、4月份个人消费分析
吃饭
54. 5%
13. 6%
9. 1%
22. 7%
日用品
学习用具
其它
吃饭
42. 3%
日用品
15. 4%
11. 5%
30. 8%
学习用具
其它

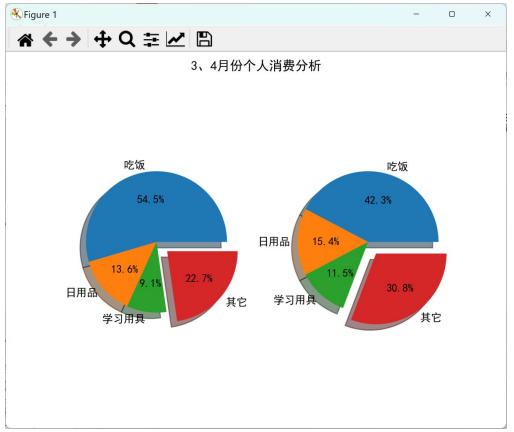
# 个人消费对比分析

```
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()
```



# 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

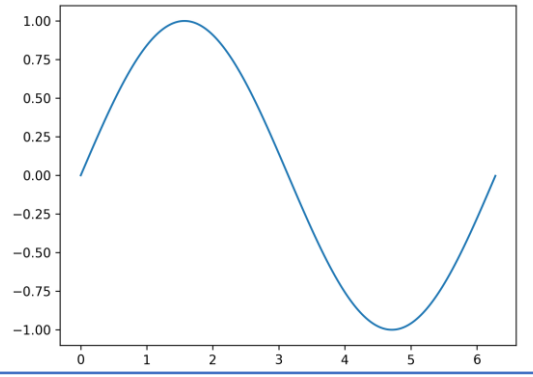
```
>>> import numpy as np          #一般以np作为别名
>>> score=np.array([80,91,78])      # 创建一维数组
>>> print(score+5)
>>> b = np.array([[10,5],[30,6]])    # 创建二维数组
>>> print(b*b)
```

# 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
```

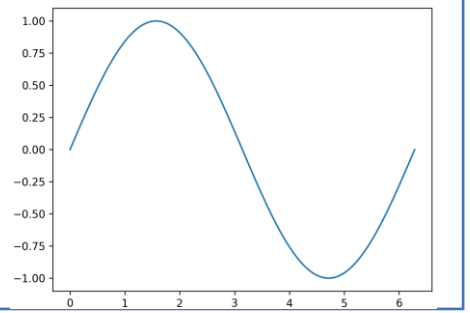
# Numpy绘制函数图

```
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

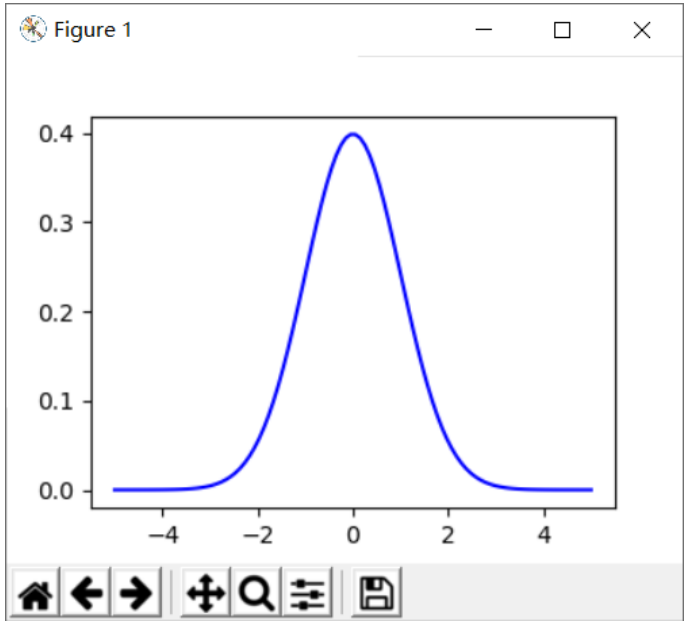
```
import matplotlib.pyplot as plt
import numpy as np
x=np.linspace(0,2*np.pi,100)  #x从0到2π分成100份
y=np.sin(x)
plt.plot(x,y)
plt.show()
```



