In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
frame=pd.DataFrame(np.arange(9).reshape(3,3),index=['a','c','d'],columns=['chennai
```

In [3]:
```
frm=pd.DataFrame(np.arange(9).reshape(3,3),index=['a','b','c'],columns=['delhi','a
frm
```

Out[3]:

	delhi	agra	ggs
a	0	1	2
b	3	4	5
c	6	7	8

In [4]:
```
frame
```

Out[4]:

	chennai	delhi	Mumbai
a	0	1	2
c	3	4	5
d	6	7	8

In [5]:
```
city=['chennai','delhi','kolkata','Mumbai']
```

In [6]:
```
frame1=frame.reindex(columns=city)
```

In [7]:
```
frame1
```

Out[7]:

	chennai	delhi	kolkata	Mumbai
a	0	1	NaN	2
c	3	4	NaN	5
d	6	7	NaN	8

In [8]:
```
frame.reindex(index=['a','c','d','e'],method='ffill')
```

Out[8]:

	chennai	delhi	Mumbai
a	0	1	2
c	3	4	5
d	6	7	8
e	6	7	8

In [9]:
```
frame1.reindex(index=['a','c','d','e'],method='ffill')
```

Out[9]:

	chennai	delhi	kolkata	Mumbai
a	0	1	NaN	2
c	3	4	NaN	5
d	6	7	NaN	8
e	6	7	NaN	8

In [10]:
```
pd.__version__
```

Out[10]:
```
'1.4.2'
```

In [11]:
```
nframe=frame.drop(['a','d'])
```

In [12]:
```
nframe
```

Out[12]:

	chennai	delhi	Mumbai
c	3	4	5

In [13]:
```
nframe=frame.drop('Mumbai', axis=1)
nframe
```

Out[13]:

	chennai	delhi
a	0	1
c	3	4
d	6	7

In [14]:
```
nframe
```

Out[14]:

	chennai	delhi
a	0	1
c	3	4
d	6	7

In [15]:
```
ob1=pd.Series(np.arange(4.),index=['a','b','c','d'])
```

In [16]:
```
ob1['b']
```

Out[16]:
```
1.0
```

In [17]:
```
ob1
```

Out[17]:
```
a    0.0
b    1.0
c    2.0
d    3.0
dtype: float64
```

In [18]:
```
ob1[['b','d']]
```

Out[18]:
```
b    1.0
d    3.0
dtype: float64
```

In [19]:
```
x=ob1[['a']]
```

In [20]:
```
ob1[[1,3]]
```

Out[20]:
```
b    1.0
d    3.0
dtype: float64
```

In [21]:
```
ob1[ob1>1]
```

Out[21]:
```
c    2.0
d    3.0
dtype: float64
```

In [22]:
```
ob1
```

Out[22]:
```
a    0.0
b    1.0
c    2.0
d    3.0
dtype: float64
```

In [23]:
```
ob1['b':'c']=7
```

In [24]:
```
ob1
```

Out[24]:
```
a    0.0
b    7.0
c    7.0
d    3.0
dtype: float64
```

In [25]:
```
data=pd.DataFrame(np.arange(16).reshape(4,4),index=['Delhi','Mumbai','Chennai','ko
```

In [26]:
```
data
```

Out[26]:

	one	two	three	four
Delhi	0	1	2	3
Mumbai	4	5	6	7
Chennai	8	9	10	11
kolkata	12	13	14	15

In [27]:
```
data['two']
```

Out[27]:
```
Delhi       1
Mumbai      5
Chennai     9
kolkata    13
Name: two, dtype: int32
```

In [28]:
```
data[:1]
```

Out[28]:

	one	two	three	four
Delhi	0	1	2	3

In [29]:
```
data[data['three']>5]
```

Out[29]:

	one	two	three	four
Mumbai	4	5	6	7
Chennai	8	9	10	11
kolkata	12	13	14	15

In [30]:
```
data[data['one']<3]
```

Out[30]:

	one	two	three	four
Delhi	0	1	2	3

In [31]:
```
data.xs('one',axis=1) #The xs() method returns a specified section of the DataFrame
```

Out[31]:
```
Delhi       0
Mumbai      4
Chennai     8
kolkata    12
Name: one, dtype: int32
```

In [32]:
```
data.xs('Delhi')
```

Out[32]:
```
one      0
two      1
three    2
four     3
Name: Delhi, dtype: int32
```

In [33]:
```
data.iloc[1] #The iloc() function in python is defined in the Pandas module, which
```

Out[33]:
```
one      4
two      5
three    6
four     7
Name: Mumbai, dtype: int32
```

In [34]:
```
data.at['Mumbai','one']=6
```

