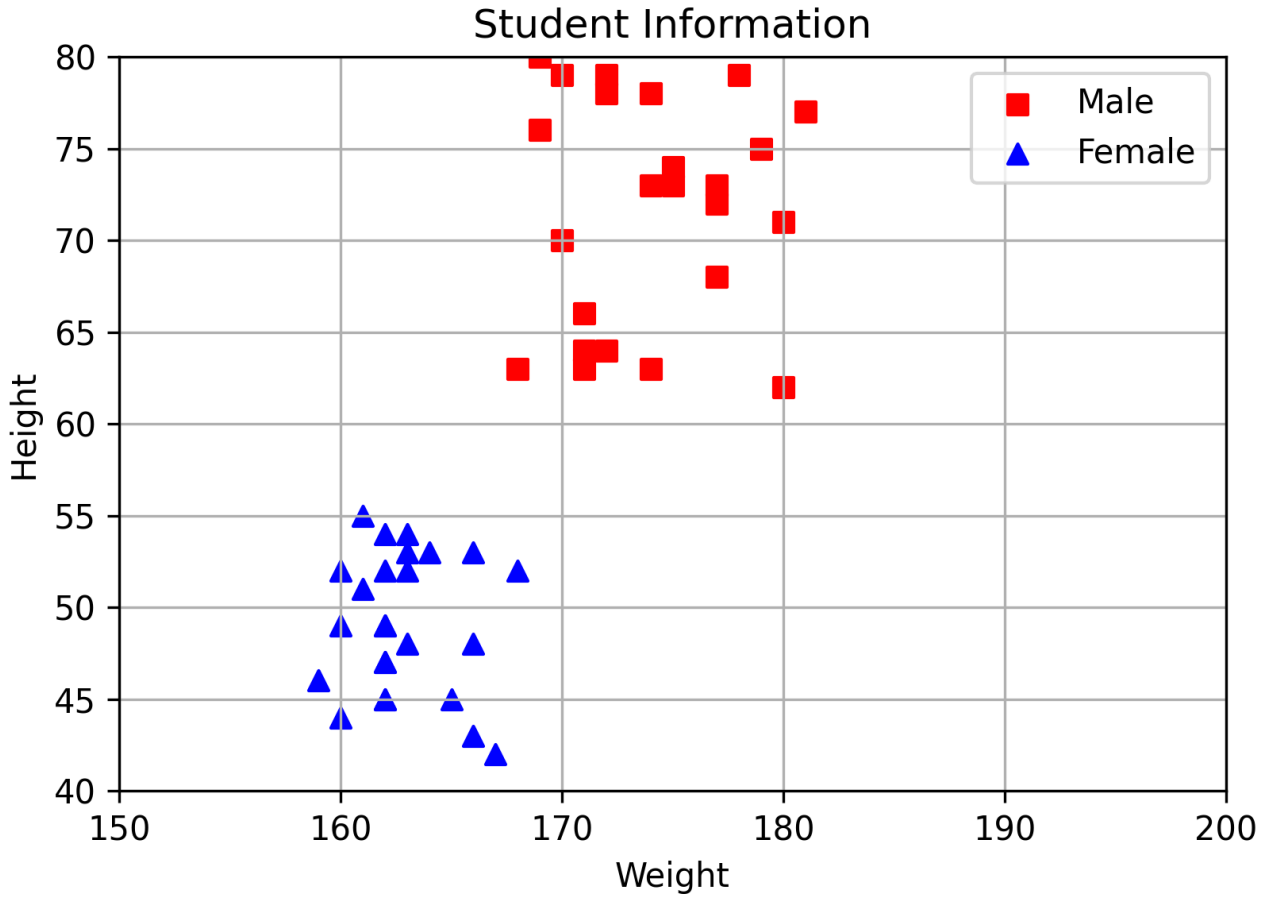
Python与金融数据挖掘(13)

文欣秀

wenxinxiu@ecust.edu.cn

分类图表绘制


Student Information
Male
Female
Height
Weight
80
75
70
65
60
55
50
45
40
150
160
170
180
190
200

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…

…

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

存取部分数据

```
import pandas as pd

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

print(data. iloc[7,1])

print(data. iloc[[0,2],[1,2]])

result=data.iloc[0:3,1:3]

result.to_csv("result.csv")
```

