```
   0  1   2   3    4
0  0  2   4  13.0  14.0
1  8  10  12  18.0  19.0
```

In [58]:
```
df6>5
```

Out[58]:

	0	1	2	3	4
0	False	False	False	True	True
1	True	True	True	True	True

In [59]:
```
print(s1)
```

```
0   0
1   1
2   2
dtype: int32
```

In [60]:
```
df6>s1
```

Out[60]:

	0	1	2	3	4
0	False	True	True	False	False
1	True	True	True	False	False

In [61]:
```
df6>(2,18)
```

Out[61]:

	0	1	2	3	4
0	False	False	True	True	True
1	False	False	False	False	True

38.7 统计信息

In [62]:
```
import numpy as np
import pandas as pd
df2 = pd.read_csv('bc_data.csv')
df2=df2[["id","diagnosis","area_mean"]]
df2.describe()
```

Out[62]:

	id	area_mean
count	5.690000e+02	569.000000
mean	3.037183e+07	654.889104
std	1.250206e+08	351.914129
min	8.670000e+03	143.500000
25%	8.692180e+05	420.300000
50%	9.060240e+05	551.100000

75%	8.813129e+06	782.700000
max	9.113205e+08	2501.000000

In [63]:
```
dt = df2[df2.diagnosis=='M']
```

In [64]:
```
dt.head()
```

Out[64]:

	id	diagnosis	area_mean
0	842302	M	1001.0
1	842517	M	1326.0
2	84300903	M	1203.0
3	84348301	M	386.1
4	84358402	M	1297.0

In [65]:
```
dt.tail()
```

Out[65]:

	id	diagnosis	area_mean
563	926125	M	1347.0
564	926424	M	1479.0
565	926682	M	1261.0
566	926954	M	858.1
567	927241	M	1265.0

In [66]:
```
df2[df2.diagnosis=='M'].count()
```

Out[66]:
```
id           212
diagnosis    212
area_mean    212
dtype: int64
```

In [67]:
```
df2[["area_mean","id"]].head()
```

Out[67]:

	area_mean	id
0	1001.0	842302
1	1326.0	842517
2	1203.0	84300903
3	386.1	84348301
4	1297.0	84358402

38.8 排序

In [68]:
```
df2.head(8)
```

Out[68]:

	id	diagnosis	area_mean
0	842302	M	1001.0
1	842517	M	1326.0
2	84300903	M	1203.0
3	84348301	M	386.1
4	84358402	M	1297.0
5	843786	M	477.1
6	844359	M	1040.0
7	84458202	M	577.9

In [69]:
```
df2.sort_values(by="area_mean",axis=0,ascending=True).head()
```

Out[69]:

	id	diagnosis	area_mean
101	862722	B	143.5
539	921362	B	170.4
538	921092	B	178.8
568	92751	B	181.0
46	85713702	B	201.9

In [70]:
```
df2.sort_index(axis=1).head(3)
```

Out[70]:

	area_mean	diagnosis	id
0	1001.0	M	842302
1	1326.0	M	842517
2	1203.0	M	84300903

In [71]:
```
df2.sort_index(axis=0,ascending=False).head(3)
```

Out[71]:

	id	diagnosis	area_mean
568	92751	B	181.0
567	927241	M	1265.0
566	926954	M	858.1

38.9导入导出

In [72]:
```
import os
print(os.getcwd())
```

C:\Users\soloman\clm

In [73]:
```
df2.head(3).to_csv("df2.csv")
```

In [74]:
```
import pandas as pd
df3 = pd.read_csv('df2.csv')
```

In [75]:
```
df3
```

Out[75]:

	Unnamed: 0	id	diagnosis	area_mean
0	0	842302	M	1001.0
1	1	842517	M	1326.0
2	2	84300903	M	1203.0

In [76]:
```
df3 = pd.read_csv('df2.csv')
```

In [77]:
```
df3
```

Out[77]:

	Unnamed: 0	id	diagnosis	area_mean
0	0	842302	M	1001.0
1	1	842517	M	1326.0
2	2	84300903	M	1203.0

In [78]:
```
df2.head(3).to_excel("df3.xls")
```

In [79]:
```
df3 = pd.read_excel("df3.xls")
df3
```

Out[79]:

	id	diagnosis	area_mean
0	842302	M	1001
1	842517	M	1326
2	84300903	M	1203

38.10 缺失数据处理

In [80]:
```
df3.empty
```

Out[80]: False

In [81]:
```
np.nan +1
```

Out[81]: nan

In [82]:
```
np.nan-np.nan
```

Out[82]: nan

In [83]:
```
None+1
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-83-6e170940e108> in <module>()
----> 1 None+1
      2 #【提示】报错信息为TypeError: unsupported operand type(s) for +: 'NoneType' and 'int'，原因分析：None不能参加算数运算。
      3
      4 #【注意】
      5     #None是Python基础语法中的特殊数据类型，不属于数值类型，不能参加算数运算

TypeError: unsupported operand type(s) for +: 'NoneType' and 'int'
```