In [35]:
```
data
```

Out[35]:

	one	two	three	four
Delhi	0	1	2	3
Mumbai	6	5	6	7
Chennai	8	9	10	11
kolkata	12	13	14	15

In [36]:
```
data['one']
```

Out[36]:
```
Delhi       0
Mumbai      6
Chennai     8
kolkata    12
Name: one, dtype: int32
```

arithmatic with fill values

In [37]:
```
df1=pd.DataFrame(np.arange(12.).reshape(3,4),columns=list('abcd'))
```

In [38]:
```
df2=pd.DataFrame(np.arange(20.).reshape(4,5),columns=list('abcde'))
```

In [39]:
```
df2
```

Out[39]:

	a	b	c	d	e
0	0.0	1.0	2.0	3.0	4.0
1	5.0	6.0	7.0	8.0	9.0
2	10.0	11.0	12.0	13.0	14.0
3	15.0	16.0	17.0	18.0	19.0

In [40]:
```
df1+df2
```

Out[40]:

	a	b	c	d	e
0	0.0	2.0	4.0	6.0	NaN
1	9.0	11.0	13.0	15.0	NaN
2	18.0	20.0	22.0	24.0	NaN
3	NaN	NaN	NaN	NaN	NaN

In [41]:
```
df1.add(df2,fill_value=0)
```

Out[41]:

	a	b	c	d	e
0	0.0	2.0	4.0	6.0	4.0
1	9.0	11.0	13.0	15.0	9.0
2	18.0	20.0	22.0	24.0	14.0
3	15.0	16.0	17.0	18.0	19.0

In [42]:
```
df1
```

Out[42]:

	a	b	c	d
0	0.0	1.0	2.0	3.0
1	4.0	5.0	6.0	7.0
2	8.0	9.0	10.0	11.0

In [43]:
```
df2
```

Out[43]:

	a	b	c	d	e
0	0.0	1.0	2.0	3.0	4.0
1	5.0	6.0	7.0	8.0	9.0
2	10.0	11.0	12.0	13.0	14.0
3	15.0	16.0	17.0	18.0	19.0

In [44]:
```
df_s1=df1.iloc[0]
```

In [45]:
```
df_s1
```

Out[45]:
```
a    0.0
b    1.0
c    2.0
d    3.0
Name: 0, dtype: float64
```

In [46]:
```
df1-df_s1
```

Out[46]:

	a	b	c	d
0	0.0	0.0	0.0	0.0
1	4.0	4.0	4.0	4.0
2	8.0	8.0	8.0	8.0

In [47]:
```
ser2=pd.Series(range(3),index=['b','d','f'])
```

In [48]:
```
ser2
```

Out[48]:
```
b    0
d    1
f    2
dtype: int64
```

In [49]:
```
df1+ser2
```

Out[49]:

	a	b	c	d	f
0	NaN	1.0	NaN	4.0	NaN
1	NaN	5.0	NaN	8.0	NaN
2	NaN	9.0	NaN	12.0	NaN

In [50]:
```
ser3=df1['b']
```

In [51]:
```
ser3
```

Out[51]:
```
0    1.0
1    5.0
2    9.0
Name: b, dtype: float64
```

In [52]:
```
df3=df1.sub(ser3,axis=0)
```

In [53]:
```
df3
```

Out[53]:

	a	b	c	d
0	-1.0	0.0	1.0	2.0
1	-1.0	0.0	1.0	2.0
2	-1.0	0.0	1.0	2.0

In [54]:
```
df1
```

Out[54]:

	a	b	c	d
0	0.0	1.0	2.0	3.0
1	4.0	5.0	6.0	7.0
2	8.0	9.0	10.0	11.0

In [55]:
```
ser3
```

Out[55]:
```
0    1.0
1    5.0
2    9.0
Name: b, dtype: float64
```

In [56]:
```
np.abs(df3)
```

Out[56]:

	a	b	c	d
0	1.0	0.0	1.0	2.0
1	1.0	0.0	1.0	2.0
2	1.0	0.0	1.0	2.0

In [57]:
```
f=lambda a:a.max()-a.min()
```

In [58]:
```
df1.apply(f)
```

Out[58]:
```
a    8.0
b    8.0
c    8.0
d    8.0
dtype: float64
```

In [59]:
```
df1.apply(f, axis=1)
```

Out[59]:
```
0    3.0
1    3.0
2    3.0
dtype: float64
```

In [60]:
```
format=lambda a:'%.2f' %a
```

In [61]:
```
df1.applymap(format)
```

Out[61]:

	a	b	c	d
0	0.00	1.00	2.00	3.00
1	4.00	5.00	6.00	7.00
2	8.00	9.00	10.00	11.00

In [62]:
```
df1['a'].map(format)
```

Out[62]:
```
0    0.00
1    4.00
2    8.00
Name: a, dtype: object
```

In [63]:
```
df1
```

Out[63]:

	a	b	c	d
0	0.0	1.0	2.0	3.0
1	4.0	5.0	6.0	7.0
2	8.0	9.0	10.0	11.0

In [64]:
```
obj1=pd.Series(range(4),index=['b','d','c','a'])
obj1
```

Out[64]:
```
b    0
d    1
c    2
a    3
dtype: int64
```

In [65]:
```
import pandas as pd
```

In [66]:
```
obj1.sort_index()
```

Out[66]:
```
a    3
b    0
c    2
d    1
dtype: int64
```

In [67]:
```
obj1
```

Out[67]:
```