	A	B	C
1	date	score	price
2	2018/9/3	70	23.55
3	2018/9/4	75	24.43
4	2018/9/5	65	23.41
5	2018/9/6	60	22.81
6	2018/9/7	70	23.21
7	2018/9/10	75	23.46
8	2018/9/11	75	23.34
9	2018/9/12	40	22.88
10	2018/9/13	60	23.1

```
40
    score  price
0      70  23.55
2      65  23.41
```

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

存取部分数据

>>> **pip install openpyxl #安装第三方库**

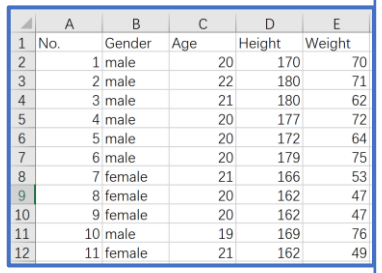
```
import pandas as pd
data=pd.read_excel("info.xlsx","Group1",index_col=0)
result=data.loc[[21,23],["身高","体重"]]
print(result)
result.to_excel("result.xlsx", columns=['身高','体重'])
```

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

男女生信息统计

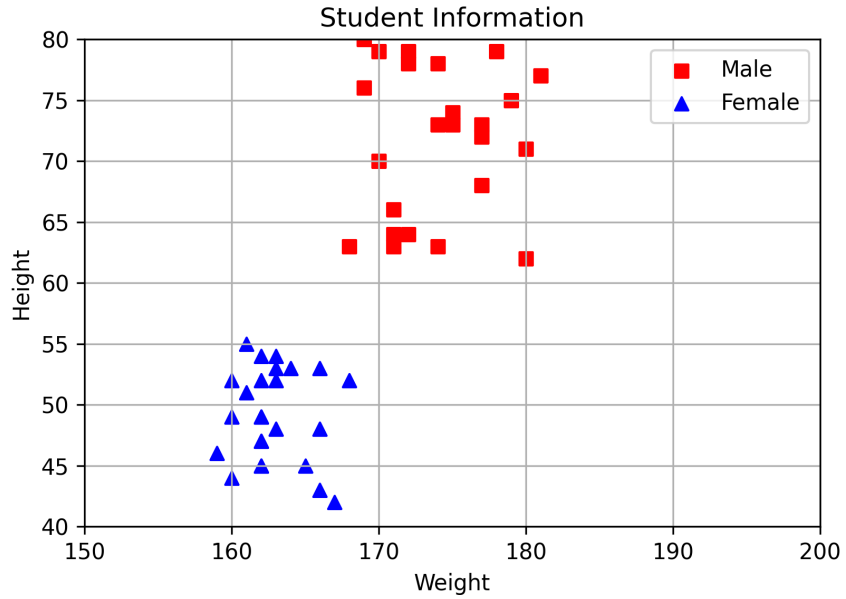
```
import matplotlib.pyplot as plt   #导入matplotlib.pyplot
import pandas as pd
#绘制散点图观察学生身高和体重之间的关系。
data = pd. read_csv('student.csv', index_col=0)
#将数据按性别分组，分别绘制散点图
data1= data. loc[data['Gender'] == 'male'] #筛选出男生
data2= data. loc[data['Gender'] == 'female'] #筛选出女生
#分组绘制男生、女生的散点图
plt. figure(figsize=(6,4))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



	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()
```