# 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： $\mu=0,\sigma=1$ $x\in[-5,5]$

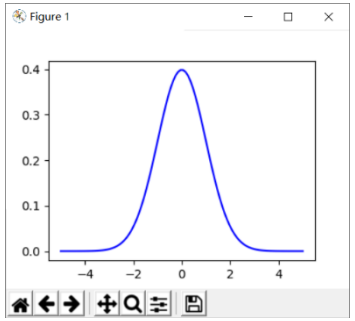




40

# 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
```

# 课堂练习

**若temp=np. arange(0,20).reshape(5,4)，则temp[3,2]的值为（ ）。**

A、12

B、 13

C、 14

D 、 15

# 随机整数

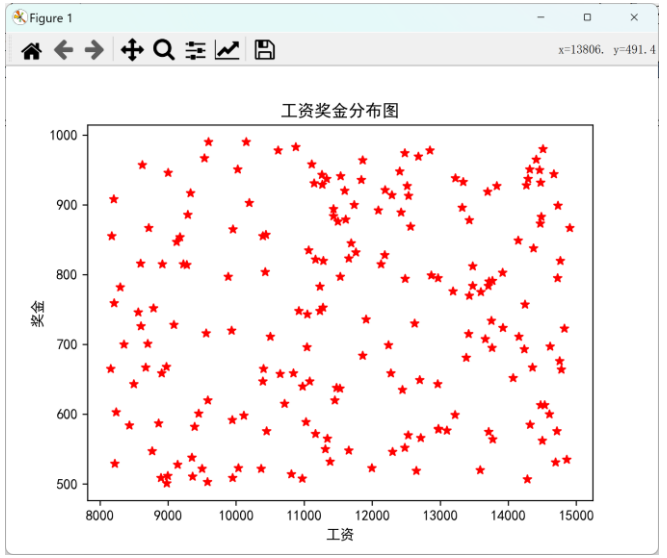
**numpy.random. randint(low, high, size, dtype=int)**：返回范围为[low, high)随机整数，size为数组尺寸

```
>>> 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？



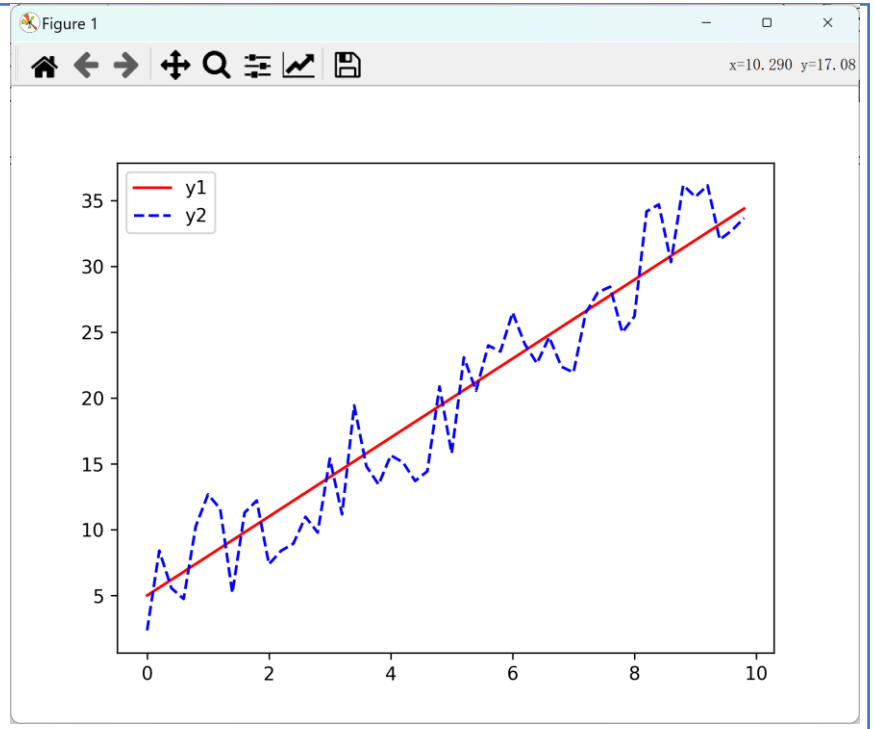
# 随机浮点数

**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=y1+np.random.uniform(-5,5,size=50)
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？

