

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、 Sympy...

数据分析： Pandas、 Matplotlib 、 Seaborn...

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

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
```

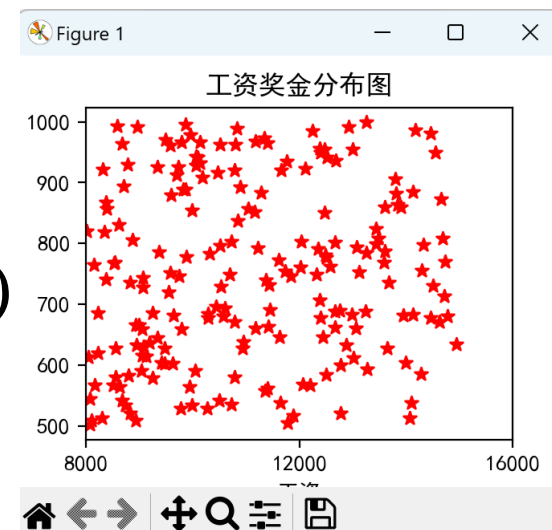
```
>>> grade=np. random. randint(1,5,size=10) # 产生10个[1,5)之间随机整数
```

```
>>> 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.xlim(8000,16000);plt.xticks([8000,12000,16000])
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
```

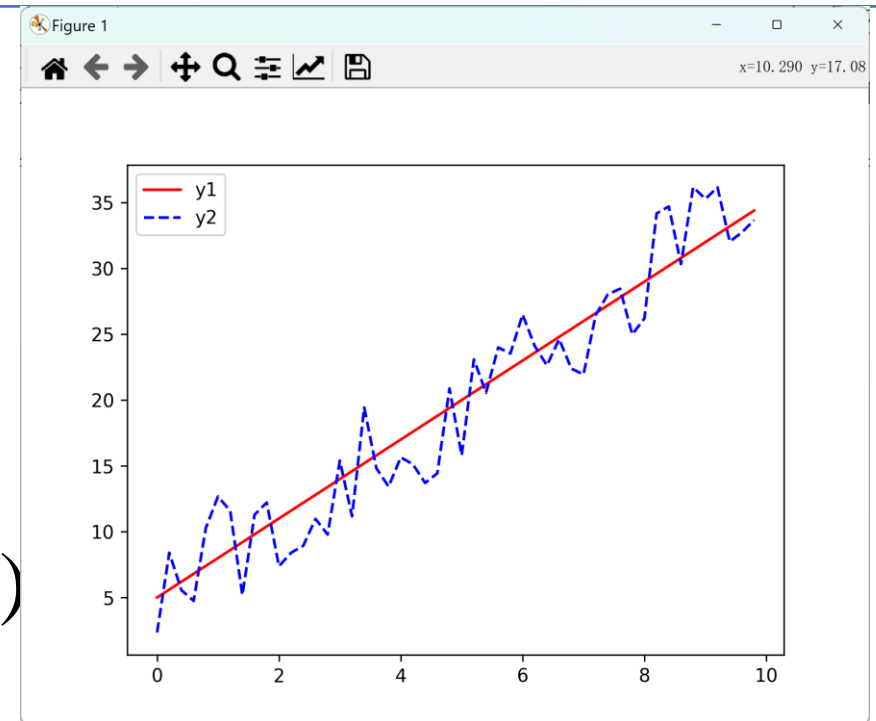
```
>>> test=np. random. uniform() # 产生1个[0,1)之间随机浮点数
```

```
>>> 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()
```



如何进行相关性分析？

案例分析

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import pearsonr
x = np.arange(0, 10, 0.2)
y1 = 3 * x + 5
y2 = []
for i in y1:
    y2.append(i + np.random.uniform(-5, 5))
corr = pearsonr(y1, y2)
print('相关系数r值为: ' + str(corr[0]))
```

相关系数 r 为: 0.95

如何保存数据到文件中?

Numpy数据存储

```
import numpy as np
```

	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 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))
```

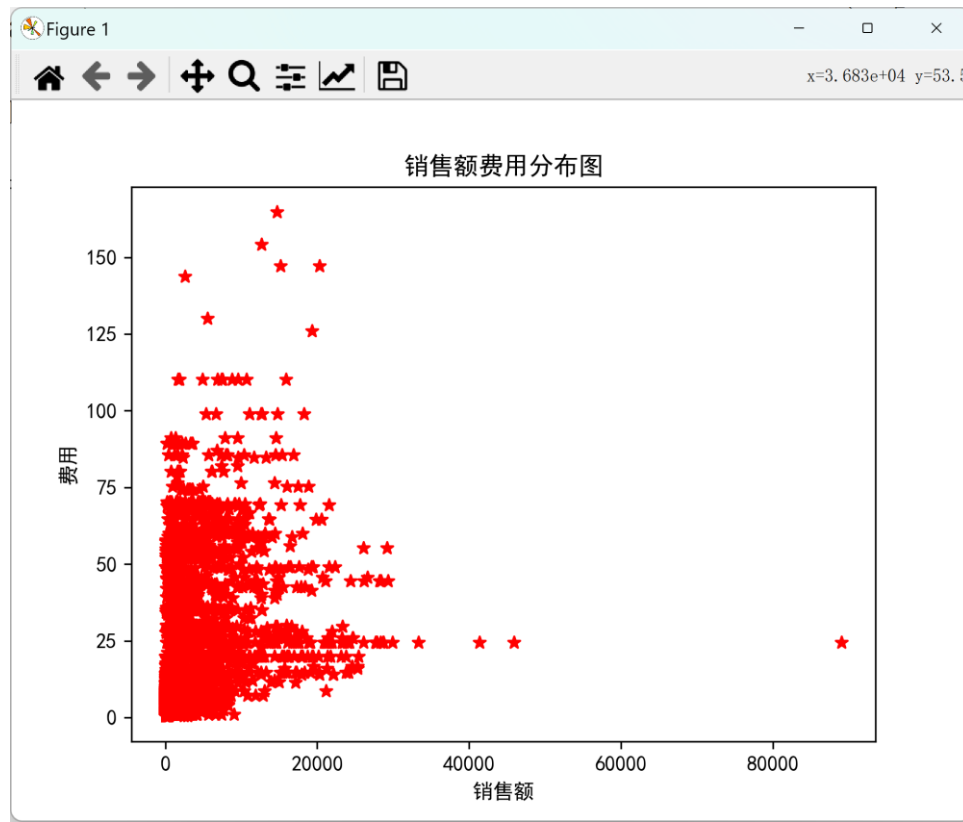
```
c=[y1,y2]
```