Pandas常用统计函数

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

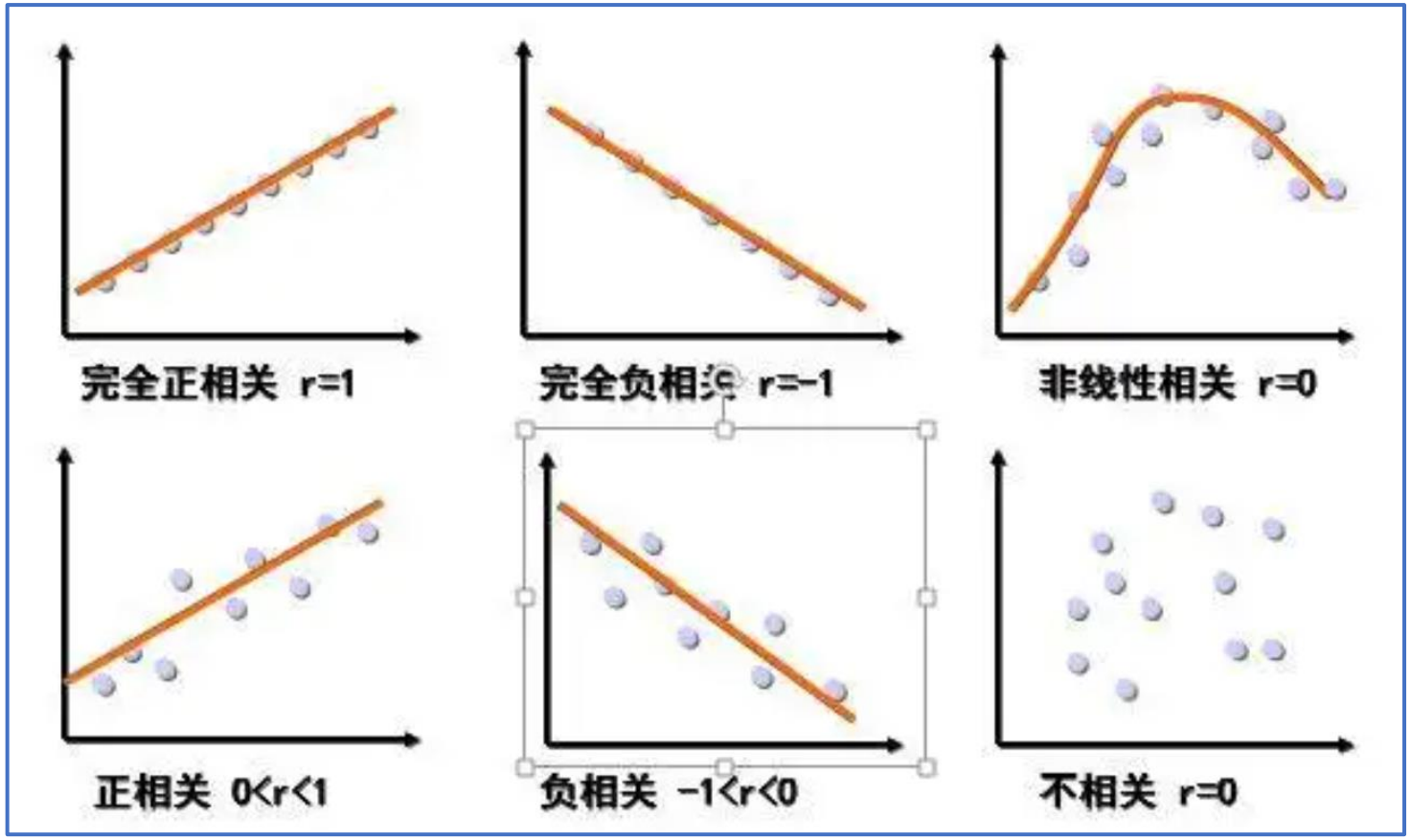
```
学生的总成绩为： 780
学生多数年龄为： 0    21
Name: 年龄, dtype: int64
```