# Numpy数据存储

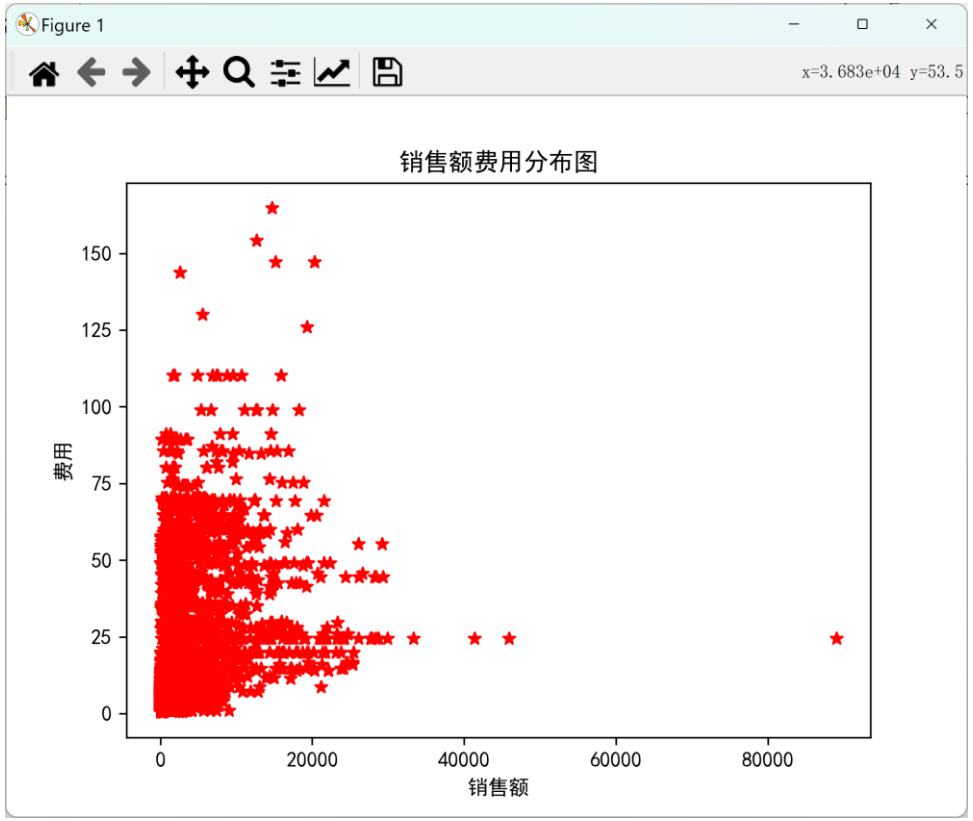
| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 5.6 | 6.2 | 6.8 | 7.4 | 8 | 8.6 | 9.2 | 9.8 | 10.4 |
| 2 | 5.2 | 9.3 | 5.4 | 10.2 | 2.5 | 12.3 | 9 | 12 | 9.4 | 12.1 |

```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(0,10,0.2)
y1=3*x+5
y2=y1+np.random.uniform(-5,5,size=50)
c=[y1,y2]
np.savetxt("result.csv",c,fmt='%.1f',delimiter=',', newline='\n')
```

# 思　　考

## 如何从文件中读取销售额和费用并绘制图形？

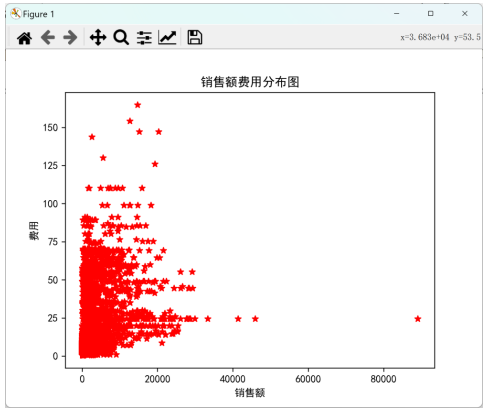
| | A | B |
|---|---|---|
| 1 | 261.54 | 35 |
| 2 | 6 | 2.56 |
| 3 | 2808.08 | 5.81 |
| 4 | 1761.4 | 89.3 |
| 5 | 160.2335 | 5.03 |
| 6 | 140.56 | 8.99 |
| 7 | 288.56 | 2.25 |
| 8 | 1892.848 | 8.99 |
| 9 | 2484.7455 | 4.2 |
| 10 | 3812.73 | 1.99 |
| 11 | 108.15 | 0.7 |
| 12 | 1186.06 | 3.92 |
| 13 | 51.53 | 0.7 |
| 14 | 90.05 | 2.58 |
| 15 | 7804.53 | 5.99 |



