Python与金融数据挖掘(12)

文欣秀

wenxinxiu@ecust.edu.cn

Python应用领域

科学计算： Numpy、SciPy…

数据分析： Pandas、Matplotlib…

机器学习： Scikit-Learn、Keras…

深度学习： Pytorch、Mindspore…

…

Matplotlib常用函数

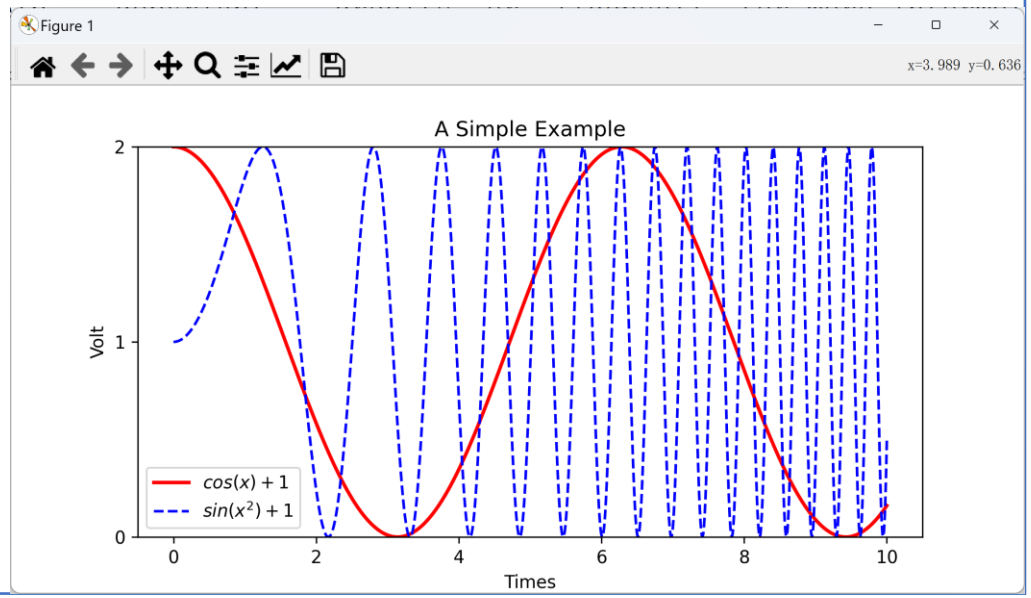
函数名称	函数作用
plot()	绘图折线图
show()	在本机显示图形
bar()	绘制垂直条形图
scatter()	绘制散点图
pie()	绘制饼图
subplot()	绘制子图
hist()	绘制直方图
boxplot()	绘制箱形图

Matplotlib常用函数

函数名称	函数作用
plt.xlabel()	添加x轴名称，可以指定位置、颜色、字体大小等
plt.ylabel()	添加y轴名称，可以指定位置、颜色、字体大小等
plt.xlim()	指定x轴的范围，确定一个数值区间
plt.ylim()	指定y轴的范围，确定一个数值区间
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()	指定图例，可以指定图例的大小、位置、标签

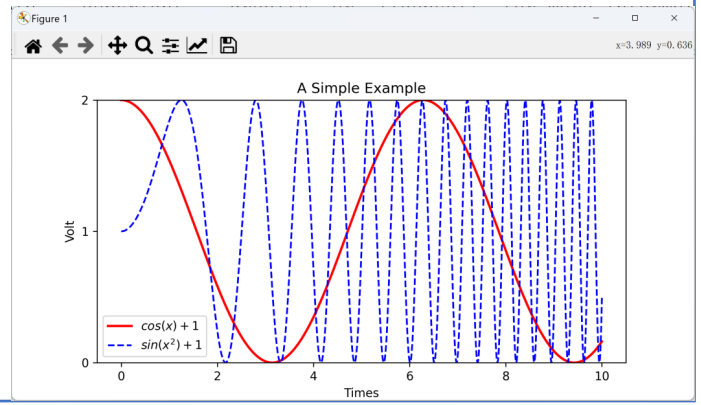
带标签的数学图形

```
import matplotlib.pyplot as plt
import numpy as np
x=np.linspace(0,10,1000)
y=np.cos(x)+1 #因变量y
z=np.sin(x**2)+1
plt.figure(figsize=(8,4))
```



带标签的数学图形

```
plt.plot(x,y,'r',label='$cos(x)+1$', linewidth=2)
plt.plot(x,z,'b--',label='$sin(x^2)+1$')
plt.xlabel('Times') ; plt.ylabel('Volt')
plt.title('A Simple Example') #标题
plt.yticks([0,1,2]) #显示的y轴范围
plt.ylim(0,2) #显示的y轴范围
plt.legend() #显示图例
plt.show()
```