```
np.savetxt('result.csv',c,fmt='%.1f',delimiter=',', newline='\n')
```

思考

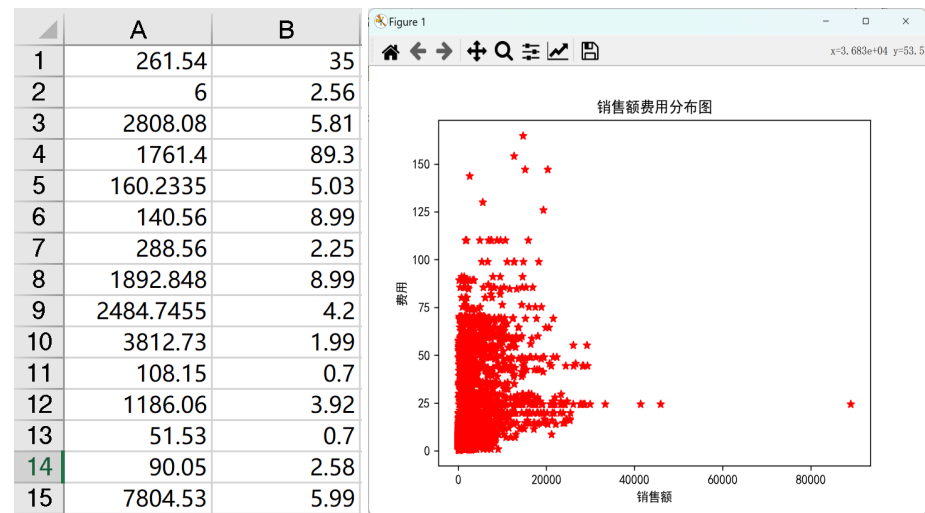
如何从文件中读取销售额和费用并绘制图形？

	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

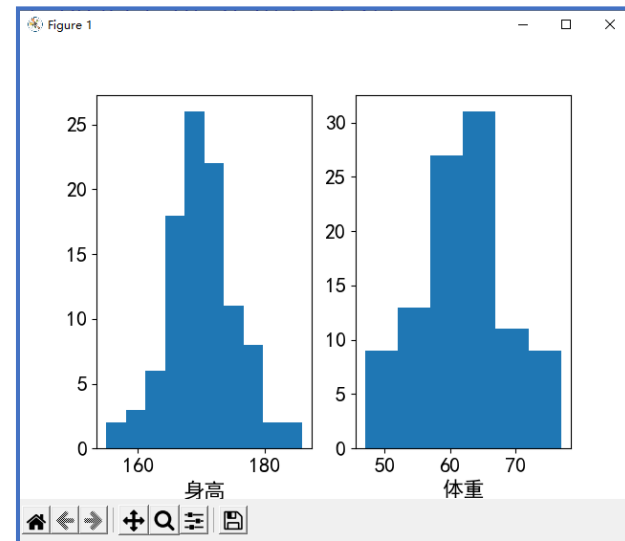
```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.family']=['SimHei']
result=np.loadtxt('trade.csv',delimiter=',').reshape(-1,2)
money=result[:,0]
cost=result[:,1]
plt.scatter(money,cost,c="r",marker="*")
plt.xlabel("销售额")
plt.ylabel("费用")
plt.title('销售额费用分布图')
plt.show()
```



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。

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



直方图

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

x: 指定要绘制直方图的数据

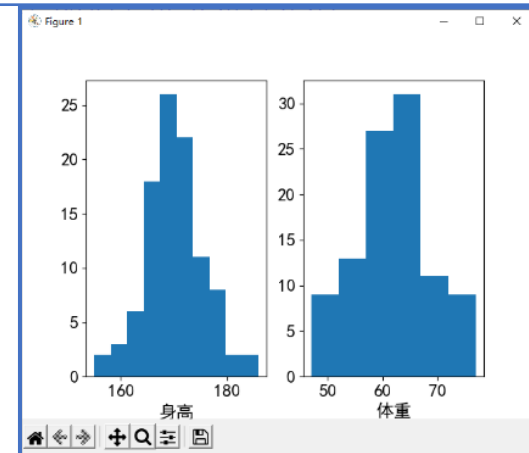
bins: 指定直方图条形的个数

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

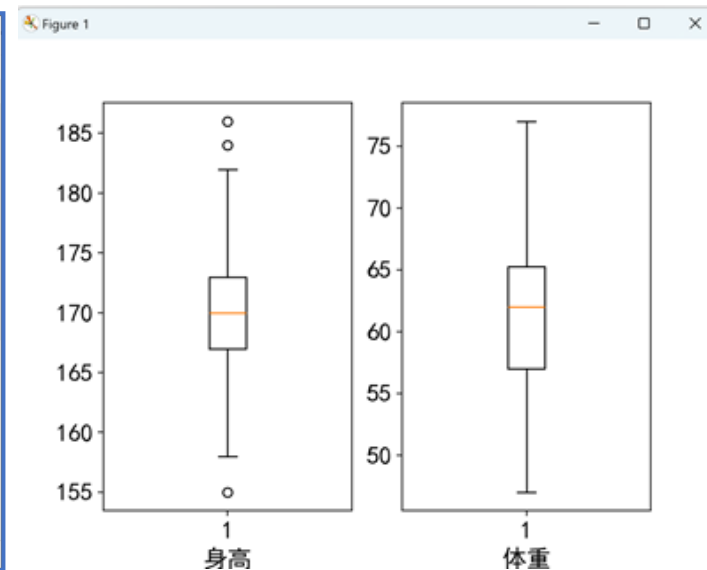
```
import numpy as np
import matplotlib.pyplot as plt
a=np.loadtxt("素材.txt")
h=a[:,::2]; w=a[:,1::2]
h=h.reshape(-1,1); w=w.reshape(-1,1)
plt.rc('font',size=16); plt.rc('font',family="SimHei")
plt.subplot(121); plt.xlabel("身高"); plt.hist(h,10)
plt.subplot(122); plt.xlabel("体重"); plt.hist(w,6)
plt.show()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