# 销售额与费用散点图

```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.family']=['SimHei']
result=np.loadtxt("trade.csv",delimiter=",")
money=result[:,0]
cost=result[:,1]
plt.scatter(money,cost,c="r",marker="*")
plt.xlabel("销售额")
plt.ylabel("费用")
plt.title('销售额费用分布图')
plt.show()
```

| | A | B |
|---|---|---|
| 1 | 261.54 | 35 |
| 2 | 6 | 2.56 |
| 3 | 2808.08 | 5.81 |
| 4 | 1761.4 | 89.3 |
| 5 | 160.2335 | 5.03 |
| 6 | 140.56 | 8.99 |
| 7 | 288.56 | 2.25 |
| 8 | 1892.848 | 8.99 |
| 9 | 2484.7455 | 4.2 |
| 10 | 3812.73 | 1.99 |
| 11 | 108.15 | 0.7 |
| 12 | 1186.06 | 3.92 |
| 13 | 51.53 | 0.7 |
| 14 | 90.05 | 2.58 |
| 15 | 7804.53 | 5.99 |



# np.random.seed()函数

**np.random.seed()：** seed()中的参数被设置了之后，可以按顺序产生一组固定的数组。如果使用相同的seed()值，则每次生成的随机数都相同。如果不设置这个值，那么每次生成的随机数不同。

# 案例分析

```
import numpy as np
np.random.seed(1)
L1 = np.random.randn(3, 3)
L2 = np.random.randn(3, 3)
print(L1)
print(L2)
```

只调用一次**seed()**，两次的产生随机数不同

```
import numpy as np
np.random.seed(1)
L1 = np.random.randn(3, 3)
np.random.seed(1)
L2 = np.random.randn(3, 3)
print(L1)
print(L2)
```