b    0
d    1
c    2
a    3
dtype: int64
```

In [68]:
```
frame1=pd.DataFrame(np.arange(8).reshape(2,4),index=['two','one'],columns=['a','c'
```

In [69]:
```
import numpy as np
```

In [70]:
```
frame1
```

Out[70]:

	a	c	b	d
two	0	1	2	3
one	4	5	6	7

In [71]:
```
frame1.sort_index()
```

Out[71]:

	a	c	b	d
one	4	5	6	7
two	0	1	2	3

In [72]:
```
frame1.sort_index(axis=1,ascending=False)
```

Out[72]:

	d	c	b	a
two	3	1	2	0
one	7	5	6	4

In [73]:
```
series1=pd.Series([5,-4,8,1])
series1
```

Out[73]:
```
0    5
1   -4
2    8
3    1
dtype: int64
```

In [74]:
```
series1.sort_values()
```

Out[74]:
```
1   -4
3    1
0    5
2    8
dtype: int64
```

In [75]:
```
obj=pd.Series([4,np.nan,8,np.nan,9,-5])
```

In [76]:
```
obj.sort_values()
```

Out[76]:
```
5   -5.0
0    4.0
2    8.0
4    9.0
1    NaN
3    NaN
dtype: float64
```

In [77]:
```
frame6=pd.DataFrame({'d':[7,5,4,9],'b':[3,7,5,9]})
```

In [78]:
```
frame6.sort_values(by=['d'])
```

Out[78]:

	d	b
2	4	5
1	5	7
0	7	3
3	9	9

In [79]:
```
obj11=pd.Series([9,7,6,3,-5,7,3,4,1,3])
```

In [80]:
```
obj11.rank()
```

Out[80]:
```
0    10.0
1     8.5
2     7.0
3     4.0
4     1.0
5     8.5
6     4.0
7     6.0
8     2.0
9     4.0
dtype: float64
```

In [81]:
```
obj11.rank(method='max')
```

Out[81]:
```
0    10.0
1     9.0
2     7.0
3     5.0
4     1.0
5     9.0
6     5.0
7     6.0
8     2.0
9     5.0
dtype: float64
```

duplicate values

In [82]:
```
import pandas as pd
```

In [83]:
```
dups=pd.Series(range(5),index=list('aabbc'))
```

In [84]:
```
dups
```

Out[84]:
```
a    0
a    1
b    2
b    3
c    4
dtype: int64
```

In [85]:
```
dups.index.unique()
```

Out[85]:
```
Index(['a', 'b', 'c'], dtype='object')
```

In [86]:
```
dups.index.is_unique
```

Out[86]:
```
False
```

In [87]:
```
dups['a']
```

Out[87]:
```
a    0
a    1
dtype: int64
```

In [88]:
```
dup_f=pd.DataFrame(np.random.randn(5,3),index=list('aabbc'))
```

In [89]:
```
dup_f
```

Out[89]:

	0	1	2
a	0.158308	3.309641	-1.169207
a	0.379315	-0.580638	-0.543815
b	1.511547	-0.684070	2.880783
b	0.611422	0.814343	-0.859751
c	0.079899	0.755809	0.646249

In [90]:
```
dup_f.loc['a']
```

Out[90]:

	0	1	2
a	0.158308	3.309641	-1.169207
a	0.379315	-0.580638	-0.543815

In [91]:
```
dup_f.xs('a')
```

Out[91]:

	0	1	2
a	0.158308	3.309641	-1.169207
a	0.379315	-0.580638	-0.543815

Descriptive statistics

In [92]:
```
new_df=pd.DataFrame([[1.7,3.4],[np.nan,5.1],[3.4, np.nan],[4.8,7.3]],index=list('al
```

In [93]:
```
new_df
```

Out[93]:

	one	two
a	1.7	3.4
b	NaN	5.1
c	3.4	NaN
d	4.8	7.3

In [94]:
```
new_df.sum()
```

Out[94]:
```
one     9.9
two    15.8
dtype: float64
```

In [95]:
```
new_df.sum(axis=1)
```

Out[95]:
```
a     5.1
b     5.1
c     3.4
d    12.1
dtype: float64
```

In [96]:
```
new_df.mean()
```

Out[96]:
```
one    3.300000
two    5.266667
dtype: float64
```

In [97]:
```
new_df.mean(axis=1,skipna=False)
```

Out[97]:
```
a    2.55
b     NaN
c     NaN
d    6.05
dtype: float64
```

In [98]:
```
new_df.idxmax()
```

Out[98]:
```
one    d
two    d
dtype: object
```

In [99]:
```
new_df.cumsum()
```

Out[99]:

	one	two
a	1.7	3.4
b	NaN	8.5
c	5.1	NaN