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=<|r|<0.7为中等相关，|r|>=0.7为高相关

相关性分析



Pandas常用统计函数

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

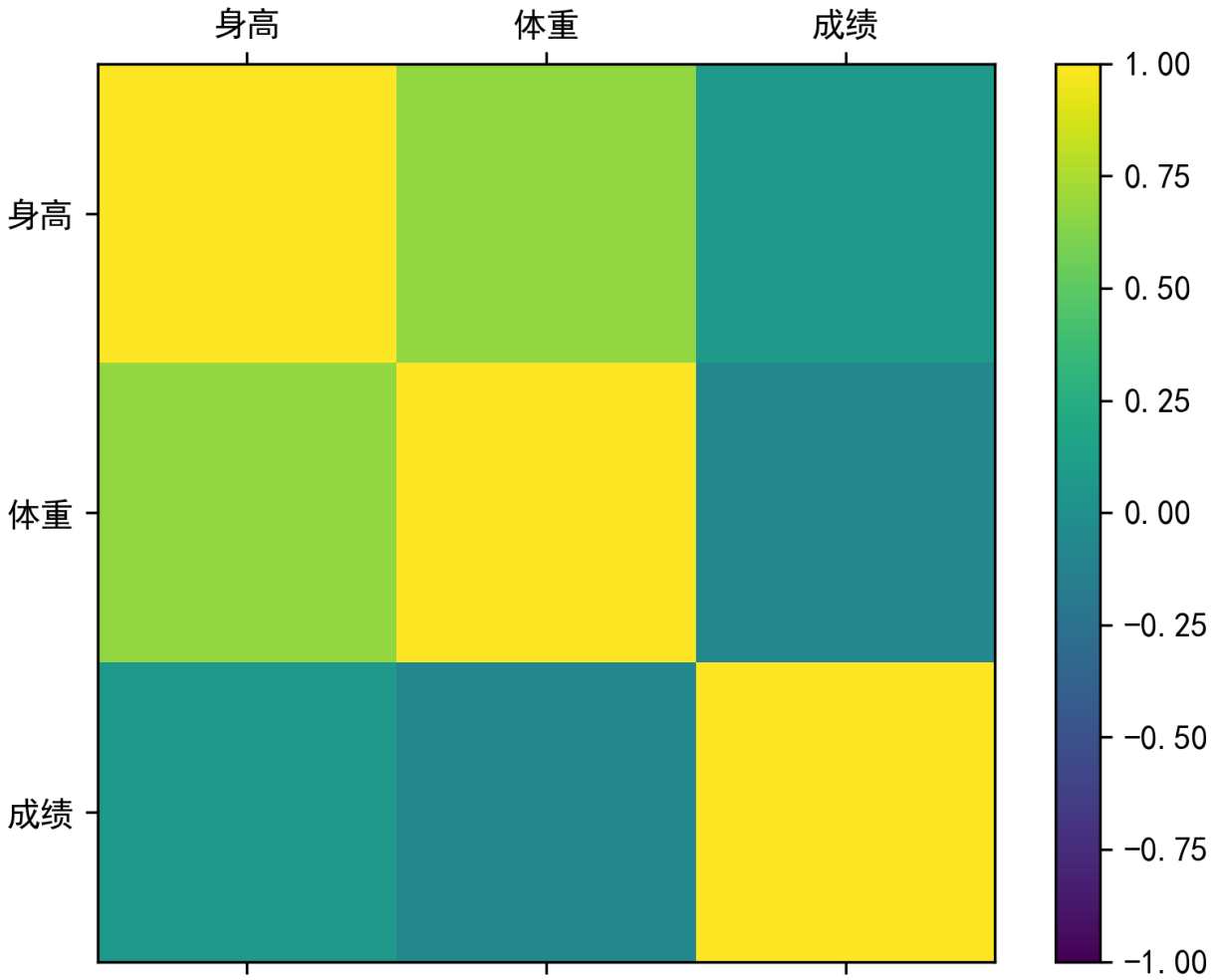
```
import pandas as pd
data=pd.read_excel("info.xlsx","Group1",index_col=0)
result=data[['身高','体重','成绩'] ].corr()
print(result)
```

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

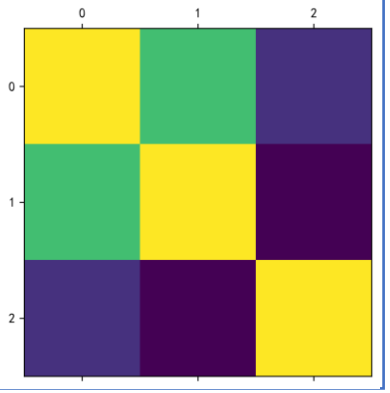
	身高	体重	成绩
身高	1.000000	0.675740	0.080587
体重	0.675740	1.000000	-0.072305
成绩	0.080587	-0.072305	1.000000

Matshow图



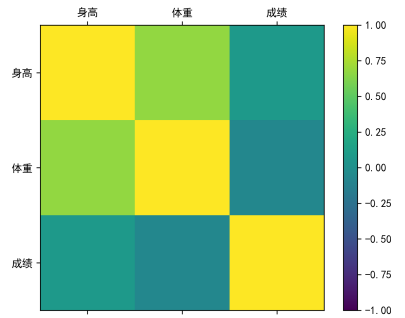
Pandas统计分析案例

```
import matplotlib.pyplot as plt   #导入matplotlib.pyplot
import pandas as pd
plt.rcParams['font.family']=['SimHei']
data=pd.read_excel("info.xlsx","Group1",index_col=0)
result=data[['身高','体重','成绩'] ].corr()
plt.matshow(result)  #相关矩阵图展示两个不同属性相互影响的程度
plt.show()
```



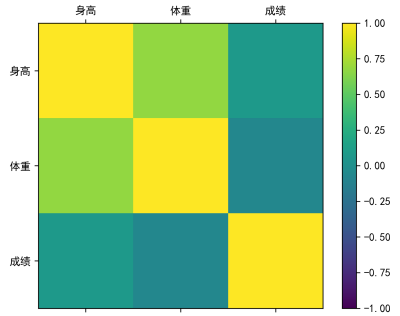
Pandas统计分析案例（拓展）

```
import matplotlib.pyplot as plt   #导入matplotlib.pyplot
import pandas as pd
import numpy as np
plt.rcParams['font.family']=['SimHei']          #显示中文
plt.rcParams['axes.unicode_minus'] = False    #显示负号
data=pd.read_excel("info.xlsx","Group1",index_col=0)
result=data[['身高','体重','成绩'] ].corr()
```