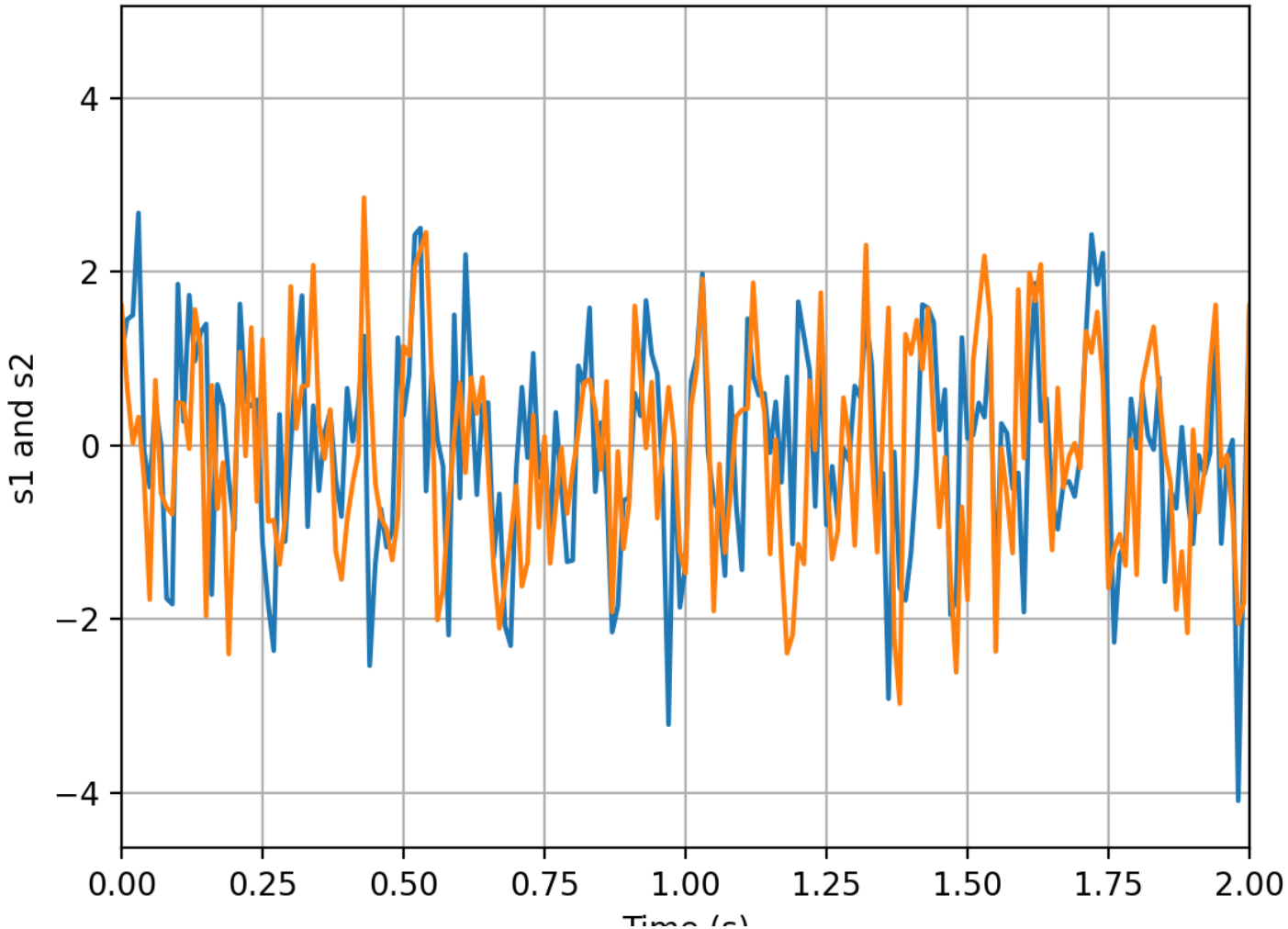
调用两次**seed()**，两次产生的随机数相同

# np.random.randn()函数

**np.random.randn()：** 生成服从标准正态分布的随机数。标准正态分布，也称为高斯分布，是一种概率分布，其概率密度函数呈钟形曲线，均值为0，标准差为1。在深度学习和统计学中，这个函数常用于生成符合正态分布的随机数据。

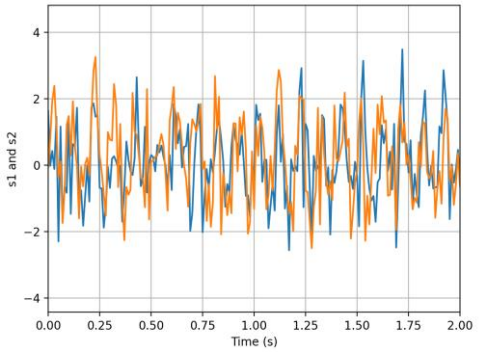
**np.random.randn(3, 4)：** 生成一个3行4列的二维数组，数组中的每个元素都是从标准正态分布中随机抽取的。

# 案例分析



# 案例分析

```
import matplotlib.pyplot as plt
import numpy as np
np.random.seed(1)
t = np.arange(0, 30, 0.01)
nse1 = np.random.randn(len(t))
nse2 = np.random.randn(len(t))
s1 = np.sin(2 * np.pi * 10 * t) + nse1
s2 = np.sin(2 * np.pi * 10 * t) + nse2
plt.plot(t, s1, t, s2); plt.xlim(0, 2)
plt.xlabel('Time (s)'); plt.ylabel('s1 and s2')
plt.grid(True); plt.show()
```