d	9.9	15.8

In [100…
```
new_df
```

Out[100]:

	one	two
a	1.7	3.4
b	NaN	5.1
c	3.4	NaN
d	4.8	7.3

In [101…
```
new_df.describe()
```

Out[101]:

	one	two
count	3.000000	3.000000
mean	3.300000	5.266667
std	1.552417	1.955335
min	1.700000	3.400000
25%	2.550000	4.250000
50%	3.400000	5.100000
75%	4.100000	6.200000
max	4.800000	7.300000

In [102…
```
new_df.quantile()
```

Out[102]:
```
one    3.4
two    5.1
Name: 0.5, dtype: float64
```

In [103…
```
obj_r=pd.Series(list('accbddbbgh'))
obj_r
```

Out[103]:
```
0    a
1    c
2    c
3    b
4    d
5    d
6    b
7    b
8    g
9    h
dtype: object
```

In [104…
```
uni=obj_r.unique()
```

In [105…
```
uni
```

Out[105]:
```
array(['a', 'c', 'b', 'd', 'g', 'h'], dtype=object)
```

In [106…
```
obj_r.value_counts()
```

Out[106]:
```
b    3
c    2
d    2
a    1
g    1
h    1
dtype: int64
```

In [107…
```
pd.value_counts(obj_r.values,sort=False)
```

Out[107]:
```
a    1
c    2
b    3
d    2
g    1
h    1
dtype: int64
```

In [108…]
```
mask=obj_r.isin(['b','c'])
```

In [109…]
```
mask
```

Out[109]:
```
0    False
1     True
2     True
3     True
4    False
5    False
6     True
7     True
8    False
9    False
dtype: bool
```

In [110…]
```
obj_r[mask]
```

Out[110]:
```
1    c
2    c
3    b
6    b
7    b
dtype: object
```

In [111…]
```
new_df.isnull()
```

Out[111]:

	one	two
a	False	False
b	True	False
c	False	True
d	False	False

In [112…]
```
new_df.dropna()
```

Out[112]:

	one	two
a	1.7	3.4
d	4.8	7.3

In [113…]
```
new_df.dropna(how='all')
```

Out[113]:

	one	two
a	1.7	3.4
b	NaN	5.1
c	3.4	NaN
d	4.8	7.3

In [114…]
```
new_df[new_df.notnull()]
```

Out[114]:

	one	two
a	1.7	3.4
b	NaN	5.1
c	3.4	NaN
d	4.8	7.3

In [115…]:
```
df3=pd.DataFrame(np.random.randn(7,3))
df3
```

Out[115]:

	0	1	2
0	-0.047804	-0.125576	-1.521705
1	-0.041407	0.019810	0.139815
2	-0.085594	-0.288197	0.514307
3	-0.667571	1.875485	-1.344551
4	-0.676911	1.149603	-0.646324
5	1.063258	1.158892	1.619210
6	-0.049164	-2.074349	-0.058741

In [116…]:
```
df3.loc[:4,1]=np.nan
df3
```

Out[116]:

	0	1	2
0	-0.047804	NaN	-1.521705
1	-0.041407	NaN	0.139815
2	-0.085594	NaN	0.514307
3	-0.667571	NaN	-1.344551
4	-0.676911	NaN	-0.646324
5	1.063258	1.158892	1.619210
6	-0.049164	-2.074349	-0.058741

In [117…]:
```
df3.loc[:2,2]=np.nan
df3
```

Out[117]:

	0	1	2
0	-0.047804	NaN	NaN
1	-0.041407	NaN	NaN
2	-0.085594	NaN	NaN
3	-0.667571	NaN	-1.344551
4	-0.676911	NaN	-0.646324
5	1.063258	1.158892	1.619210
6	-0.049164	-2.074349	-0.058741

In [118…
```
df3
```

Out[118]:

	0	1	2
0	-0.047804	NaN	NaN
1	-0.041407	NaN	NaN
2	-0.085594	NaN	NaN
3	-0.667571	NaN	-1.344551
4	-0.676911	NaN	-0.646324
5	1.063258	1.158892	1.619210
6	-0.049164	-2.074349	-0.058741

In [119…
```
df3.fillna(0)
```

Out[119]:

	0	1	2
0	-0.047804	0.000000	0.000000
1	-0.041407	0.000000	0.000000
2	-0.085594	0.000000	0.000000
3	-0.667571	0.000000	-1.344551
4	-0.676911	0.000000	-0.646324
5	1.063258	1.158892	1.619210
6	-0.049164	-2.074349	-0.058741

In [120…
```
df3
```

Out[120]:

	0	1	2
0	-0.047804	NaN	NaN
1	-0.041407	NaN	NaN
2	-0.085594	NaN	NaN
3	-0.667571	NaN	-1.344551
4	-0.676911	NaN	-0.646324
5	1.063258	1.158892	1.619210