数据统计

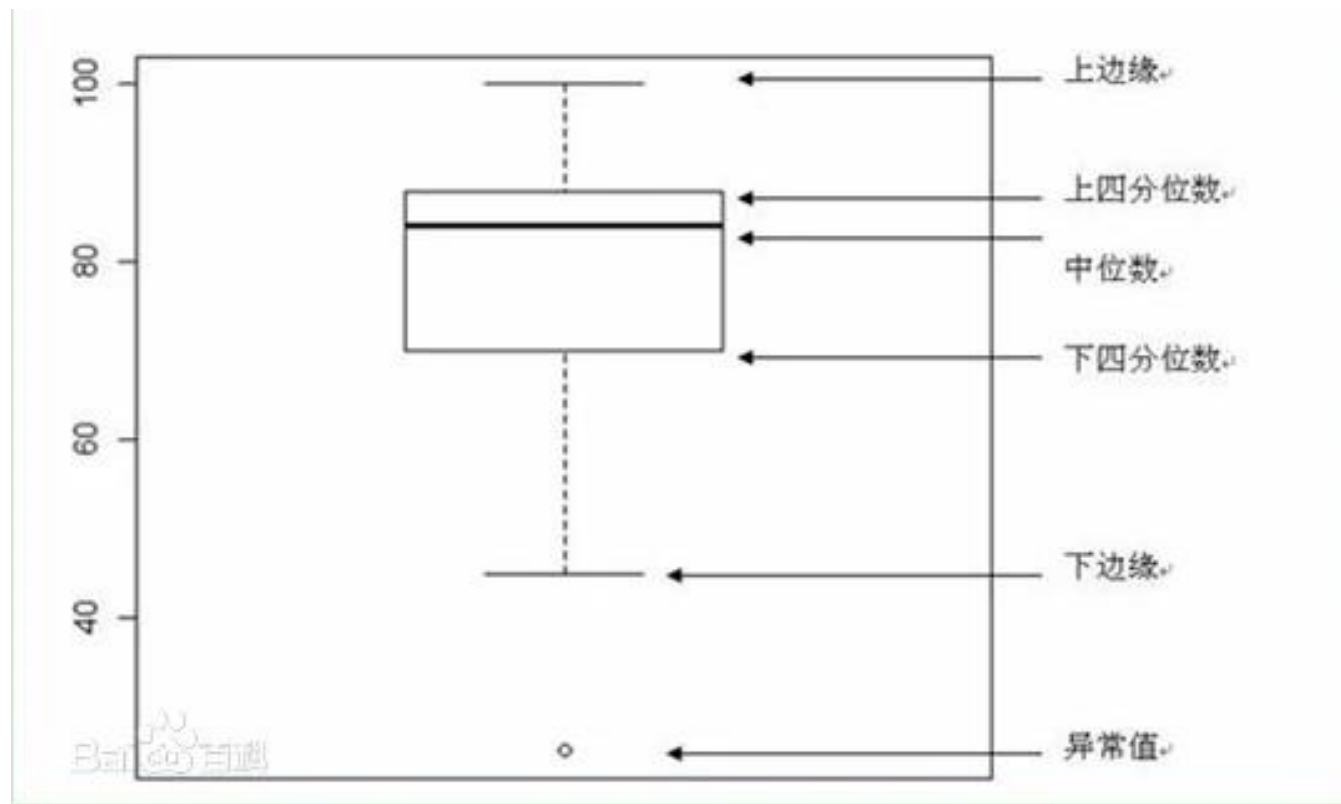
学校随机抽取100名学生，测量他们的身高和体重，所得数据如下表，画出身高和体重的箱型图。