In [84]:
```
import pandas as pd
import numpy as np
A=pd.DataFrame(np.array([10,10,20,20]).reshape(2,2),columns=list("ab"),index=list("SW"))
A
```

Out[84]:

	a	b
S	10	10
W	20	20

In [85]:
```
list("ab")
```

Out[85]:
```
['a', 'b']
```

In [86]:
```
B=pd.DataFrame(np.array([1,1,1,2,2,2,3,3,3]).reshape(3,3), columns=list("abc"),index=list("SWT"))
B
```

Out[86]:

	a	b	c
S	1	1	1
W	2	2	2
T	3	3	3

In [87]:
```
C=A+B
C
```

Out[87]:

	a	b	c
S	11.0	11.0	NaN
T	NaN	NaN	NaN
W	22.0	22.0	NaN

In [88]:
```
A.add(B,fill_value=0)
```

Out[88]:

	a	b	c
S	11.0	11.0	1.0

T	3.0	3.0	3.0
W	22.0	22.0	2.0

In [89]:
```
A.add(B,fill_value=A.stack().mean())
```

Out[89]:

	a	b	c
S	11.0	11.0	16.0
T	18.0	18.0	18.0
W	22.0	22.0	17.0

In [90]:
```
A.mean()
```

Out[90]:
```
a    15.0
b    15.0
dtype: float64
```

In [91]:
```
A.stack()
```

Out[91]:
```
S  a    10
   b    10
W  a    20
   b    20
dtype: int32
```

In [92]:
```
A.stack().mean()
```

Out[92]:
```
15.0
```

In [93]:
```
C
```

Out[93]:

	a	b	c
S	11.0	11.0	NaN
T	NaN	NaN	NaN
W	22.0	22.0	NaN

In [94]:
```
C.isnull()
```

Out[94]:

	a	b	c
S	False	False	True
T	True	True	True
W	False	False	True

In [95]:
```
C.notnull()
```

Out[95]:

	a	b	c
S	True	True	False

T	False	False	False
W	True	True	False

In [96]:
```
C.dropna(axis='index')
```

Out[96]:

	a	b	c

In [97]:
```
C.fillna(0)
```

Out[97]:

	a	b	c
S	11.0	11.0	0.0
T	0.0	0.0	0.0
W	22.0	22.0	0.0

In [98]:
```
C.fillna(method="ffill")
```

Out[98]:

	a	b	c
S	11.0	11.0	NaN
T	11.0	11.0	NaN
W	22.0	22.0	NaN

In [99]:
```
C.fillna(method="bfill",axis=1)
```

Out[99]:

	a	b	c
S	11.0	11.0	NaN
T	NaN	NaN	NaN
W	22.0	22.0	NaN

38.11 分组统计

In [100]:
```
import pandas as pd
df2 = pd.read_csv('bc_data.csv')
df2=df2[["id","diagnosis","area_mean"]]
df2.head()
```

Out[100]:

	id	diagnosis	area_mean
0	842302	M	1001.0
1	842517	M	1326.0
2	84300903	M	1203.0
3	84348301	M	386.1
4	84358402	M	1297.0

In [101]:
```
df2.groupby("diagnosis")["area_mean"].mean()
```

Out[101]:
```
diagnosis
B    462.790196
M    978.376415
Name: area_mean, dtype: float64
```

In [102]:
```
df2.groupby("diagnosis")["area_mean"].aggregate(["mean","sum","max",np.median])
```

Out[102]:

	mean	sum	max	median
diagnosis				
B	462.790196	165216.1	992.1	458.4
M	978.376415	207415.8	2501.0	932.0

In [103]:
```
df2.groupby("diagnosis")["area_mean"].aggregate(["mean","sum"]).unstack()
```

Out[103]:
```
      diagnosis
mean  B            462.790196
      M            978.376415
sum   B         165216.100000
      M         207415.800000
dtype: float64
```

In [104]:
```
def myfunc(x):
  x["area_mean"]/=x["area_mean"].sum()
  return x

df2.groupby("diagnosis").apply(myfunc).head()
```

Out[104]:

	id	diagnosis	area_mean
0	842302	M	0.004826
1	842517	M	0.006393
2	84300903	M	0.005800
3	84348301	M	0.001861
4	84358402	M	0.006253