Python应用领域

科学计算： Numpy、SciPy…

数据分析： Pandas、Matplotlib…

机器学习： Scikit-Learn、Keras…

深度学习： Pytorch、Mindspore…

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

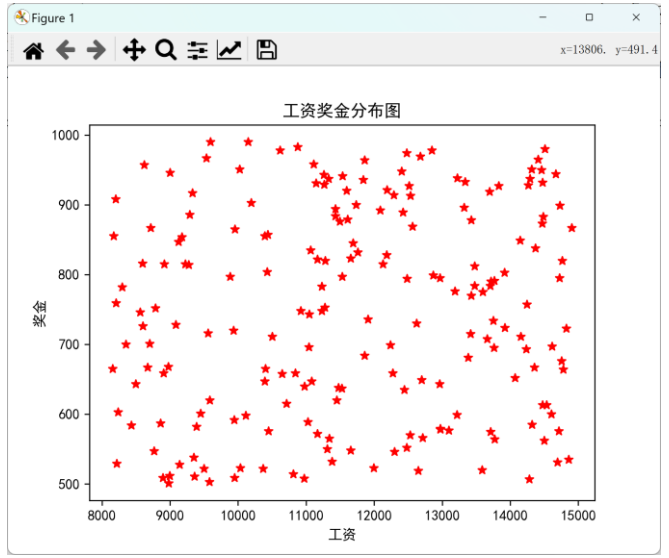
numpy.random. randint(low, high, size, dtype=int)：返回范围为[low, high)随机整数，size为数组尺寸

```
>>> 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?



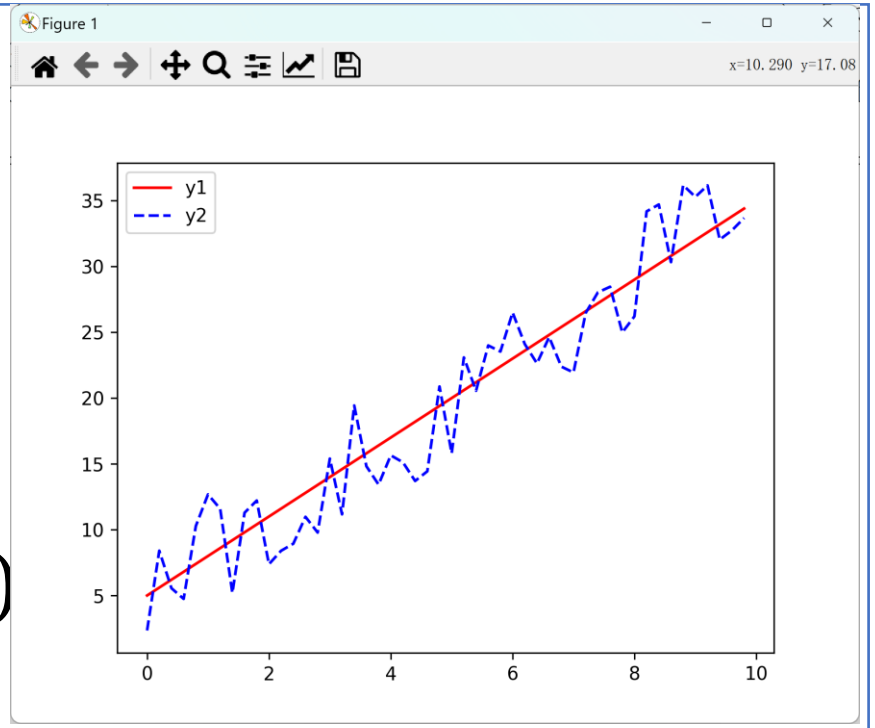
随机浮点数

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？

Numpy数据存储

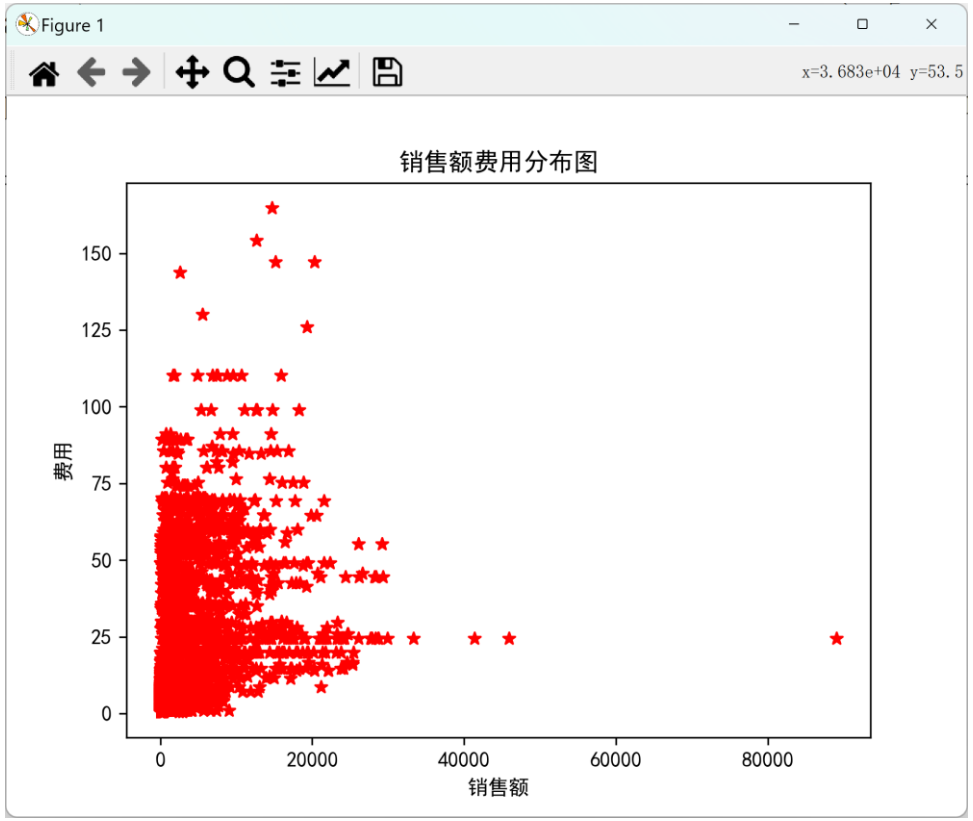
	A	B	C	D	E	F	G	H	I	J
1	5	5.6	6.2	6.8	7.4	8	8.6	9.2	9.8	10.4
2	5.2	9.3	5.4	10.2	2.5	12.3	9	12	9.4	12.1

```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(0,10,0.2)
y1=3*x+5; y2=[]
for i in y1:
    y2.append(i+np.random.uniform(-5,5))
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？

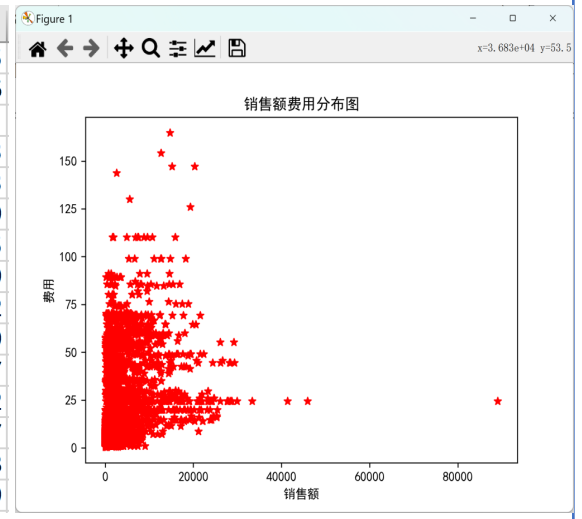
	A	B
1	261.54	35
2	6	2.56
3	2808.08	5.81
4	1761.4	89.3
5	160.2335	5.03
6	140.56	8.99
7	288.56	2.25
8	1892.848	8.99
9	2484.7455	4.2
10	3812.73	1.99
11	108.15	0.7
12	1186.06	3.92
13	51.53	0.7
14	90.05	2.58
15	7804.53	5.99