6	-0.049164	-2.074349	-0.058741

In [121…
```
df3.fillna({1:0.5, 2:-1},inplace=True)
```

In [122…
```
df3
```

Out[122]:

	0	1	2
0	-0.047804	0.500000	-1.000000
1	-0.041407	0.500000	-1.000000
2	-0.085594	0.500000	-1.000000
3	-0.667571	0.500000	-1.344551
4	-0.676911	0.500000	-0.646324
5	1.063258	1.158892	1.619210
6	-0.049164	-2.074349	-0.058741

hierarchical indexing

In [123…
```
data=pd.Series(np.random.randn(10),index=[list('aaabbbccdd'),[1,2,3,1,2,3,1,2,2,3]]
```

In [124…
```
data
```

Out[124]:
```
a  1    0.216727
   2   -0.220883
   3    0.565680
b  1    0.484475
   2    0.846357
   3    1.820405
c  1   -0.473590
   2   -0.667612
d  2   -0.643173
   3   -1.682514
dtype: float64
```

In [125…
```
data.index
```

Out[125]:
```
MultiIndex([('a', 1),
            ('a', 2),
            ('a', 3),
            ('b', 1),
            ('b', 2),
            ('b', 3),
            ('c', 1),
            ('c', 2),
            ('d', 2),
            ('d', 3)],
           )
```

In [126…
```
data['c']
```

Out[126]:
```
1   -0.473590
2   -0.667612
dtype: float64
```

In [127…
```
data['b':'c']
```

Out[127]:
```
b  1    0.484475
   2    0.846357
   3    1.820405
c  1   -0.473590
   2   -0.667612
dtype: float64
```

In [128...]
```
data[:,3]
```

Out[128]:
```
a    0.565680
b    1.820405
d   -1.682514
dtype: float64
```

In [129...]
```
data[['b','d']]
```

Out[129]:
```
b  1    0.484475
   2    0.846357
   3    1.820405
d  2   -0.643173
   3   -1.682514
dtype: float64
```

In [130...]
```
data1=data.unstack()
```

In [131...]
```
data1
```

Out[131]:

	1	2	3
a	0.216727	-0.220883	0.565680
b	0.484475	0.846357	1.820405
c	-0.473590	-0.667612	NaN
d	NaN	-0.643173	-1.682514

In [132...]
```
data1.stack()
```

Out[132]:
```
a  1    0.216727
   2   -0.220883
   3    0.565680
b  1    0.484475
   2    0.846357
   3    1.820405
c  1   -0.473590
   2   -0.667612
d  2   -0.643173
   3   -1.682514
dtype: float64
```

In [133...]
```
frame11=pd.DataFrame(np.arange(12).reshape(4,3),index=[['a','a','b','b'],[1,2,1,2]
```

In [134...]
```
frame11
```

Out[134]:

		c1	c2
		red	**green**	**red**
a	**1**	0	1	2
	2	3	4	5
b	**1**	6	7	8
	2	9	10	11

In [135...]
```
frame11.index.names=['key1','key2']
```

In [136...]
```
frame11.columns.names=['col','color']
```

In [137…
```
frame11
```

Out[137]:

		col		c1	c2
		color	red	green	red
key1	key2				
a	1		0	1	2
	2		3	4	5
b	1		6	7	8
	2		9	10	11

In [138…
```
frame11['c1']
```

Out[138]:

		color	red	green
key1	key2			
a	1		0	1
	2		3	4
b	1		6	7
	2		9	10

In [139…
```
frame11['c2']
```

Out[139]:

		color	red
key1	key2		
a	1		2
	2		5
b	1		8
	2		11

reordering / sorting levels

In [140…
```
frame11.swaplevel('key1','key2')
```

Out[140]:

		col		c1	c2
		color	red	green	red
key2	key1				
1	a		0	1	2
2	a		3	4	5
1	b		6	7	8
2	b		9	10	11

In [141…
```
frame11.sum(level='key1')
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_10732\1813465319.py:1: FutureWarning:
Using the level keyword in DataFrame and Series aggregations is deprecated and wil
l be removed in a future version. Use groupby instead. df.sum(level=1) should use
df.groupby(level=1).sum().