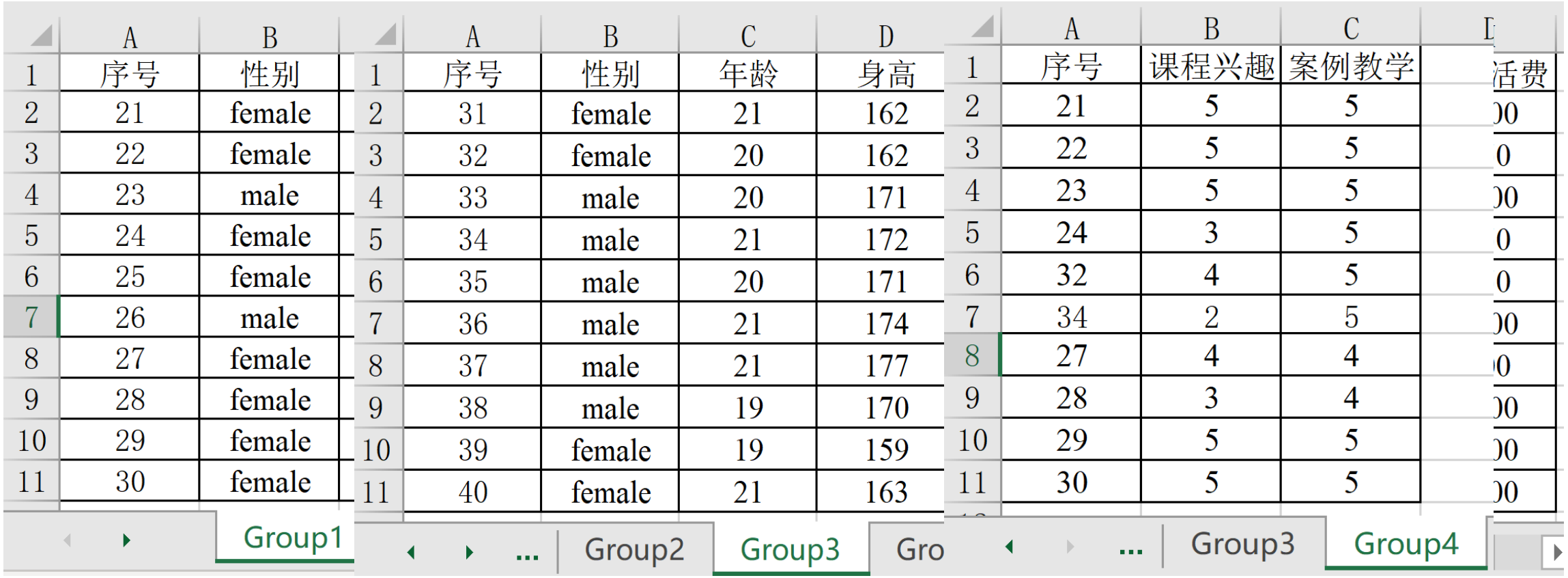


Pandas统计分析案例（拓展）

```
fig=plt.figure()
ax=fig.add_subplot(111)
cax=ax.matshow(result, vmin=-1, vmax=1)    #相关矩阵图
fig.colorbar(cax)
ticks=np.arange(0,3,1)
names=['身高','体重','成绩']
ax.set_xticks(ticks); ax.set_yticks(ticks)
ax.set_xticklabels(names); ax.set_yticklabels(names)
plt.show()
```



数据规整化



	A	B
1	序号	性别
2	21	female
3	22	female
4	23	male
5	24	female
6	25	female
7	26	male
8	27	female
9	28	female
10	29	female
11	30	female

Group1

	A	B	C	D
1	序号	性别	年龄	身高
2	31	female	21	162
3	32	female	20	162
4	33	male	20	171
5	34	male	21	172
6	35	male	20	171
7	36	male	21	174
8	37	male	21	177
9	38	male	19	170
10	39	female	19	159
11	40	female	21	163

... Group2 Group3 Gro

	A	B	C	D
1	序号	课程兴趣	案例教学	活费
2	21	5	5	00
3	22	5	5	0
4	23	5	5	00
5	24	3	5	0
6	32	4	5	0
7	34	2	5	00
8	27	4	4	0
9	28	3	4	00
10	29	5	5	00
11	30	5	5	00