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=",").reshape(-1,2)
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



数据统计

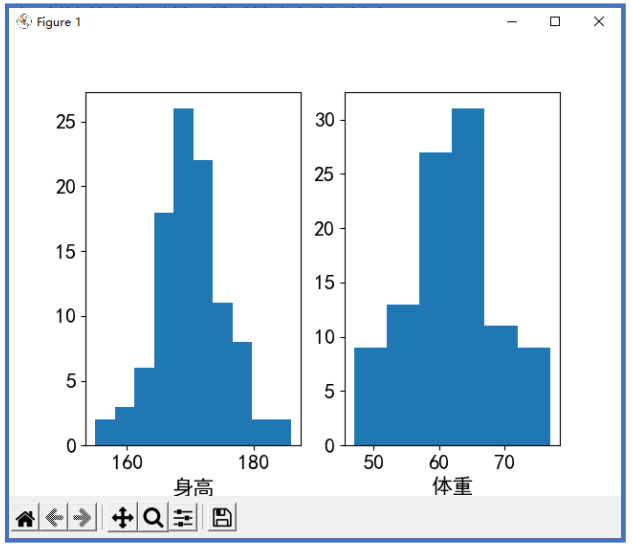
学校随机抽取100名学生，测量他们的身高和体重，所得数据如下表，画出身高和体重的直方图。

Pdata4_6_2.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

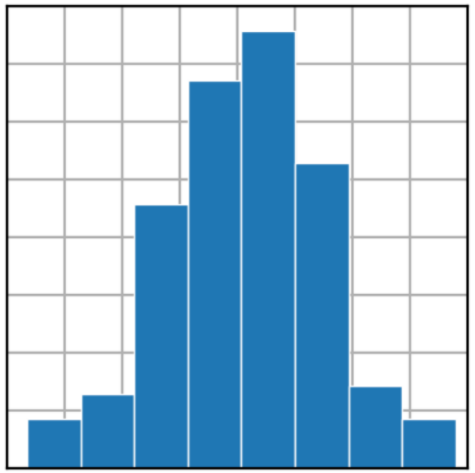
172	75	169	55	169	64	171	65	167	47
171	62	168	67	165	52	169	62	168	65
166	62	168	65	164	59	170	58	165	64
160	55	175	67	173	74	172	64	168	57
155	57	176	64	172	69	169	58	176	57
173	58	168	50	169	52	167	72	170	57
166	55	161	49	173	57	175	76	158	51
170	63	169	63	173	61	164	59	165	62
167	53	171	61	166	70	166	63	172	53
173	60	178	64	163	57	169	54	169	66
178	60	177	66	170	56	167	54	169	58
173	73	170	58	160	65	179	62	172	50

第 1 行，第 1 列　100%　Windows (CRLF)　UTF-8



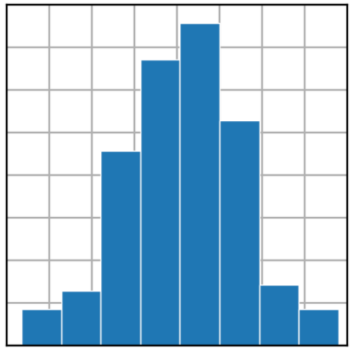
直方图

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

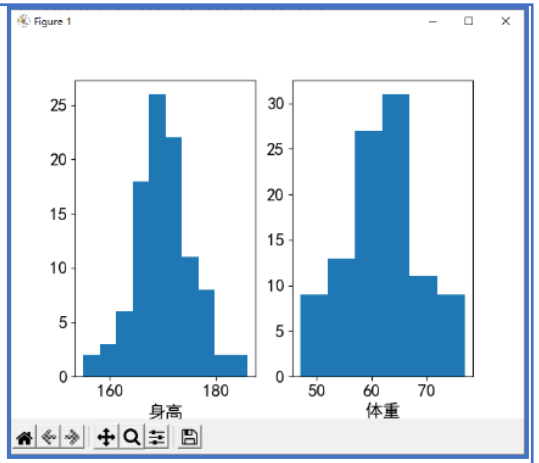
bins： 指定直方图条形的个数

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

数据统计

```
import numpy as np
import matplotlib.pyplot as plt
a=np.loadtxt("素材.txt")
h=a[:,::2]; w=a[:,1::2]
h=np.reshape(h,(-1,1)); w=np.reshape(w,(-1,1))
plt.rc('font',size=16); plt.rc('font',family="SimHei")
plt.subplot(121); plt.xlabel("身高"); plt.hist(h,10)
plt.subplot(122); plt.xlabel("体重"); plt.hist(w,6)
plt.show()
```



数据统计

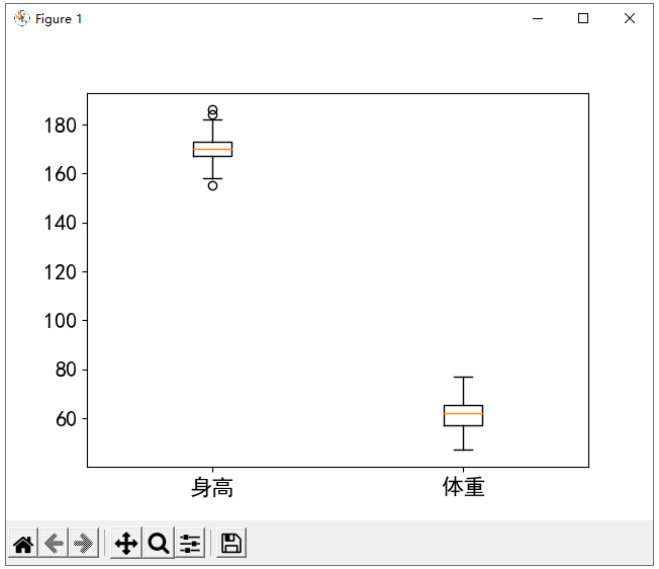
学校随机抽取100名学生，测量他们的身高和体重，所得数据如下表，画出身高和体重的箱型图。