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



箱型图

1977年由美国统计学家John Tukey发明



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。

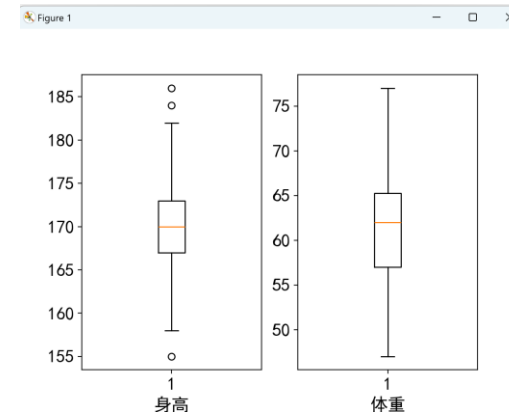
第一四分位数 (Q1): 第25%的数字

第二四分位数 (Q2): 第50%的数字

第三四分位数 (Q3): 第75%的数字

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=h.reshape(-1,1); w=w.reshape(-1,1)
plt.rc('font',size=16); plt.rc('font',family="SimHei")
plt.subplot(121); plt.xlabel("身高"); plt.boxplot(h)
plt.subplot(122); plt.xlabel("体重"); plt.boxplot(w)
plt.show()
```

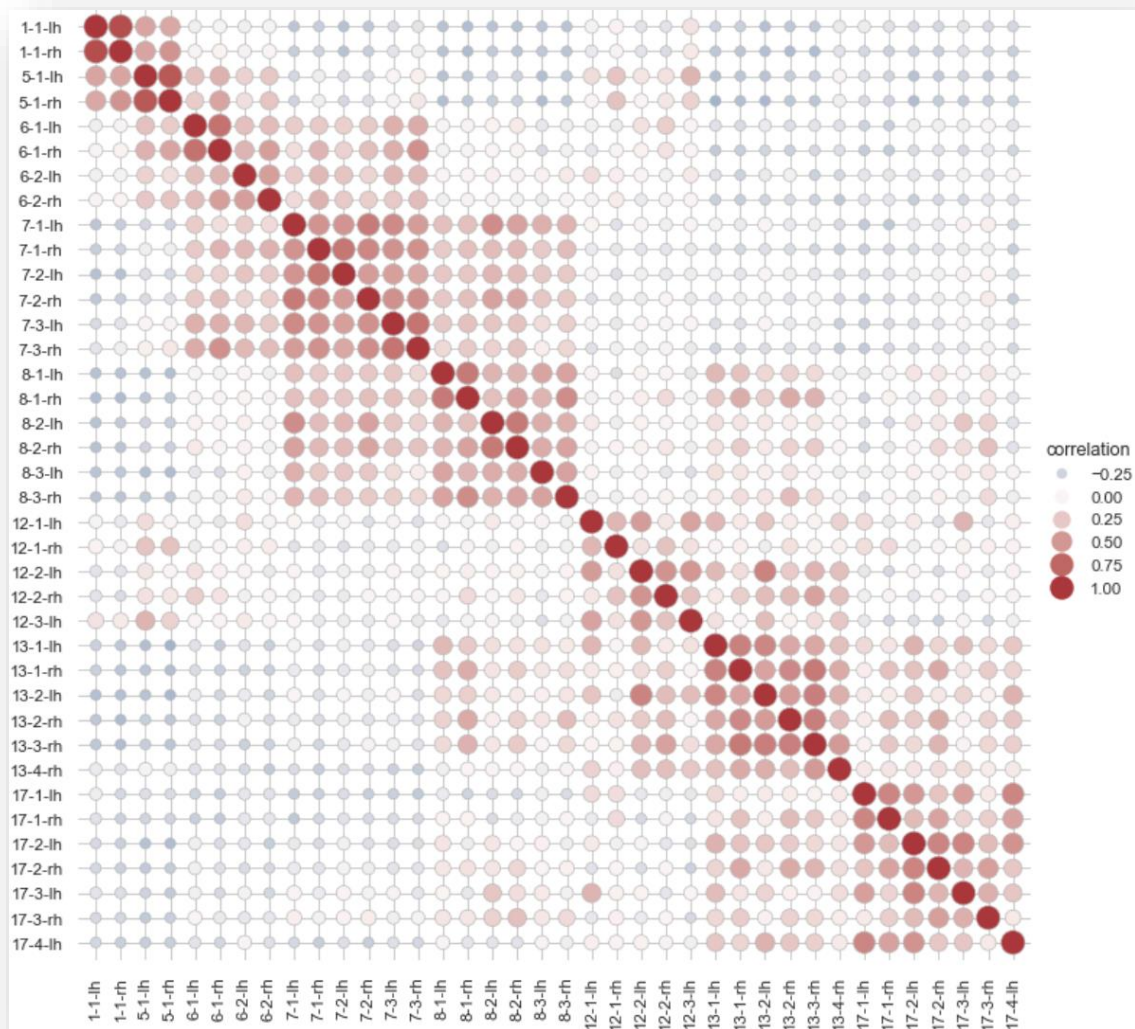


SEABORN

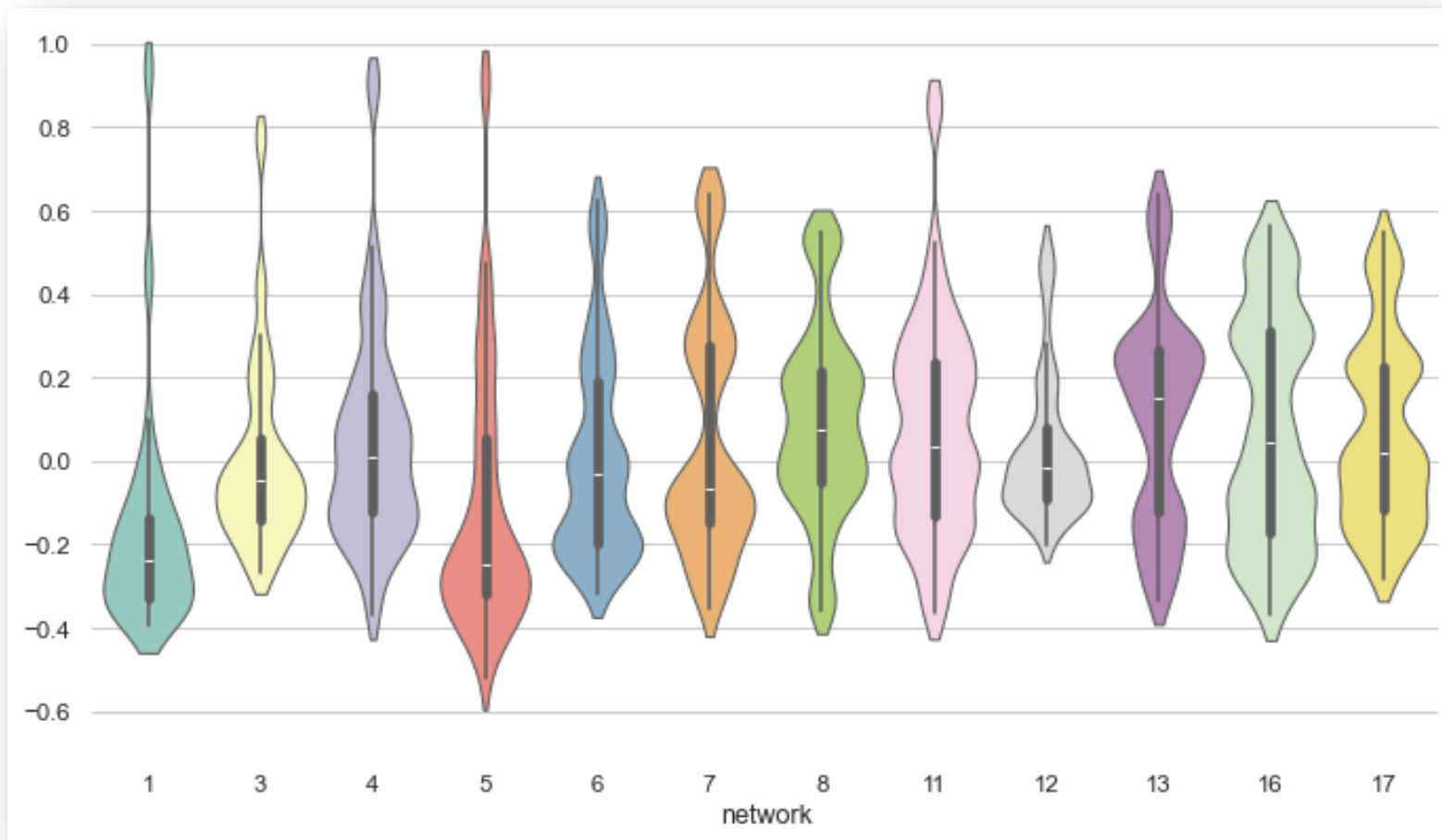
Seaborn: 是一种基于matplotlib的图形可视化python library。它提供了一种高度交互式界面，便于用户能够做出各种有吸引力的统计图表。Seaborn其实是在matplotlib的基础上进行了更高级的API封装，从而使得作图更加容易，应该把Seaborn视为matplotlib的补充，而不是替代物。

<http://seaborn.pydata.org/>

散点热力图案例



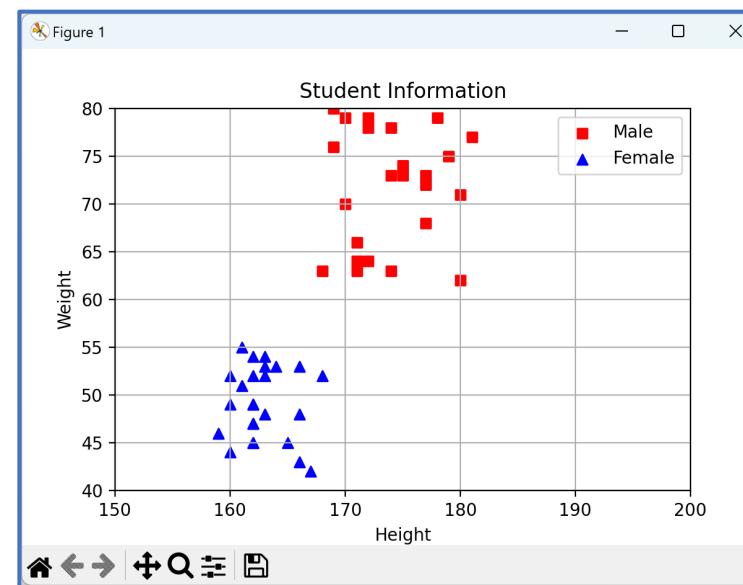
小提琴图案案例



思考

如何根据学生的性别分类绘制不同颜色图形？

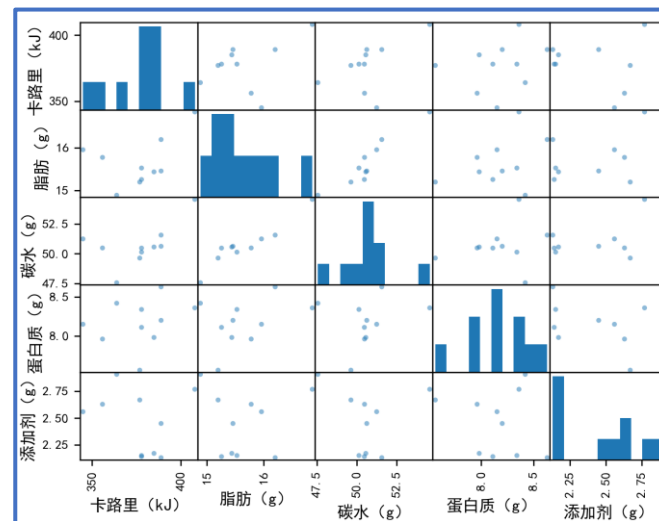
	A	B	C	D	E
1	No.	Gender	Age	Height	Weight
2	1	male	20	170	70
3	2	male	22	180	71
4	3	male	21	180	62
5	4	male	20	177	72
6	5	male	20	172	64
7	6	male	20	179	75
8	7	female	21	166	53
9	8	female	20	162	47
10	9	female	20	162	47
11	10	male	19	169	76
12	11	female	21	162	49



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，类似于二维数组，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

索引	行	列名	数据
	0	crim	0.00632
	1	medv	24.0
	2	crim	0.02731
	3	medv	21.6
	4	crim	0.02729
	5	medv	34.7
	6	crim	0.03237
	7	medv	33.4
	8	crim	0.06905
	9	medv	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5行数据

print("*****")

print(data. tail()) #打印后5行数据
```