  frame11.sum(level='key1')
```

Out[141]:

col	c1	c2
color	red	green	red
key1			
a	3	5	7
b	15	17	19

In [142…
```
frame11.sum(level='color', axis=1)
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_10732\1716229740.py:1: FutureWarning:
Using the level keyword in DataFrame and Series aggregations is deprecated and wil
l be removed in a future version. Use groupby instead. df.sum(level=1) should use
df.groupby(level=1).sum().
  frame11.sum(level='color', axis=1)
```

Out[142]:

	color	red	green
key1	key2		
a	1	2	1
	2	8	4
b	1	14	7
	2	20	10

In [143…
```
import pandas as pd
```

In [144…
```
frame=pd.DataFrame({'a':range(7),'b':range(7,0,-1),'c':['one','one','one','two','tw
```

In [145…
```
frame
```

Out[145]:

	a	b	c	d
0	0	7	one	0
1	1	6	one	1
2	2	5	one	2
3	3	4	two	0
4	4	3	two	1
5	5	2	two	2
6	6	1	two	3

In [146…
```
frame1=frame.set_index(['c','d'])
```

In [147…
```
frame
```

Out[147]:

	a	b	c	d
0	0	7	one	0
1	1	6	one	1
2	2	5	one	2
3	3	4	two	0
4	4	3	two	1
5	5	2	two	2
6	6	1	two	3

In [148…
```
frame1
```

Out[148]:

		a	b
c	**d**		
one	**0**	0	7
	1	1	6
	2	2	5
two	**0**	3	4
	1	4	3
	2	5	2
	3	6	1

In [149…
```
frame11=frame.set_index(['c','d'],drop=False)
```

In [150…
```
frame11
```

Out[150]:

		a	b	c	d
c	**d**				
one	**0**	0	7	one	0
	1	1	6	one	1
	2	2	5	one	2
two	**0**	3	4	two	0
	1	4	3	two	1
	2	5	2	two	2
	3	6	1	two	3

In [151…
```
frame1.reset_index()
```

Out[151]:

	c	d	a	b
0	one	0	0	7
1	one	1	1	6
2	one	2	2	5
3	two	0	3	4
4	two	1	4	3
5	two	2	5	2
6	two	3	6	1

In [156…
```
df1=pd.read_csv('ex1.csv')
```

In [157…
```
df1
```

Out[157]:

	a	b	c	d	message
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

In [163…
```
df2=pd.read_table('ex1.csv',sep=',')
```

In [164…
```
df2
```

Out[164]:

	a	b	c	d	message
0	1	2	3	4	hello
1	5	6	7	8	world
2	9	10	11	12	foo

In [165…
```
df2.to_csv('out1.csv',sep='|')
```

In [166…
```
 !cat 'out1.csv'
```

```
'cat' is not recognized as an internal or external command,
operable program or batch file.
```

In [ ]:
```

```

In [ ]:
```
import sys
```

In [ ]:
```
df2.to_csv(sys.stdout,sep='|')
```

In [168…
```
df11=pd.read_csv('IRIS.csv')
```

In [171…
```
df1=pd.DataFrame({'key':['b','b','a','c','a','a','b'],'data1':range(7)})
df1
```

Out[171]:

	key	data1
0	b	0
1	b	1
2	a	2
3	c	3
4	a	4
5	a	5
6	b	6

In [172…
```
df11
```

Out[172]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [173…
```
df11['petal_length']
```

Out[173]:
```
0      1.4
1      1.4
2      1.3
3      1.5
4      1.4
      ...
145    5.2
146    5.0
147    5.2
148    5.4
149    5.1
Name: petal_length, Length: 150, dtype: float64
```

In [174…
```
df11.loc[146]
```

Out[174]:
```
sepal_length              6.3
sepal_width               2.5
petal_length              5.0
petal_width               1.9
species         Iris-virginica
Name: 146, dtype: object
```

In [175…
```
df2=pd.DataFrame({'key':['a','b','d'],'data2':range(3)})
df2
```

Out[175]:

	key	data2
0	a	0
1	b	1
2	d	2

In [176…
```
df11.describe()
```

Out[176]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [177…
```
df1
```

Out[177]:

	key	data1
0	b	0
1	b	1
2	a	2
3	c	3
4	a	4
5	a	5
6	b	6

In [178…
```
df2
```

Out[178]:

	key	data2
0	a	0
1	b	1
2	d	2

In [179…
```
pd.merge(df1,df2)
```

Out[179]:

	key	data1	data2
0	b	0	1
1	b	1	1
2	b	6	1
3	a	2	0
4	a	4	0
5	a	5	0

In [180…
```
pd.merge(df2,df1)
```

Out[180]:

	key	data2	data1
0	a	0	2
1	a	0	4
2	a	0	5
3	b	1	0
4	b	1	1
5	b	1	6

In [181…
```
pd.merge(df1,df2,on='key')
```

Out[181]:

	key	data1	data2
0	b	0	1
1	b	1	1
2	b	6	1
3	a	2	0
4	a	4	0
5	a	5	0

In [182…
```
df3=pd.DataFrame({'lkey':['b','b','a','c','a','a','b'],'data1':range(7)})
```

In [183…
```
df4=pd.DataFrame({'rkey':['a','b','d'],'data2':range(3)})
```

In [184…
```
df3
```

Out[184]:

	lkey	data1
0	b	0
1	b	1
2	a	2
3	c	3
4	a	4
5	a	5
6	b	6

In [186...]