... Group3 Group4

数据规整化

➢**concat()：** 行数据连接函数

```
import pandas as pd
data1=pd.read_excel("info.xlsx","Group1",index_col=0)
data2=pd.read_excel("info.xlsx","Group3",index_col=0)
#axis=0表示按行追加
data = pd.concat([data1,data2], axis=0)
print(data)
```

数据规整化

- **merge(x,y,how,left_on,right_on…)**

  x：左数据对象

  y：右数据对象

  how：数据对象连接方式：inner, outer, left, right

  - inner：内连接，连接两个数据对象中键值交集的行
  - left:　左连接，取出x的全部行，连接y中匹配的键值行

  left_on：左数据用于连接的键

  right_on：右数据用于连接的键

数据规整化

➢**merge()：** 列数据连接函数

```
import pandas as pd
data1=pd.read_excel("info.xlsx","Group1",index_col=0)
data2=pd.read_excel("info.xlsx","Group4",index_col=0)
result= pd.merge(data1,data2, how='left',left_on="序号",right_on="序号")
print(result) #左连接
```

数据排序

- 按索引排序

```
import pandas as pd
data=pd.read_excel("info.xlsx","Group1",index_col=0)
#按行索引降序排序
result1=data.sort_index(ascending=False)
print(result1)
```

数据排序

- 按值排序

```
import pandas as pd
data=pd.read_excel("info.xlsx","Group1",index_col=0)
result2=data.sort_values(by='成绩', ascending=False)
print(result2)
result3=data.sort_values(by=['身高','体重'], ascending=True)
print(result3)
```

DataFrame数据排序

- 排名

```
import pandas as pd
data=pd.read_excel("info.xlsx","Group1",index_col=0)
#对成绩数据降序排名，增加“排名”列,method为并列名次取值
#比如（2，3名成绩相同，min去2,max取3）
data['排名'] = data['成绩'].rank(method='min', ascending=False)
print( data )
```

数据清洗

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

数据清洗

数据清洗：对采集的数据进行重新审查和校验的过程，其目的在于删除重复信息、纠正存在的错误，保证数据的一致性。

常见问题：

- 数据缺失
- 数据重复
- 数据不一致

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

数据清洗

丢弃缺失值dropna(axis,how,thresh,…)

axis： 0表示按行滤除，1表示按列滤除， 默认为axis=0

data. dropna()　　#每行只要有空值，就将该行删除

data. dropna(axis=1)　#每列只要有空值，就将该列删除

数据清洗案例

```
import pandas as pd
data=pd.read_excel("info.xlsx","Group2",index_col=0)
data1=data. dropna()   #默认按行删除
print(data1)
```

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

序号	性别	年龄	身高	体重	省份	成绩	月生活费
2.0	male	22.0	180.0	71.0	GuangXi	77.0	1300.0
4.0	male	20.0	177.0	72.0	LiaoNing	79.0	900.0
6.0	male	20.0	179.0	75.0	YunNan	92.0	950.0
7.0	female	21.0	166.0	53.0	LiaoNing	80.0	1200.0
8.0	female	20.0	162.0	47.0	AnHui	78.0	1000.0
9.0	female	20.0	162.0	47.0	AnHui	78.0	1000.0
10.0	male	120.0	169.0	76.0	HeiLongJiang	88.0	1100.0

数据清洗案例

```
import pandas as pd
data=pd. read_excel("info.xlsx","Group2",index_col=0)
data1=data. dropna(axis=1) #按列删除
print(data1)
```

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

```
Empty DataFrame
Columns: []
Index: [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, nan, 7.0, 8.0, 9.0, 10.0]
```

数据清洗

丢弃缺失值dropna(axis,how,thresh,…)

how: "all"表示滤除全部值都为NaN的行或列

data. dropna(how='all') #一行中全部为NaN才丢弃该行

数据清洗案例

```
import pandas as pd
data=pd.read_excel("info.xlsx","Group2",index_col=0)
data1=data. dropna(how="all")  #一行全部为NaN才删
print(data1)
```

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

```
      性别   年龄   身高   体重        省份   成绩   月生活费
序号
1.0   male  20.0  170.0  70.0   LiaoNing   NaN  800.0
2.0   male  22.0  180.0  71.0   GuangXi  77.0 1300.0
3.0   male   NaN  180.0  62.0    FuJian  57.0 1000.0
4.0   male  20.0  177.0  72.0   LiaoNing  79.0  900.0
5.0   male  20.0  172.0   NaN   ShanDong  91.0   NaN
6.0   male  20.0  179.0  75.0    YunNan  92.0  950.0
7.0 female  21.0  166.0  53.0   LiaoNing  80.0 1200.0
8.0 female  20.0  162.0  47.0     AnHui  78.0 1000.0
9.0 female  20.0  162.0  47.0     AnHui  78.0 1000.0
10.0  male 120.0  169.0  76.0 HeiLongJiang  88.0 1100.0
```