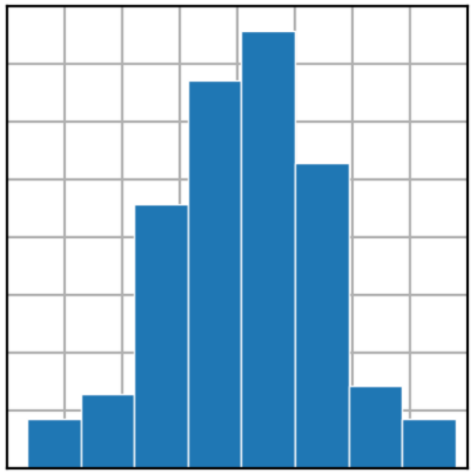


# Matplotlib常用函数

| 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 |

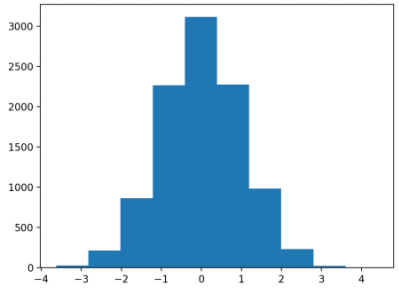
# 直方图

**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

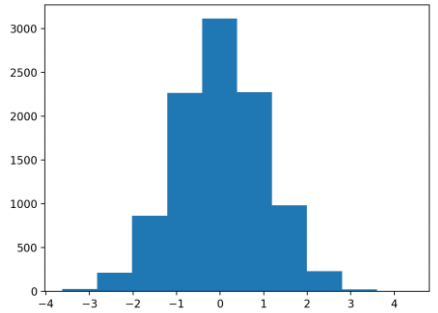
**bins：** 指定直方图条形的个数

**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

# 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

x=np. random. randn(10000)

plt. hist(x)