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
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	<code>Obj.iloc[iloc, cloc]</code>	选取某行某列
	<code>Obj.iloc[ilocList, clocList]</code>	选取多行多列
	<code>Obj.iloc[a:b, c:d]</code>	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

	Height	Weight
No.		
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"]])
```

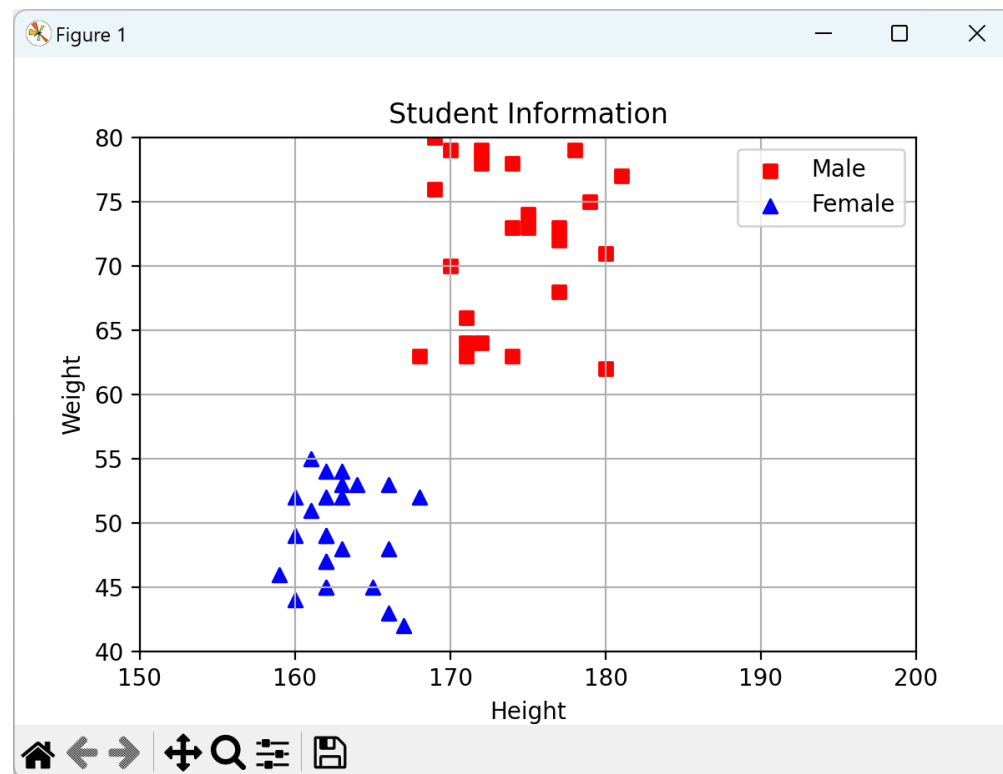
	Height	Weight
No.		
202201	170	70
202202	180	71

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

```
data2= data[data['Gender'] == 'female'] #筛选出女生
```

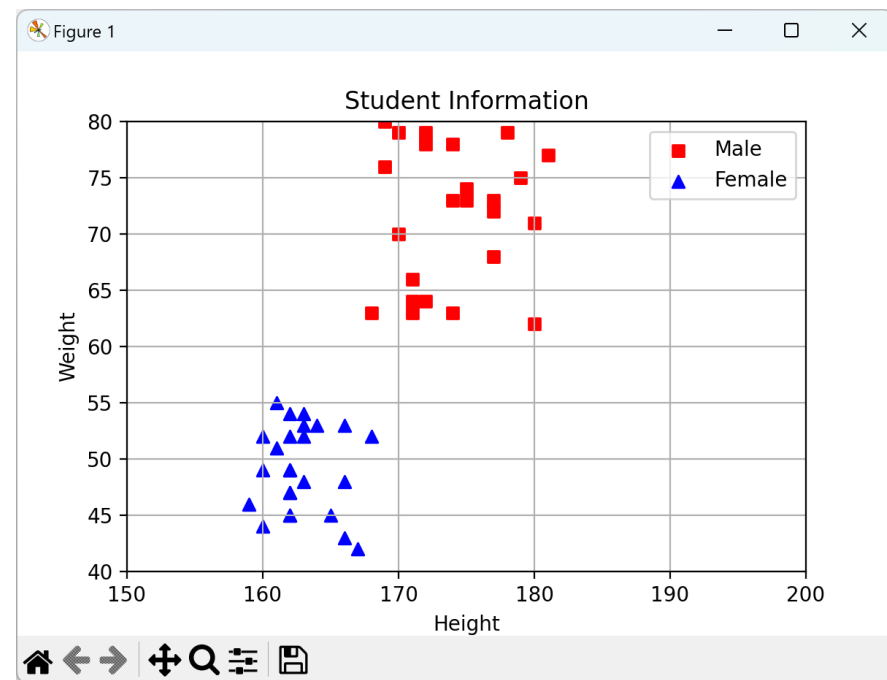
```
#分组绘制男生、女生的散点图
```