Pdata4_6_2.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

172	75	169	55	169	64	171	65	167	47
171	62	168	67	165	52	169	62	168	65
166	62	168	65	164	59	170	58	165	64
160	55	175	67	173	74	172	64	168	57
155	57	176	64	172	69	169	58	176	57
173	58	168	50	169	52	167	72	170	57
166	55	161	49	173	57	175	76	158	51
170	63	169	63	173	61	164	59	165	62
167	53	171	61	166	70	166	63	172	53
173	60	178	64	163	57	169	54	169	66
178	60	177	66	170	56	167	54	169	58
173	73	170	58	160	65	179	62	172	50

第 1 行，第 1 列　100%　Windows (CRLF)　UTF-8



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

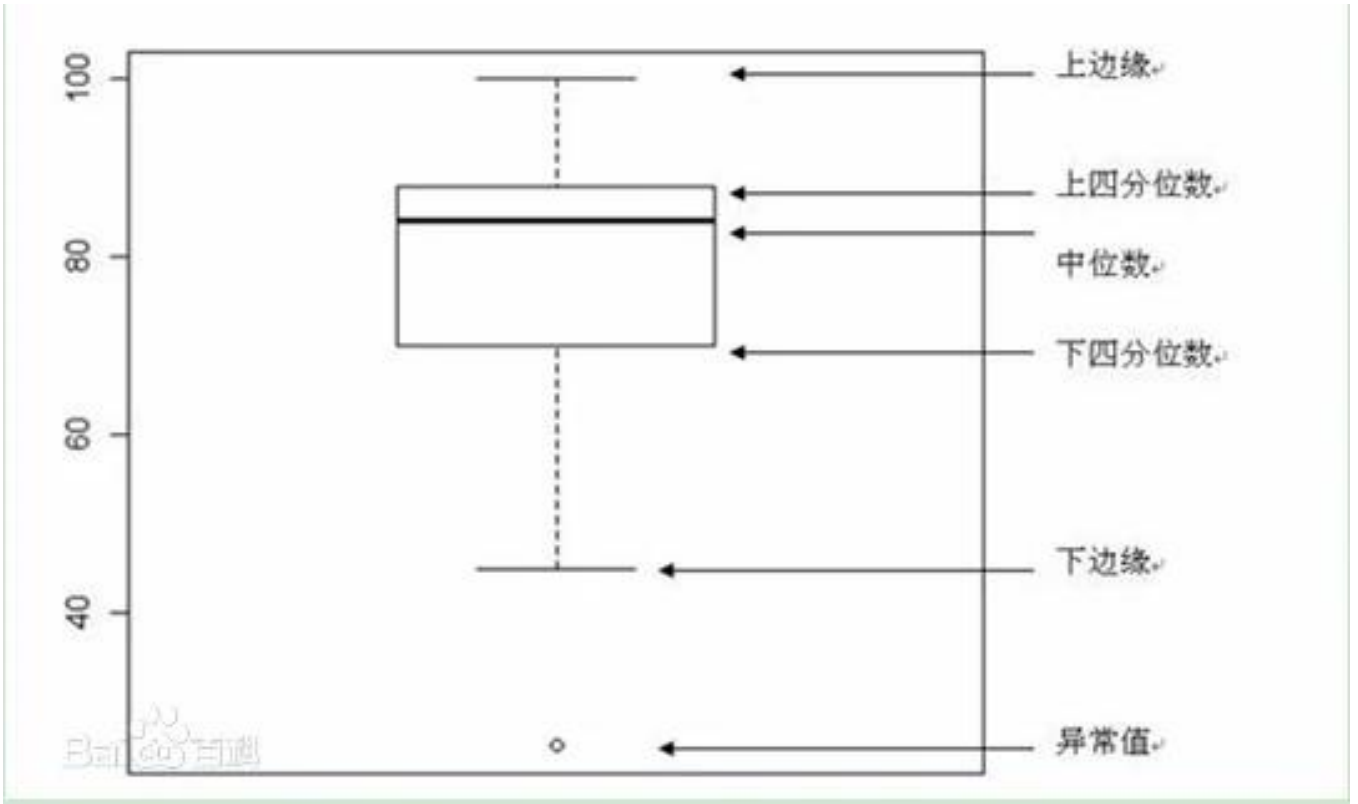
第二四分位数 (Q2)：第50%的数字

第三四分位数 (Q3)：第75%的数字

23

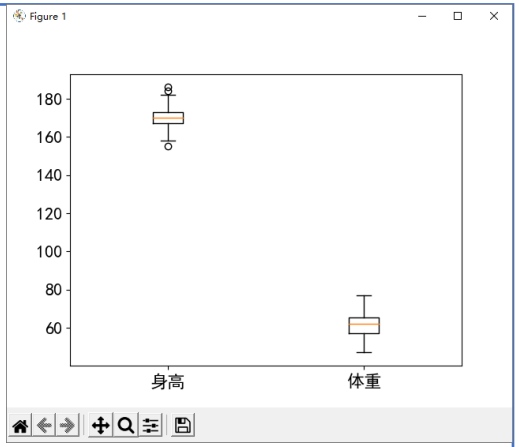
箱型图

1977年由美国统计学家John Tukey发明



身高体重箱型图

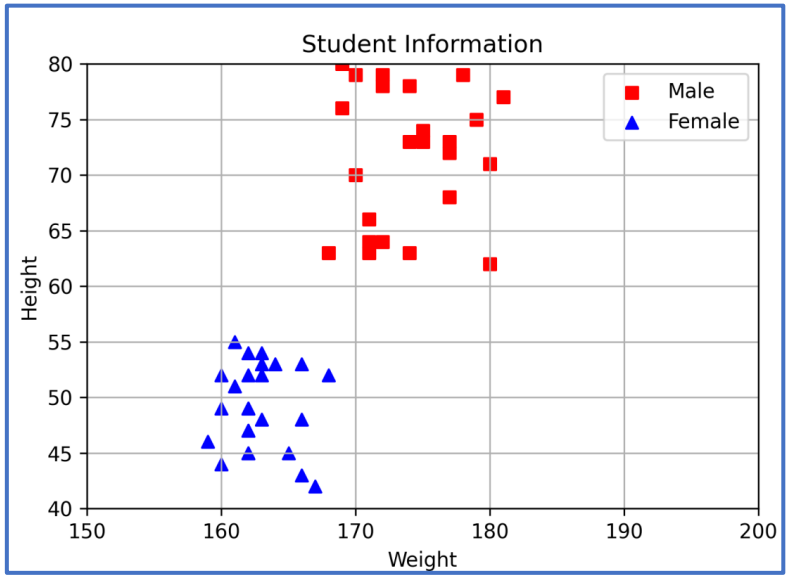
```
import numpy as np
import matplotlib.pyplot as plt
a=np.loadtxt("素材.txt")
h=a[:,::2]; w=a[:,1::2]
h=np.reshape(h,(-1,1)); w=np.reshape(w,(-1,1))
hw=np.hstack((h,w)) #数组行连接
plt.rc('font',size=16);plt.rc('font',family='SimHei')
plt.boxplot(hw,labels=['身高','体重'])
plt.show()
```



思　　考

如何根据学生的性别分类绘制不同颜色图形？

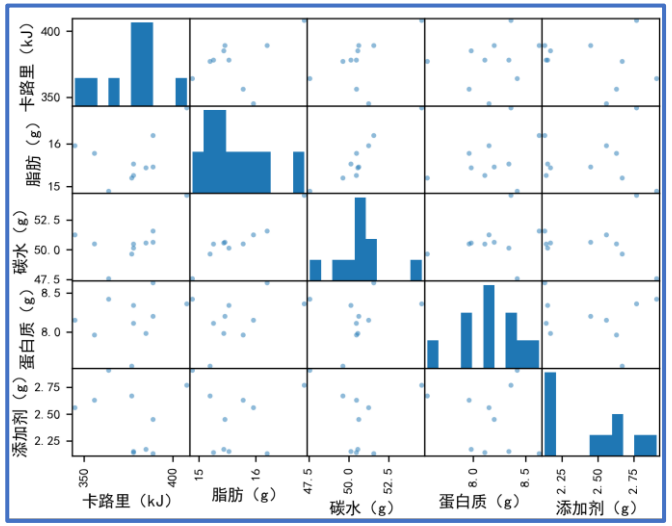
	A	B	C	D	E
1	No.	Gender	Age	Height	Weight
2	1	male	20	170	70
3	2	male	22	180	71
4	3	male	21	180	62
5	4	male	20	177	72
6	5	male	20	172	64
7	6	male	20	179	75
8	7	female	21	166	53
9	8	female	20	162	47
10	9	female	20	162	47