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 |
| boxplot() | 绘制箱型图 |

# 样本分位数

四分位数（Quartile）：指在统计学中把所有数值由小到大排列并分成四等份，处于三个分割点位置的数值。多应用于统计学中的箱线图绘制。

第一四分位数 (Q1)：第25%的数字

第二四分位数 (Q2)：第50%的数字

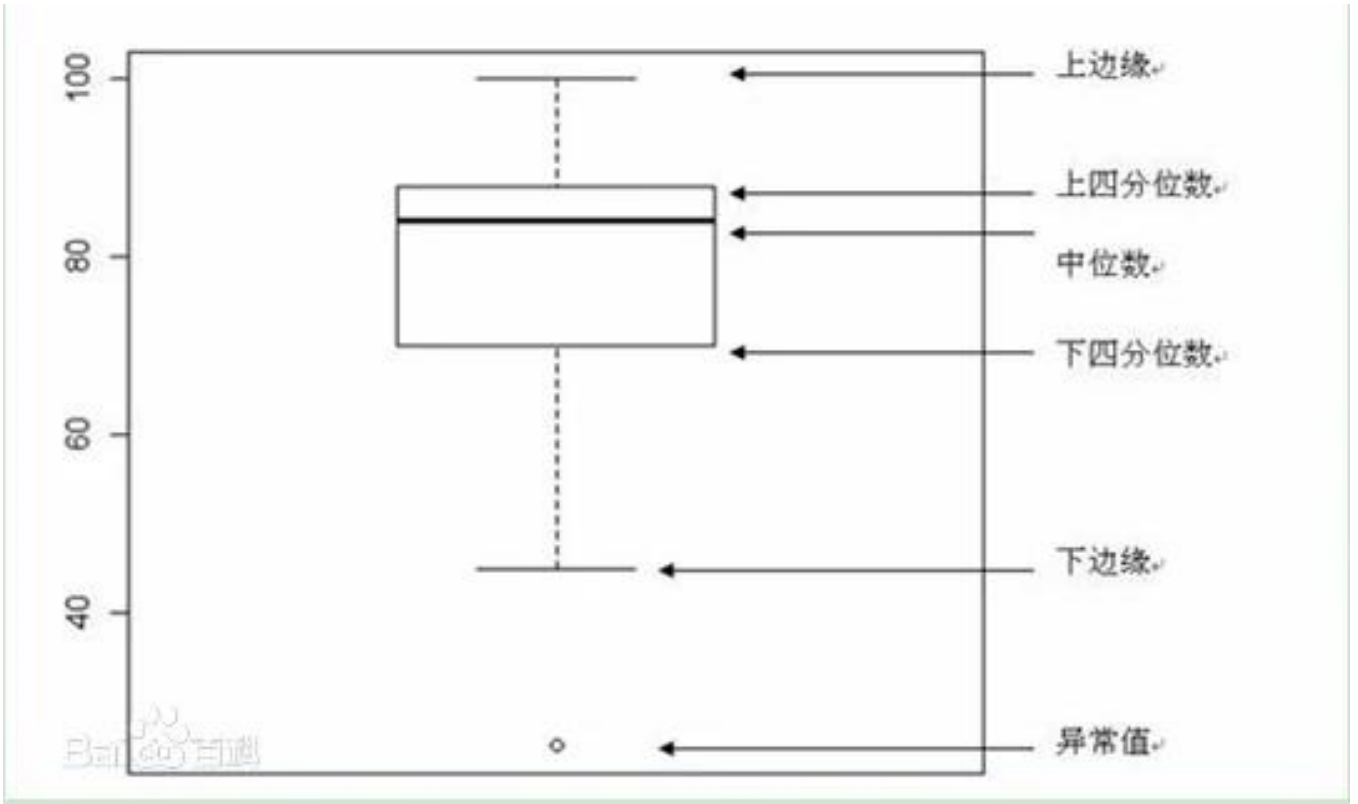
第三四分位数 (Q3)：第75%的数字



62

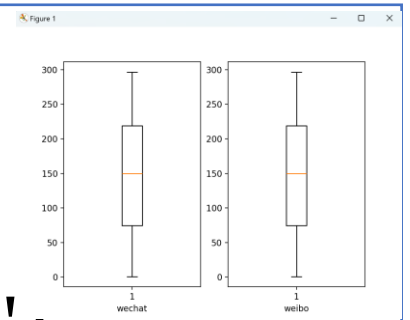
# 箱型图

## 1977年由美国统计学家John Tukey发明



# 案例分析

```
import numpy as np
import matplotlib.pyplot as plt
a=np.loadtxt("advertising.csv",delimiter=",")
h=a[:,0]; w=a[:,1]
h=h.reshape(-1,1); w=h.reshape(-1,1)
plt.subplot(121); plt.xlabel("wechat"); plt.boxplot(h)
plt.subplot(122); plt.xlabel("weibo"); plt.boxplot(w)
plt.show()
```

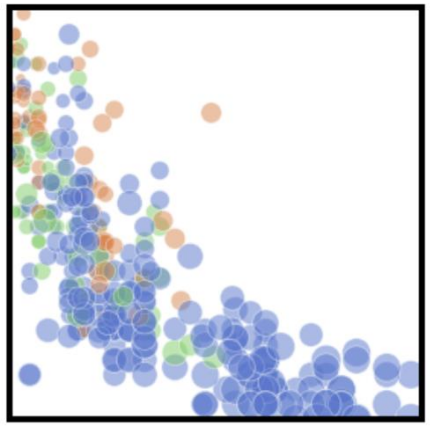


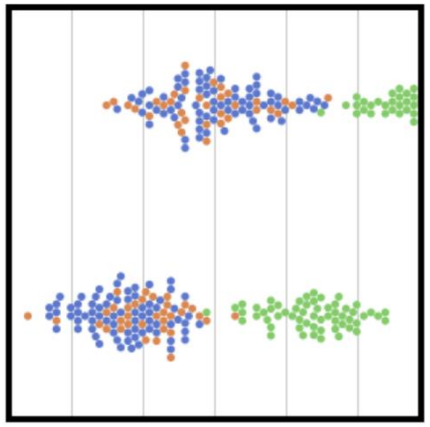
# Seaborn

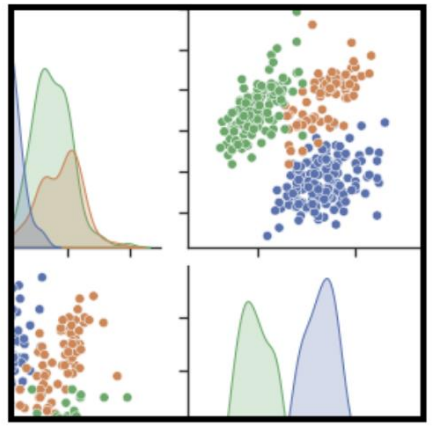
**Seaborn：**是一种基于matplotlib的图形可视化python libraty。它提供了一种高度交互式界面，便于用户能够做出各种有吸引力的统计图表。Seaborn其实是在matplotlib的基础上进行了更高级的API封装，从而使得作图更加容易，应该把Seaborn视为matplotlib的补充，而不是替代物。