```
pd.merge(df3,df4,left_on='lkey',right_on='rkey')
```

Out[186]:

	lkey	data1	rkey	data2
0	b	0	b	1
1	b	1	b	1
2	b	6	b	1
3	a	2	a	0
4	a	4	a	0
5	a	5	a	0

df4

In [185...]
```
df4
```

Out[185]:

	rkey	data2
0	a	0
1	b	1
2	d	2

In [189...]
```
pd.merge(df1,df2,how='outer')
```

Out[189]:

	key	data1	data2
0	b	0.0	1.0
1	b	1.0	1.0
2	b	6.0	1.0
3	a	2.0	0.0
4	a	4.0	0.0
5	a	5.0	0.0
6	c	3.0	NaN
7	d	NaN	2.0

In [190...]
```
pd.merge(df1,df2,on='key',how='inner')
```

Out[190]:

	key	data1	data2
0	b	0	1
1	b	1	1
2	b	6	1
3	a	2	0
4	a	4	0
5	a	5	0

In [191…
```
left=pd.DataFrame({'key1':['foo','foo','ber'],'key2':['one','two','one'],'lval':[1
```

In [192…
```
left
```

Out[192]:

	key1	key2	lval
0	foo	one	1
1	foo	two	2
2	ber	one	3

In [193…
```
rightt=pd.DataFrame({'key1':['foo','foo','ber','bar'],'key2':['one','one','one','tw
```

In [194…
```
rightt
```

Out[194]:

	key1	key2	lval
0	foo	one	4
1	foo	one	5
2	ber	one	6
3	bar	two	7

In [195…
```
pd.merge(left,rightt,on=['key1','key2'])
```

Out[195]:

	key1	key2	lval_x	lval_y
0	foo	one	1	4
1	foo	one	1	5
2	ber	one	3	6

In [196…
```
pd.merge(left,rightt,on=['key1','key2'],how='outer')
```

Out[196]:

	key1	key2	lval_x	lval_y
0	foo	one	1.0	4.0
1	foo	one	1.0	5.0
2	foo	two	2.0	NaN
3	ber	one	3.0	6.0
4	bar	two	NaN	7.0

In [197…
```
pd.merge(left,rightt,on='key1',suffixes=('_left','_right'))
```

Out[197]:

	key1	key2_left	lval_left	key2_right	lval_right
0	foo	one	1	one	4
1	foo	one	1	one	5
2	foo	two	2	one	4
3	foo	two	2	one	5
4	ber	one	3	one	6

In [198…
```
left1=pd.DataFrame({'key':['a','b','a','a','b','c'],'value':range(6)})
```

In [199…
```
right1=pd.DataFrame({'group_val':[3.5,7]},index=['a','b'])
```

In [200…
```
left1
```

Out[200]:

	key	value
0	a	0
1	b	1
2	a	2
3	a	3
4	b	4
5	c	5

In [201…
```
right1
```

Out[201]:

	group_val
a	3.5
b	7.0

In [202…
```
pd.merge(left1,right1,left_on='key',right_index=True,how='outer')
```

Out[202]:

	key	value	group_val
0	a	0	3.5
2	a	2	3.5
3	a	3	3.5
1	b	1	7.0
4	b	4	7.0
5	c	5	NaN

In [203…
```
left_h=pd.DataFrame({'key1':['c1','c1','c1','c2','c2'],'key2':[2000,2001,2002,2001
```

In [204…
```
import numpy as np
```

In [205…
```
right_h=pd.DataFrame((np.arange(12.).reshape(6,2)),index=[['c2','c2','c1','c1','c1
```

In [206…
```
left_h
```

Out[206]:

	key1	key2	data
0	c1	2000	0.0
1	c1	2001	1.0
2	c1	2002	2.0
3	c2	2001	3.0
4	c2	2002	4.0

In [207…
```
right_h
```

Out[207]:

		col1	col2
c2	**2001**	0.0	1.0
	2000	2.0	3.0
c1	**2000**	4.0	5.0
	2000	6.0	7.0
	2001	8.0	9.0
	2002	10.0	11.0

In [208…
```
pd.merge(left_h,right_h,left_on=['key1','key2'],right_index=True)
```

Out[208]:

	key1	key2	data	col1	col2