11	10	male	19	169	76
12	11	female	21	162	49



思　　考

如何根据绘制图形描述辣条各类信息之间的关系？

	A	B	C	D	E	F
1	辣条种类	卡路里（kJ）	脂肪（g）	碳水（g）	蛋白质（g）	添加剂（g）
2	辣条1	378	15.53	50.13	8.34	2.15
3	辣条2	389	15.46	50.62	8.2	2.45
4	辣条3	356	15.78	50.48	7.96	2.63
5	辣条4	377	15.2	49.63	7.56	2.67
6	辣条5	364	14.89	47.56	8.42	2.91
7	辣条6	408	16.85	54.56	8.36	2.77
8	辣条7	345	15.96	51.24	8.15	2.56
9	辣条8	389	16.2	51.56	8.63	2.13
10	辣条9	378	15.26	50.47	8.11	2.14
11	辣条10	385	15.44	50.56	7.98	2.17



Pandas

Pandas ： 基于NumPy 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了大量能快速便捷地处理数据的函数和方法。Pandas有三个重要的数据结构：一维系列(Series)和二维数据框(DataFrame)、三维(Panel)。

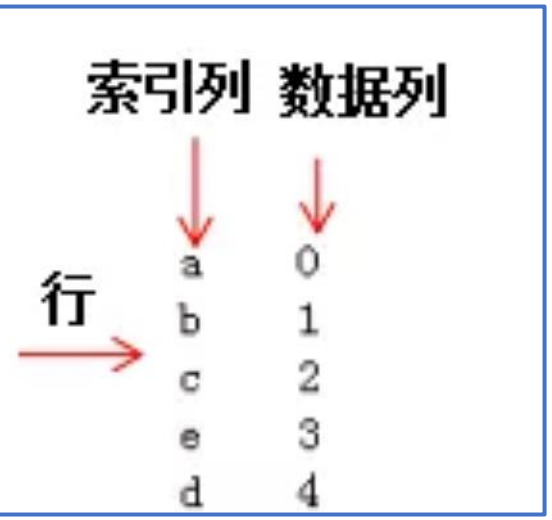
官网： https://pandas.pydata.org/

Pandas

Series： 序列，Pandas的基本数据结构，类似于一维数组，可以用于存储任意类型数据，是由一组数据以及与之相关的数据标签(即索引)组成。

```
import pandas as pd
import numpy as np
data = np. array([0, 1, 2, 3, 4])
s = pd. Series(data, index=['a','b','c','d','e'])
print(s)
```

```
a    0
b    1
c    2
d    3
e    4
dtype: int32
```

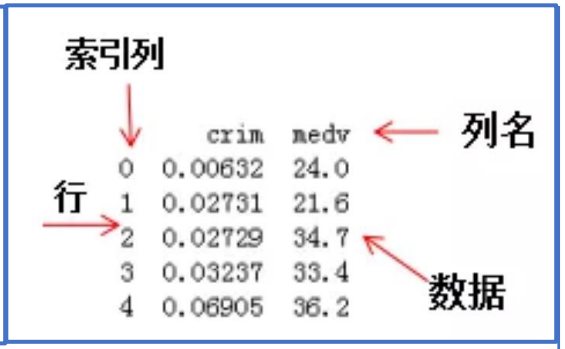


DataFrame

DataFrame： 二维数据结构，即数据以行和列的表格方式排列。

```
import numpy as np
import pandas as pd
data=np. array([[0.00632, 24.0],
 [0.02731, 21.6],[0.02729, 34.7],
 [0.03237, 33.4],[0.06905, 36.2]])
result = pd. DataFrame(data, columns=[ 'crim', 'medv'])
print(result)
```

```
    crim  medv
0  0.00632  24.0
1  0.02731  21.6
2  0.02729  34.7
3  0.03237  33.4
4  0.06905  36.2
```