**http://seaborn.pydata.org/**

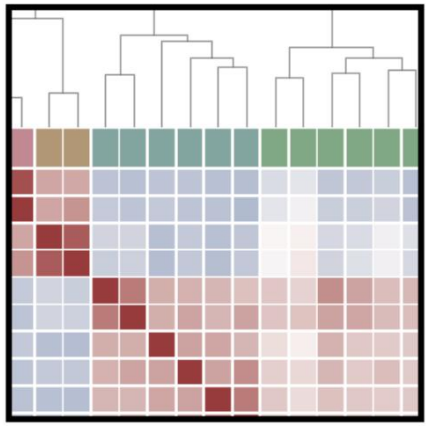
# Seaborn

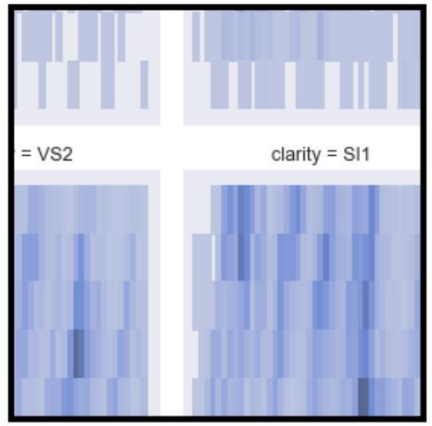







193 236 235 267 317 356 362
181 235 227 269 313 348 348
183 229 234 270 318 355 363
218 243 264 315 374 422 435
230 264 302 364 413 465 491
242 272 293 347 405 467 505
209 237 259 312 355 404 404
191 211 229 274 306 347 359
172 180 203 237 271 305 310




= VS2
clarity = SI1

# np.random.rand()函数

**np.random.rand()：** 生成[0, 1)区间内的均匀分布的随机数。

**np.random.rand(3, 4)：** 生成一个3行4列的二维数组，数组中的每个元素都是在[0, 1)区间内随机生成的。

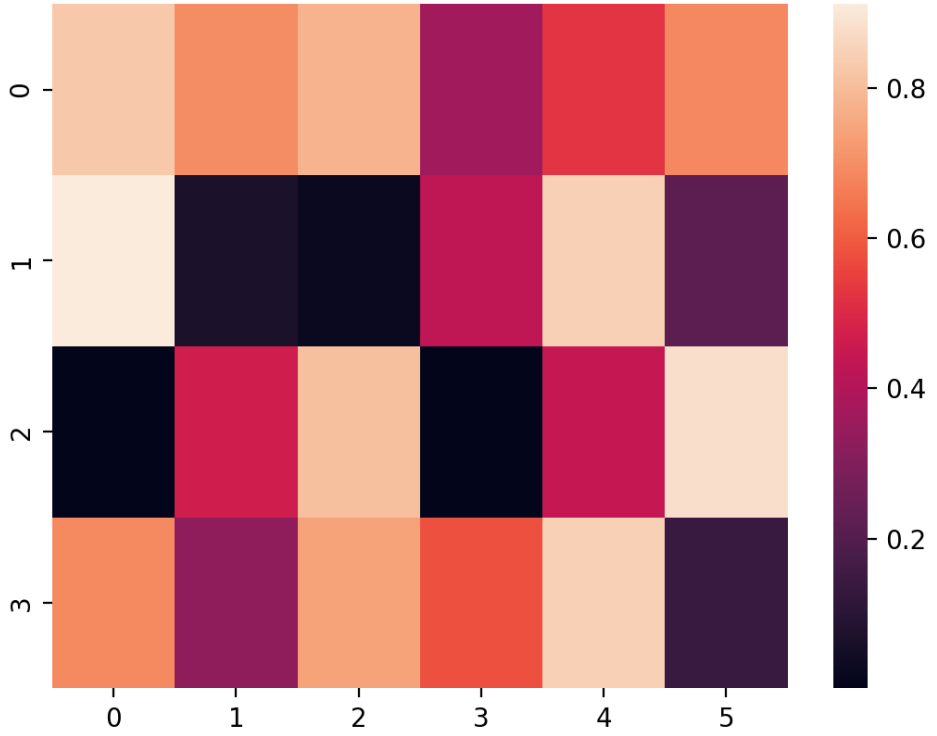
# np.random.random()函数

**np.random.random()：** 与np.random.rand()函数在功能上相同

**np.random.rand(3, 4)：** 生成一个3行4列的二维数组，数组中的每个元素都是在[0, 1)区间内随机生成的。

# 热力图

```
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
data = np.random.rand(4,6)
heat_map = sb.heatmap(data)
plt.show()
```



# 谢　谢