```
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

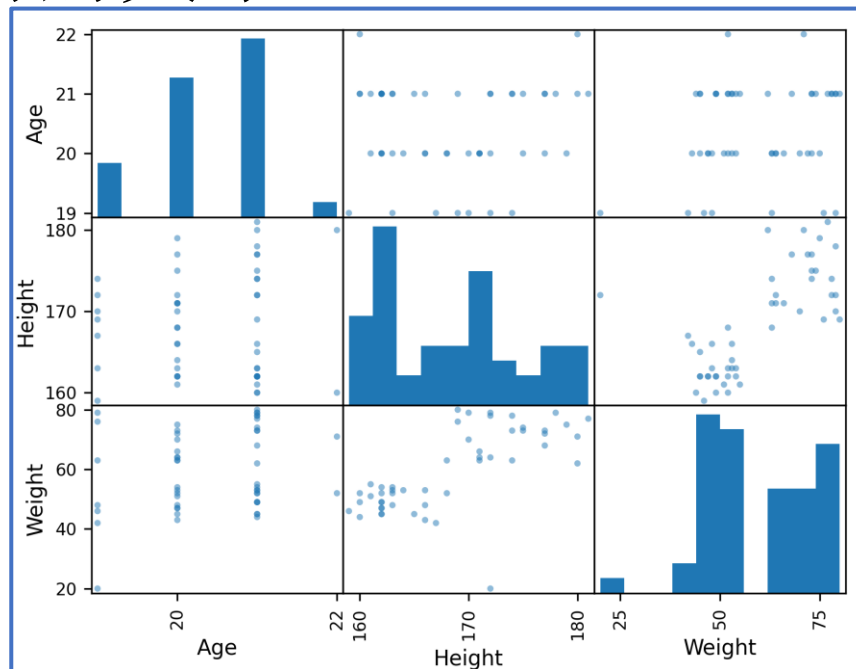
男女生信息统计

```
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('Height')         #x轴标题  
plt.ylabel('Weight')         #y轴标题  
plt.grid()                   #网格线  
plt.legend(loc='upper right') #图例显示位置  
plt.show()
```



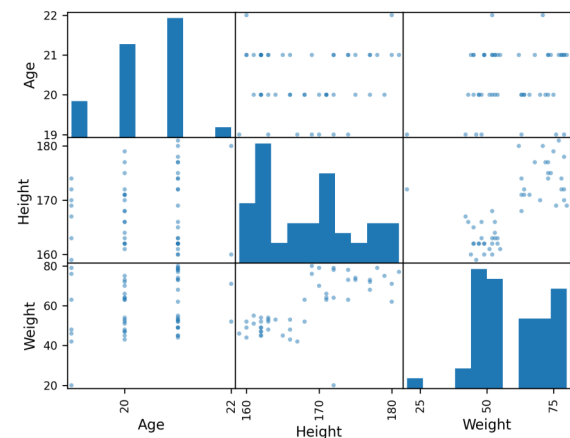
多项信息图形绘制

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。



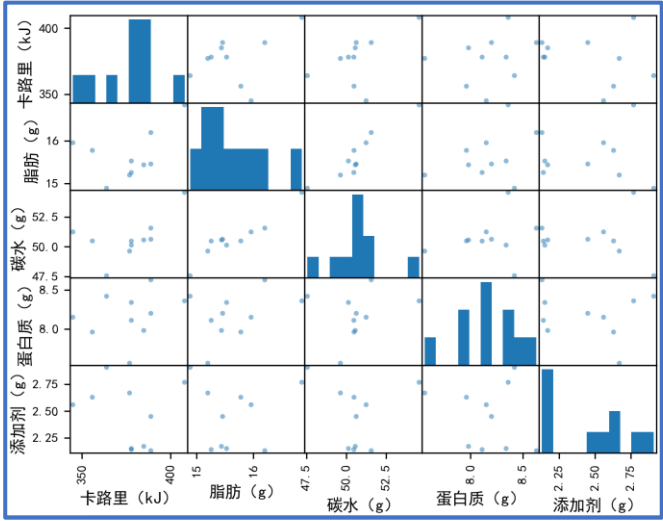
男女生信息统计

```
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()
```



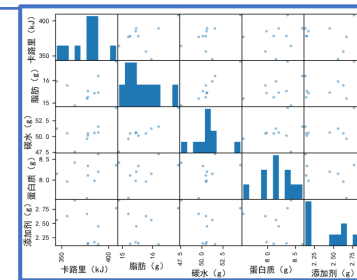
思考

	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()
```



Pandas读Excel文档

>>> pip install openpyxl #安装第三方库

```
import pandas as pd  
data=pd.read_excel("info.xlsx","Group1",index_col=0)  
print(data)
```

	A	B	C	D	E	F	G	H
1	序号	性别	年龄	身高	体重	省份	成绩	月生活费
2	21	female	21	165	45	Shanghai	93	1200
3	22	female	19	167	42	HuBei	89	800
4	23	male	21	169	80	GanSu	93	900
5	24	female	21	160	49	HeBei	59	1100
6	25	female	21	162	54	GanSu	68	1300
7	26	male	21	181	77	SiChuan	62	800
8	27	female	21	162	49	ShanDong	65	950
9	28	female	22	160	52	ShanXi	73	800
10	29	female	20	161	51	GuangXi	80	1250
11	30	female	20	168	52	JiangSu	98	700

序号	性别	年龄	身高	体重	省份	成绩	月生活费
21	female	21	165	45	Shanghai	93	1200
22	female	19	167	42	HuBei	89	800
23	male	21	169	80	GanSu	93	900
24	female	21	160	49	HeBei	59	1100
25	female	21	162	54	GanSu	68	1300
26	male	21	181	77	SiChuan	62	800
27	female	21	162	49	ShanDong	65	950
28	female	22	160	52	ShanXi	73	800
29	female	20	161	51	GuangXi	80	1250
30	female	20	168	52	JiangSu	98	700

Pandas常用统计函数

函数	描述
<code>df.mean()</code>	计算样本数据的算术平均值
<code>df.value_counts()</code>	统计频数
<code>df.describe()</code>	返回基本统计量和分位数
<code>df.corr(sr)</code>	df与sr的相关系数
<code>df.count()</code> 、 <code>df.sum()</code>	统计每列(或行)数据的个数或总和
<code>df.max()</code> 、 <code>df.min()</code>	最大值和最小值
<code>df.idxmax()</code> 、 <code>df.idxmin()</code>	最大值、最小值对应的索引
<code>df.qantile()</code>	计算给定的四分位数
<code>df.var()</code> 、 <code>df.std()</code>	计算方差、标准差
<code>df.mode()</code>	计算众数
<code>df.cov()</code>	计算协方差矩阵

Pandas常用统计案例