DataFrame(data=二维数据[, index=行索引] [, columns=列索引]…)

DataFrame特点

- 列可以是不同的类型
- 索引为行标签
- 列名为列标签
- 可对行和列做算术运算

No.	Gender	Age	Height	Weight
202201	male	20	170	70
202202	male	22	180	71
202203	male	21	180	62
202204	male	20	177	72
202205	male	20	172	64

	No.	Gender	Age	Height	Weight
0	202201	male	20	170	70
1	202202	male	22	180	71
2	202203	male	21	180	62
3	202204	male	20	177	72
4	202205	male	20	172	64
5	202206	male	20	179	75

DataFrame案例

```
import numpy as np
import pandas as pd
data=np. array([[0.00632, 24.0],
 [0.02731, 21.6],[0.02729, 34.7],
 [0.03237, 33.4],[0.06905, 36.2]])
result = pd. DataFrame(data, columns=[ 'crim', 'medv'])
print(result)
print("**********")
print(result. max())
```

```
     crim  medv
0  0.00632  24.0
1  0.02731  21.6
2  0.02729  34.7
3  0.03237  33.4
4  0.06905  36.2
**********
crim     0.06905
medv    36.20000
dtype: float64
```

Pandas读CSV文档

```
import pandas as pd
data=pd. read_csv('student.csv')
print(data)              #打印全部数据
print(data[3:5])         #打印行索引为3、4行的数据
```

	A	B	C	D	E
1	No.	Gender	Age	Height	Weight
2	202201	male	20	170	70
3	202202	male	22	180	71
4	202203	male	21	180	62
5	202204	male	20	177	72
6	202205	male	20	172	64

Squeezed text (51 lines).

```
     No. Gender  Age  Height  Weight
3  202204   male   20     177      72
4  202205   male   20     172      64
```

Pandas读CSV文档

```
import pandas as pd
data=pd. read_csv('student.csv', index_col=0)
print(data)              #打印全部数据
print(data[3:5])         #打印行索引为3、4行的数据
```

Squeezed text (52 lines).

No.	Gender	Age	Height	Weight
202204	male	20	177	72
202205	male	20	172	64

Pandas读CSV文档显示部分数据

```
import pandas as pd

data=pd. read_csv('student.csv')

print(data. head())                    #打印前5行数据

print("**********************************")

print(data. tail())                    #打印后5行数据
```

```
   No. Gender  Age  Height  Weight
0  202201   male   20     170      70
1  202202   male   22     180      71
2  202203   male   21     180      62
3  202204   male   20     177      72
4  202205   male   20     172      64
******************************
    No.  Gender  Age  Height  Weight
45  202246   male   21     174      73
46  202247 female   21     163      53
47  202248   male   21     175      74
48  202249   male   21     172      79
49  202250 female   20     166      48
```

Pandas读CSV文档显示部分数据

```
import pandas as pd
data=pd. read_csv('student.csv', index_col=0)
print(data. head())                    #打印前
print("**********************************
print(data. tail())                    #打印后
```

```
        Gender  Age  Height  Weight
No.
202201    male   20     170      70
202202    male   22     180      71
202203    male   21     180      62
202204    male   20     177      72
202205    male   20     172      64
**********************************
        Gender  Age  Height  Weight
No.
202246    male   21     174      73
202247  female   21     163      53
202248    male   21     175      74
202249    male   21     172      79
202250  female   20     166      48
```

DataFrame数据选取方法

选取类型	选取方法	说明
基于位置序号选取	Obj.iloc[iloc, cloc]	选取某行某列
	Obj.iloc[ilocList,clocList]	选取多行多列
	Obj.iloc[a:b,c:d]	选取a~b-1行，c~d-1列

获取部分数据

```
import pandas as pd

data=pd. read_csv('student.csv')

print(data. iloc[1,2])

print(data. iloc[[0,2],[3,4]])

print(data. iloc[0:3,3:5])
```

No.	Gender	Age	Height	Weight
202201	male	20	170	70
202202	male	22	180	71
202203	male	21	180	62
202204	male	20	177	72
202205	male	20	172	64

```
22
```

```
   Height  Weight
0     170      70
2     180      62
```

```
   Height  Weight
0     170      70
1     180      71
2     180      62
```

DataFrame数据选取方法

选取类型	选取方法	说明
基于索引名选取	Obj[col]	选取某列
	Obj[colList]	选取某几列
	Obj.loc[index,col]	选取某行某列
	Obj.loc[indexList,colList]	选取多行多列

获取部分数据

```
import pandas as pd
data=pd. read_csv('student.csv' , index_col=0)
print(data["Gender"])
print(data[["Gender","Age"]])
print(data. loc[202202])
print(data. loc[[202201, 202203],["Height","Weight"]])
```

No.	Gender	Age	Height	Weight
202201	male	20	170	70
202202	male	22	180	71
202203	male	21	180	62
202204	male	20	177	72
202205	male	20	172	64

Squeezed text (52 lines).