数据清洗

丢弃缺失值dropna(axis,how,thresh,…)

thresh： 只留下有效数据数大于或等于thresh的行或列

```
data. dropna(thresh=6)      # 每行至少6个非空值才保留
```

数据清洗案例

```
import pandas as pd
data=pd.read_excel("info.xlsx","Group2",index_col=0)
data1=data. dropna(thresh=6)  # 每行至少6个非空值才保留
print(data1)
```

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

```
        性别    年龄    身高    体重        省份    成绩    月生活费
序号
1.0     male   20.0  170.0  70.0    LiaoNing   NaN   800.0
2.0     male   22.0  180.0  71.0    GuangXi  77.0  1300.0
3.0     male    NaN  180.0  62.0      FuJian  57.0  1000.0
4.0     male   20.0  177.0  72.0    LiaoNing  79.0   900.0
6.0     male   20.0  179.0  75.0      YunNan  92.0   950.0
7.0   female   21.0  166.0  53.0    LiaoNing  80.0  1200.0
8.0   female   20.0  162.0  47.0      AnHui  78.0  1000.0
9.0   female   20.0  162.0  47.0      AnHui  78.0  1000.0
10.0    male  120.0  169.0  76.0 HeiLongJiang  88.0  1100.0
```

数据清洗

缺失值填充fillna(value, method,…)

value： 填充值，可以是标量、字典等

data.fillna(0)　**#用0填充**

数据清洗案例

```
import pandas as pd
data=pd. read_excel("info.xlsx","Group2",index_col=0)
data1=data. fillna(0)   #用0填充
print(data1)
```

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

```
      性别   年龄   身高   体重         省份   成绩   月生活费
序号
1.0   male  20.0  170.0  70.0     LiaoNing  0.0   800.0
2.0   male  22.0  180.0  71.0     GuangXi  77.0  1300.0
3.0   male   0.0  180.0  62.0      FuJian  57.0  1000.0
4.0   male  20.0  177.0  72.0     LiaoNing  79.0   900.0
5.0   male  20.0  172.0   0.0    ShanDong  91.0     0.0
6.0   male  20.0  179.0  75.0      YunNan  92.0   950.0
NaN      0   0.0    0.0   0.0           0   0.0     0.0
7.0 female  21.0  166.0  53.0     LiaoNing  80.0  1200.0
8.0 female  20.0  162.0  47.0       AnHui  78.0  1000.0
9.0 female  20.0  162.0  47.0       AnHui  78.0  1000.0
10.0  male 120.0  169.0  76.0  HeiLongJiang  88.0  1100.0
```

数据清洗

缺失值填充fillna(value, method,…)

value： 填充值，可以是标量、字典等

data. fillna({'年龄': data['年龄'].mean(), '性别': 'male'})

数据清洗案例

```
import pandas as pd
data=pd.read_excel("info.xlsx","Group2",index_col=0)
data1=data. fillna({'年龄': data['年龄'].mean(), '性别': 'male'})
print(data1)
```

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

序号	性别	年龄	身高	体重	省份	成绩	月生活费
1.0	male	20.000000	170.0	70.0	LiaoNing	NaN	800.0
2.0	male	22.000000	180.0	71.0	GuangXi	77.0	1300.0
3.0	male	31.444444	180.0	62.0	FuJian	57.0	1000.0
4.0	male	20.000000	177.0	72.0	LiaoNing	79.0	900.0
5.0	male	20.000000	172.0	NaN	ShanDong	91.0	NaN
6.0	male	20.000000	179.0	75.0	YunNan	92.0	950.0
NaN	male	31.444444	NaN	NaN	NaN	NaN	NaN
7.0	female	21.000000	166.0	53.0	LiaoNing	80.0	1200.0
8.0	female	20.000000	162.0	47.0	AnHui	78.0	1000.0
9.0	female	20.000000	162.0	47.0	AnHui	78.0	1000.0
10.0	male	120.000000	169.0	76.0	HeiLongJiang	88.0	1100.0