```
import pandas as pd  
data=pd.read_excel("info.xlsx","Group1",index_col=0)  
result=data.describe() #对数据进行统计描述  
print(result)
```

	A	B	C	D	E	F	G	H
1	序号	性别	年龄	身高	体重	省份	成绩	月生活费
2	21	female	21	165	45	Shanghai	93	1200
3	22	female	19	167	42	HuBei	89	800
4	23	male	21	169	80	GanSu	93	900
5	24	female	21	160	49	HeBei	59	1100
6	25	female	21	162	54	GanSu	68	1300
7	26	male	21	181	77	SiChuan	62	800
8	27	female	21	162	49	ShanDong	65	950
9	28	female	22	160	52	ShanXi	73	800
10	29	female	20	161	51	GuangXi	80	1250
11	30	female	20	168	52	JiangSu	98	700

	年龄	身高	体重	成绩	月生活费
count	10.000000	10.000000	10.0000	10.000000	10.000000
mean	20.700000	165.500000	55.1000	78.000000	980.000000
std	0.823273	6.381397	12.8448	14.476034	216.281709
min	19.000000	160.000000	42.0000	59.000000	700.000000
25%	20.250000	161.250000	49.0000	65.750000	800.000000
50%	21.000000	163.500000	51.5000	76.500000	925.000000
75%	21.000000	167.750000	53.5000	92.000000	1175.000000
max	22.000000	181.000000	80.0000	98.000000	1300.000000

Pandas常用统计函数

函数	描述
<code>df.mean()</code>	计算样本数据的算术平均值
<code>df.value_counts()</code>	统计频数
<code>df.describe()</code>	返回基本统计量和分位数
<code>df.corr(sr)</code>	df与sr的相关系数
<code>df.count()</code> 、 <code>df.sum()</code>	统计每列(或行)数据的个数或总和
<code>df.max()</code> 、 <code>df.min()</code>	最大值和最小值
<code>df.idxmax()</code> 、 <code>df.idxmin()</code>	最大值、最小值对应的索引
<code>df.qantile()</code>	计算给定的四分位数
<code>df.var()</code> 、 <code>df.std()</code>	计算方差、标准差
<code>df.mode()</code>	计算众数
<code>df.cov()</code>	计算协方差矩阵

Pandas常用统计案例

```
import pandas as pd  
data=pd.read_excel("info.xlsx","Group1",index_col=0)  
avg=data['成绩'].mean()  
print("成绩的平均值为: {}".format(avg))  
max_age=data['年龄'].max()  
print("年龄的最大值为: {}".format(max_age))
```

	A	B	C	D	E	F	G	H
1	序号	性别	年龄	身高	体重	省份	成绩	月生活费
2	21	female	21	165	45	Shanghai	93	1200
3	22	female	19	167	42	HuBei	89	800
4	23	male	21	169	80	GanSu	93	900
5	24	female	21	160	49	HeBei	59	1100
6	25	female	21	162	54	GanSu	68	1300
7	26	male	21	181	77	SiChuan	62	800
8	27	female	21	162	49	ShanDong	65	950
9	28	female	22	160	52	ShanXi	73	800
10	29	female	20	161	51	GuangXi	80	1250
11	30	female	20	168	52	JiangSu	98	700

成绩的平均值为: 78.0
年龄的最大值为: 22

Pandas常用统计函数

函数	描述
<code>df.mean()</code>	计算样本数据的算术平均值
<code>df.value_counts()</code>	统计频数
<code>df.describe()</code>	返回基本统计量和分位数
<code>df.corr(sr)</code>	df与sr的相关系数
<code>df.count()</code> 、 <code>df.sum()</code>	统计每列(或行)数据的个数或总和
<code>df.max()</code> 、 <code>df.min()</code>	最大值和最小值
<code>df.idxmax()</code> 、 <code>df.idxmin()</code>	最大值、最小值对应的索引
<code>df.qantile()</code>	计算给定的四分位数
<code>df.var()</code> 、 <code>df.std()</code>	计算方差、标准差
<code>df.mode()</code>	计算众数
<code>df.cov()</code>	计算协方差矩阵

Pandas常用统计案例