Squeezed text (52 lines).

```
Gender    male
Age         22
Height     180
Weight      71
Name: 202202, dtype: object
```

No.	Height	Weight
202201	170	70
202203	180	62

Pandas读CSV文档显示指定数据

- 编写程序，打印前2位同学的身高、体重信息

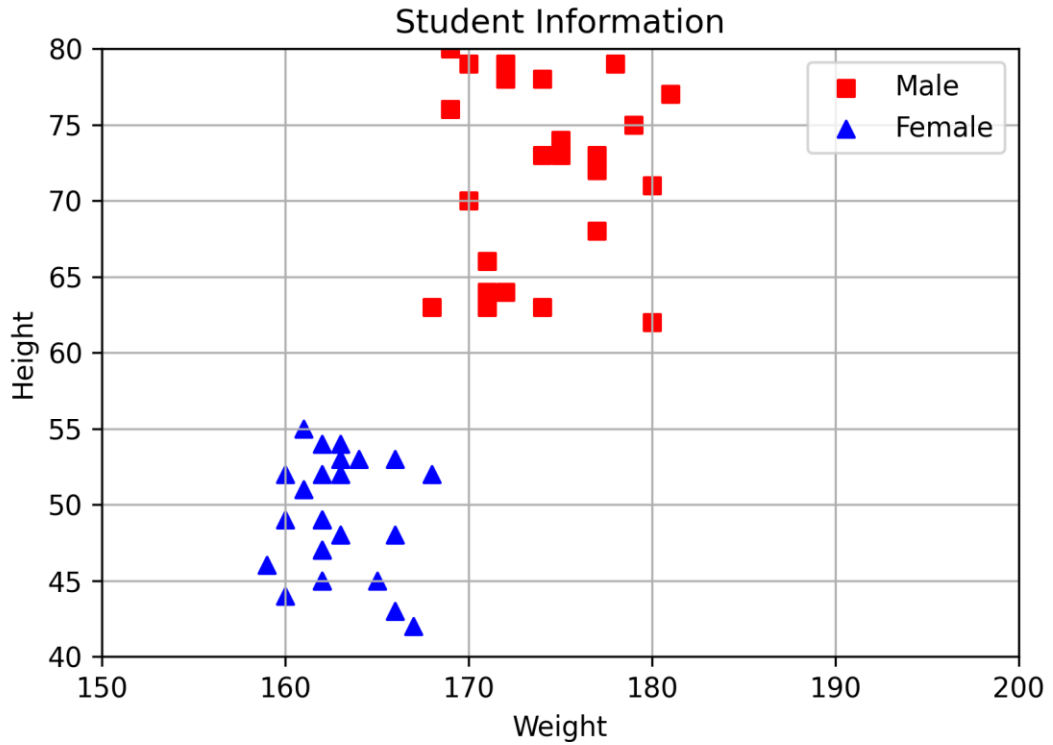
```
import pandas as pd
data=pd. read_csv('student.csv', index_col=0)
print(data. iloc[0:2,2:4])
print("**********")
print(data. loc[[202201,202202],["Height","Weight"]])
```

```
        Height  Weight
No.
202201     170      70
202202     180      71
**********
        Height  Weight
No.
202201     170      70
202202     180      71
```

DataFrame数据选取方法

选取类型	选取方法	说明
条件筛选	**Obj.loc[condition,colList]**	使用索引构造条件表达式
	Obj.iloc[condition,clocList]	使用位置序号构造条件表达式

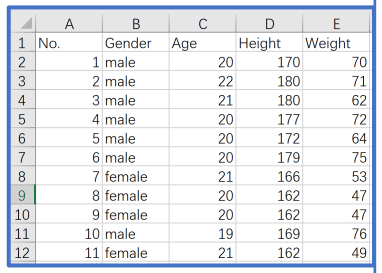
分类图表绘制




```
data1= data[data['Gender'] == 'male']  #筛选出男生
data1= data. loc[data['Gender'] == 'male'] #筛选出男生
```

男女生信息统计

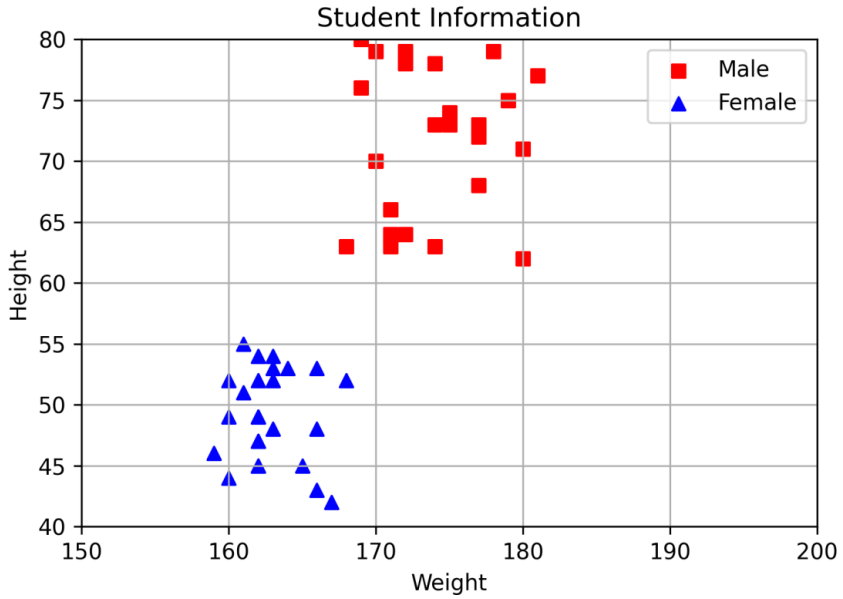
```
import matplotlib.pyplot as plt   #导入matplotlib.pyplot
import pandas as pd
#绘制散点图观察学生身高和体重之间的关系。
data = pd. read_csv('student.csv', index_col=0)
#将数据按性别分组，分别绘制散点图
data1= data. loc[data['Gender'] == 'male'] #筛选出男生
data2= data. loc[data['Gender'] == 'female'] #筛选出女生
#分组绘制男生、女生的散点图
plt. figure(figsize=(6,4))
```



	A	B	C	D	E
1	No.	Gender	Age	Height	Weight
2	1	male	20	170	70
3	2	male	22	180	71
4	3	male	21	180	62
5	4	male	20	177	72
6	5	male	20	172	64
7	6	male	20	179	75
8	7	female	21	166	53
9	8	female	20	162	47
10	9	female	20	162	47
11	10	male	19	169	76
12	11	female	21	162	49