数据清洗

缺失值填充fillna(value, method,…)

value： 填充值，可以是标量、字典等

method： 'ffill'，'bfill' 用同列前一行或后一行数据填充

data. fillna(method='ffill') #在列方向上以前一个值替换

数据清洗案例

```
import pandas as pd
data=pd. read_excel("info.xlsx","Group2",index_col=0)
data1=data. fillna(method='ffill') #在列方向上以前一个值替换
print(data1)
```

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

```
      性别   年龄   身高   体重        省份   成绩   月生活费
序号
1.0   male  20.0  170.0  70.0   LiaoNing   NaN  800.0
2.0   male  22.0  180.0  71.0   GuangXi  77.0 1300.0
3.0   male  22.0  180.0  62.0    FuJian  57.0 1000.0
4.0   male  20.0  177.0  72.0   LiaoNing  79.0  900.0
5.0   male  20.0  172.0  72.0   ShanDong  91.0  900.0
6.0   male  20.0  179.0  75.0    YunNan  92.0  950.0
NaN   male  20.0  179.0  75.0    YunNan  92.0  950.0
7.0 female  21.0  166.0  53.0   LiaoNing  80.0 1200.0
8.0 female  20.0  162.0  47.0     AnHui  78.0 1000.0
9.0 female  20.0  162.0  47.0     AnHui  78.0 1000.0
10.0  male 120.0  169.0  76.0 HeiLongJiang 88.0 1100.0
```

数据清洗

值替换replace(to_replace, value, …)

to_replace： 将被替代的值

value: 替换为的值

data['年龄'].replace(120, 20)#将年龄120替换为20

数据清洗案例

```
import pandas as pd
data=pd. read_excel("info.xlsx","Group2",index_col=0)
data['年龄'].replace(120, 20,inplace=True) #将年龄120替换为20
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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

```
      性别   年龄    身高   体重        省份   成绩   月生活费
序号
1.0   male  20.0  170.0  70.0    LiaoNing   NaN   800.0
2.0   male  22.0  180.0  71.0     GuangXi  77.0  1300.0
3.0   male   NaN  180.0  62.0      FuJian  57.0  1000.0
4.0   male  20.0  177.0  72.0    LiaoNing  79.0   900.0
5.0   male  20.0  172.0   NaN    ShanDong  91.0     NaN
6.0   male  20.0  179.0  75.0      YunNan  92.0   950.0
NaN    NaN   NaN    NaN   NaN         NaN   NaN     NaN
7.0 female  21.0  166.0  53.0    LiaoNing  80.0  1200.0
8.0 female  20.0  162.0  47.0      AnHui  78.0  1000.0
9.0 female  20.0  162.0  47.0      AnHui  78.0  1000.0
10.0  male  20.0  169.0  76.0 HeiLongJiang  88.0  1100.0
```

数据清洗

去掉重复值drop_duplicates()

data. drop_duplicates() #去掉重复的数据

数据清洗案例

```
import pandas as pd
data=pd. read_excel("info.xlsx","Group2",index_col=0)
data1=data. drop_duplicates()  #去掉重复的数据
print(data1)
```

	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	1	male	20	170	70	LiaoNing		800
3	2	male	22	180	71	GuangXi	77	1300
4	3	male		180	62	FuJian	57	1000
5	4	male	20	177	72	LiaoNing	79	900
6	5	male	20	172		ShanDong	91	
7	6	male	20	179	75	YunNan	92	950
8								
9	7	female	21	166	53	LiaoNing	80	1200
10	8	female	20	162	47	AnHui	78	1000
11	9	female	20	162	47	AnHui	78	1000
12	10	male	120	169	76	HeiLongJiang	88	1100

```
      性别   年龄   身高   体重       省份   成绩   月生活费
序号
1.0   male  20.0  170.0  70.0   LiaoNing   NaN   800.0
2.0   male  22.0  180.0  71.0   GuangXi  77.0  1300.0
3.0   male   NaN  180.0  62.0     FuJian  57.0  1000.0
4.0   male  20.0  177.0  72.0   LiaoNing  79.0   900.0
5.0   male  20.0  172.0   NaN   ShanDong  91.0     NaN
6.0   male  20.0  179.0  75.0     YunNan  92.0   950.0
NaN    NaN   NaN   NaN   NaN        NaN   NaN     NaN
7.0 female  21.0  166.0  53.0   LiaoNing  80.0  1200.0
8.0 female  20.0  162.0  47.0     AnHui  78.0  1000.0
10.0  male 120.0  169.0  76.0 HeiLongJiang  88.0  1100.0
```

谢　谢