```
import pandas as pd  
data=pd.read_excel("info.xlsx","Group1",index_col=0)  
score=data["成绩"].sum()  
print("学生的总成绩为: {}".format(score))  
age=data["年龄"].mode()  
print("学生多数年龄为: {}".format(age))
```

	A	B	C	D	E	F	G	H
1	序号	性别	年龄	身高	体重	省份	成绩	月生活费
2	21	female	21	165	45	Shanghai	93	1200
3	22	female	19	167	42	HuBei	89	800
4	23	male	21	169	80	GanSu	93	900
5	24	female	21	160	49	HeBei	59	1100
6	25	female	21	162	54	GanSu	68	1300
7	26	male	21	181	77	SiChuan	62	800
8	27	female	21	162	49	ShanDong	65	950
9	28	female	22	160	52	ShanXi	73	800
10	29	female	20	161	51	GuangXi	80	1250
11	30	female	20	168	52	JiangSu	98	700

学生的总成绩为: 780
学生多数年龄为: 0 21
Name: 年龄, dtype: int64

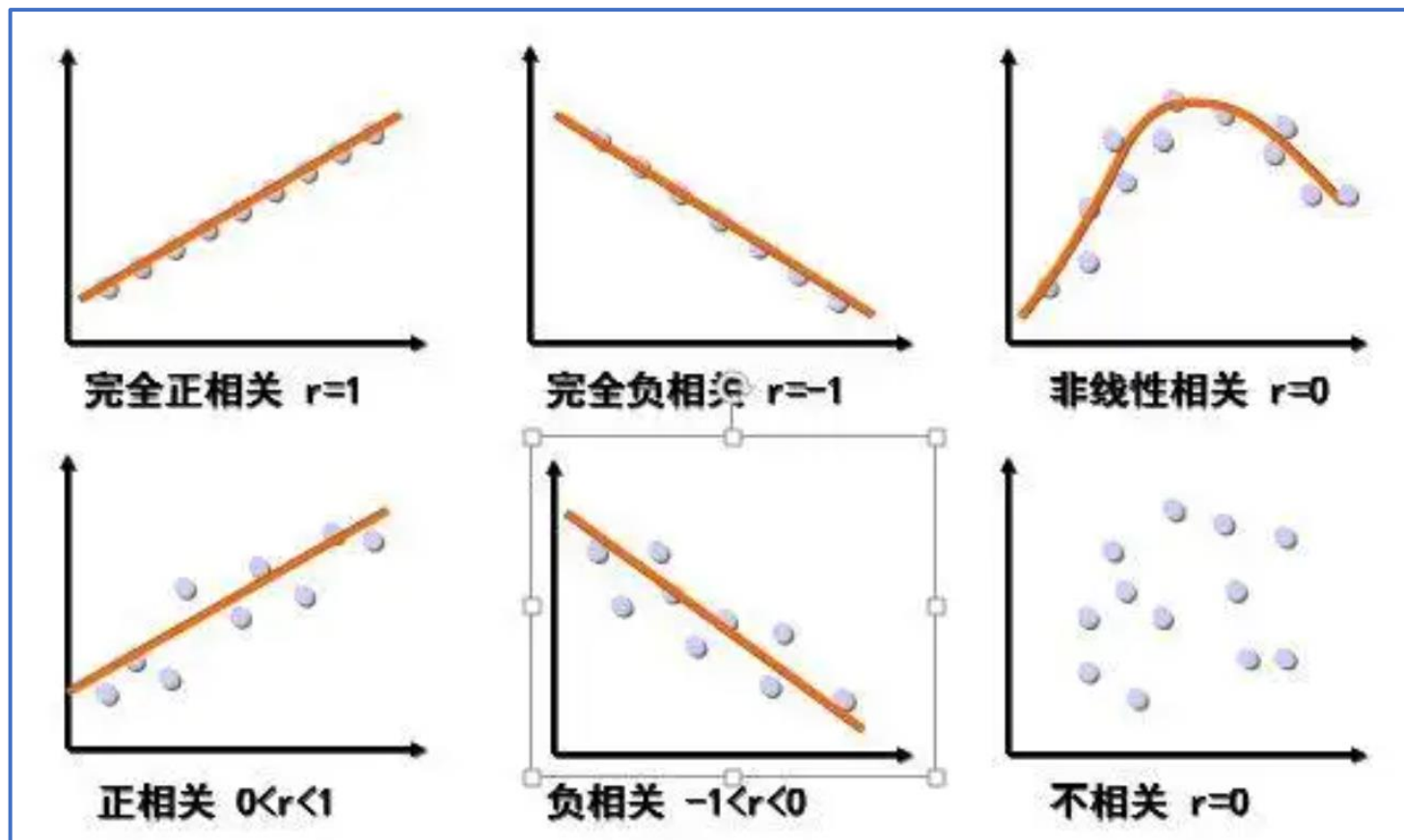
相关性分析

相关性分析：研究现象之间是否存在依赖关系，定量分析可以

通过计算样本之间的相关系数 r 来实现， r 具有以下特征：

1. r 的值介于-1和+1之间， $r=1$ 表示正相关， $r=0$ 表示不相关， $r=-1$ 表示负相关
2. 当 $0<|r|<1$ ，表示两个对象存在一定程度的相关性， $|r|$ 越接近1，关系越密切，越接近0，相关性越弱
3. $|r|<0.4$ 为低相关； $0.4\leq|r|<0.7$ 为中等相关， $|r|\geq 0.7$ 为高相关

相关性分析



Pandas常用统计函数

函数	描述
<code>df.mean()</code>	计算样本数据的算术平均值
<code>df.value_counts()</code>	统计频数
<code>df.describe()</code>	返回基本统计量和分位数
<code>df.corr(sr)</code>	<code>df</code> 与 <code>sr</code> 的相关系数
<code>df.count()</code> 、 <code>df.sum()</code>	统计每列(或行)数据的个数或总和
<code>df.max()</code> 、 <code>df.min()</code>	最大值和最小值
<code>df.idxmax()</code> 、 <code>df.idxmin()</code>	最大值、最小值对应的索引
<code>df.qantile()</code>	计算给定的四分位数
<code>df.var()</code> 、 <code>df.std()</code>	计算方差、标准差
<code>df.mode()</code>	计算众数
<code>df.cov()</code>	计算协方差矩阵

Pandas常用统计案例

```
import pandas as pd  
data=pd.read_excel("info.xlsx","Group1",index_col=0)  
result=data['身高'].corr( data['体重'] )  
print("身高和体重的相关性为： {}".format(result))
```

	A	B	C	D	E	F	G	H
1	序号	性别	年龄	身高	体重	省份	成绩	月生活费
2	21	female	21	165	45	Shanghai	93	1200
3	22	female	19	167	42	HuBei	89	800
4	23	male	21	169	80	GanSu	93	900
5	24	female	21	160	49	HeBei	59	1100
6	25	female	21	162	54	GanSu	68	1300
7	26	male	21	181	77	SiChuan	62	800
8	27	female	21	162	49	ShanDong	65	950
9	28	female	22	160	52	ShanXi	73	800
10	29	female	20	161	51	GuangXi	80	1250
11	30	female	20	168	52	JiangSu	98	700

身高和体重的相关性为： 0.6757399098527682

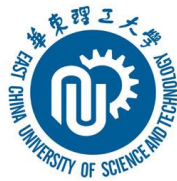
Pandas常用统计案例

```
import pandas as pd
data=pd.read_excel("info.xlsx","Group1",index_col=0)
result=data[['身高','体重','成绩']].corr()
print(result)
```

如果想画图显示三者关系如何处理？

	A	B	C	D	E	F	G	H
1	序号	性别	年龄	身高	体重	省份	成绩	月生活费
2	21	female	21	165	45	Shanghai	93	1200
3	22	female	19	167	42	HuBei	89	800
4	23	male	21	169	80	GanSu	93	900
5	24	female	21	160	49	HeBei	59	1100
6	25	female	21	162	54	GanSu	68	1300
7	26	male	21	181	77	SiChuan	62	800
8	27	female	21	162	49	ShanDong	65	950
9	28	female	22	160	52	ShanXi	73	800
10	29	female	20	161	51	GuangXi	80	1250
11	30	female	20	168	52	JiangSu	98	700

	身高	体重	成绩
身高	1.000000	0.675740	0.080587
体重	0.675740	1.000000	-0.072305
成绩	0.080587	-0.072305	1.000000



谢 谢