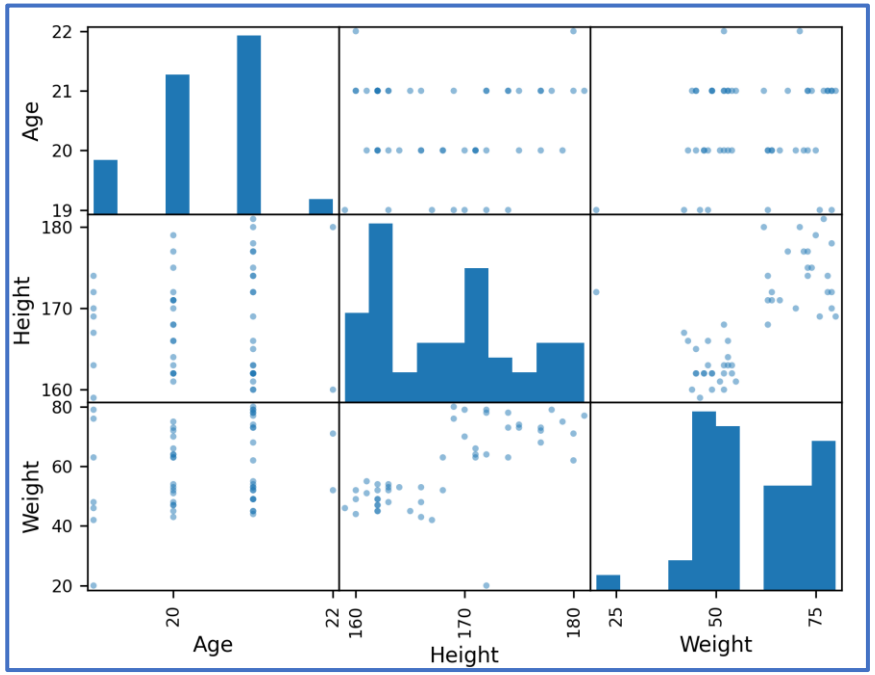
男女生信息统计

```
plt.scatter(data1['Height'],data1['Weight'],c='r',marker='s',label='Male')#正方形
plt.scatter(data2['Height'],data2['Weight'],c='b',marker='^',label='Female') #正三角形
plt.xlim(150,200)                #x轴范围
plt.ylim(40,80)                  #y轴范围
plt.title('Student Information')    #标题
plt.xlabel('Weight')             #x轴标题
plt.ylabel('Height')             #y轴标题
plt.grid()                         #网格线
plt.legend(loc='upper right')  #图例显示位置
plt.show()
```



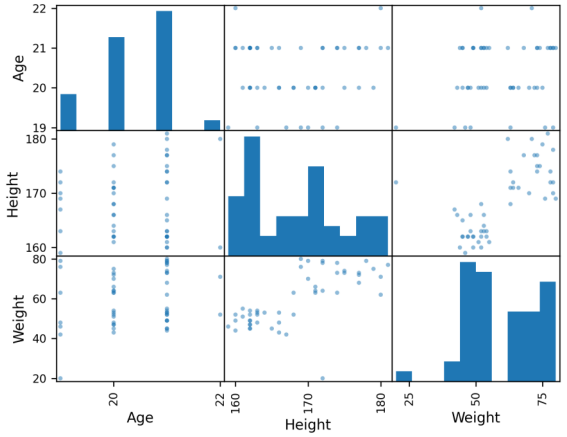
多项信息图形绘制

编写程序，绘制散点图矩阵观察学生各项信息（年龄、身高、体重）之间的关系。



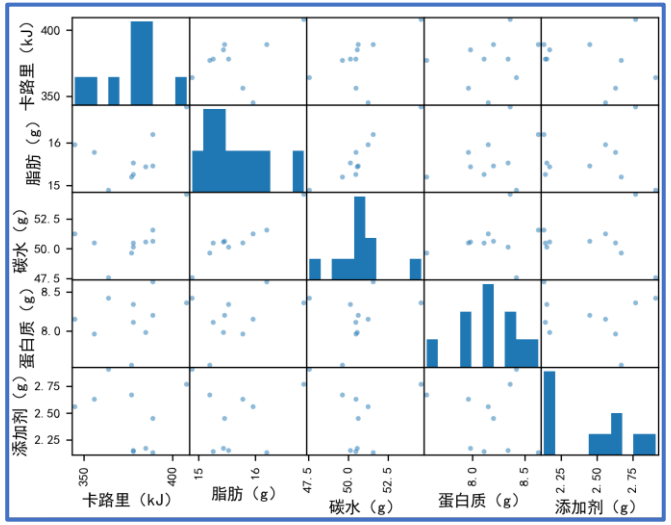
男女生信息统计

```
import matplotlib.pyplot as plt   #导入matplotlib.pyplot
import pandas as pd
data = pd. read_csv('student.csv', index_col=0)
result=data[['Age','Height','Weight']]
pd.plotting. scatter_matrix(result)
plt.show()
```



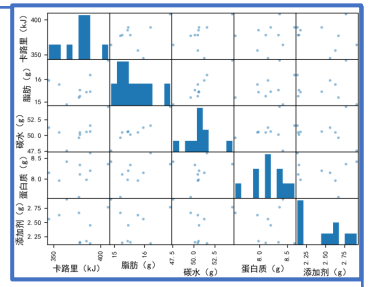
思　　考

	A	B	C	D	E	F
1	辣条种类	卡路里（kJ）	脂肪（g）	碳水（g）	蛋白质（g）	添加剂（g）
2	辣条1	378	15.53	50.13	8.34	2.15
3	辣条2	389	15.46	50.62	8.2	2.45
4	辣条3	356	15.78	50.48	7.96	2.63
5	辣条4	377	15.2	49.63	7.56	2.67
6	辣条5	364	14.89	47.56	8.42	2.91
7	辣条6	408	16.85	54.56	8.36	2.77
8	辣条7	345	15.96	51.24	8.15	2.56
9	辣条8	389	16.2	51.56	8.63	2.13
10	辣条9	378	15.26	50.47	8.11	2.14
11	辣条10	385	15.44	50.56	7.98	2.17



辣条信息统计

```
import matplotlib.pyplot as plt   #导入matplotlib.pyplot
import pandas as pd
plt.rcParams['font.family']=['SimHei']
data = pd. read_csv('mydata.csv',index_col=0, encoding="gb2312")
result=data['卡路里（kJ）','脂肪（g）','碳水（g）','蛋白质（g）','添加剂（g）']
pd.plotting.scatter_matrix(result)
plt.show()
```



谢　谢