0	c1	2000	0.0	4.0	5.0
0	c1	2000	0.0	6.0	7.0
1	c1	2001	1.0	8.0	9.0
2	c1	2002	2.0	10.0	11.0
3	c2	2001	3.0	0.0	1.0

In [209…
```
left2=pd.DataFrame([[1.,2.],[3.,4.],[5.,6.]], index=['a','c','e'],columns=['c1','c
```

In [210…
```
right2=pd.DataFrame([[7.,8.],[9.,10.],[11.,12.],[13.,14.]], index=['b','c','d','e'
```

In [211…
```
left2
```

Out[211]:

	c1	c2
a	1.0	2.0
c	3.0	4.0
e	5.0	6.0

In [212…
```
right2
```

Out[212]:

	c3	c4
b	7.0	8.0
c	9.0	10.0
d	11.0	12.0
e	13.0	14.0

In [215…
```
pd.merge(left2,right2,how='outer',left_index=True, right_index=True)
```

Out[215]:

	c1	c2	c3	c4
a	1.0	2.0	NaN	NaN
b	NaN	NaN	7.0	8.0
c	3.0	4.0	9.0	10.0
d	NaN	NaN	11.0	12.0
e	5.0	6.0	13.0	14.0

In [216…
```
left2.join(right2,how='outer')
```

Out[216]:

	c1	c2	c3	c4
a	1.0	2.0	NaN	NaN
b	NaN	NaN	7.0	8.0
c	3.0	4.0	9.0	10.0
d	NaN	NaN	11.0	12.0
e	5.0	6.0	13.0	14.0

In [217…
```
left1.join(right1,on='key')
```

Out[217]:

	key	value	group_val
0	a	0	3.5
1	b	1	7.0
2	a	2	3.5
3	a	3	3.5
4	b	4	7.0
5	c	5	NaN

In [218…
```
new_df=pd.DataFrame([[7.,8.],[9.,10.],[11.,12.],[16.,17.]],index=['a','c','e','f']
```

In [219…
```
left2
```

Out[219]:

	c1	c2
a	1.0	2.0
c	3.0	4.0
e	5.0	6.0

In [220…
```
right2
```

Out[220]:

	c3	c4
b	7.0	8.0
c	9.0	10.0
d	11.0	12.0
e	13.0	14.0

In [221…
```
left2.join([right2,new_df])
```

Out[221]:

	c1	c2	c3	c4	c5	c6
a	1.0	2.0	NaN	NaN	7.0	8.0
c	3.0	4.0	9.0	10.0	9.0	10.0
e	5.0	6.0	13.0	14.0	11.0	12.0

In [222…
```
new_df
```

Out[222]:

	c5	c6
a	7.0	8.0
c	9.0	10.0
e	11.0	12.0
f	16.0	17.0

In [223…
```
left2.join([right2,new_df],how='outer')
```

Out[223]:

	c1	c2	c3	c4	c5	c6
a	1.0	2.0	NaN	NaN	7.0	8.0
c	3.0	4.0	9.0	10.0	9.0	10.0
e	5.0	6.0	13.0	14.0	11.0	12.0
b	NaN	NaN	7.0	8.0	NaN	NaN
d	NaN	NaN	11.0	12.0	NaN	NaN
f	NaN	NaN	NaN	NaN	16.0	17.0

In [245…
```
arr=np.arange(12).reshape((3,4))
```

In [246…
```
arr
```

Out[246]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

In [247…
```
np.concatenate([arr,arr],axis=1)
```

Out[247]:
```
array([[ 0,  1,  2,  3,  0,  1,  2,  3],
       [ 4,  5,  6,  7,  4,  5,  6,  7],
       [ 8,  9, 10, 11,  8,  9, 10, 11]])
```

In [252…
```
s1=pd.Series([0,1],index=['a','b'])
print(s1)
```

```
a    0
b    1
dtype: int64
```

In [253…
```
s2=pd.Series([2,3,4],index=['c','d','e'])
print(s2)
```

```
c    2
d    3
e    4
dtype: int64
```

In [254…
```
s3=pd.Series([5,6],index=['f','g'])
print(s3)
```

```
f    5
g    6
dtype: int64
```

In [255…
```
pd.concat([s1,s2,s3])
```

Out[255]:
```
a    0
b    1
c    2
d    3
e    4
f    5
g    6
dtype: int64
```

In [256…
```
pd.concat([s1,s2,s3],axis=1)
```

Out[256]:

	0	1	2
a	0.0	NaN	NaN
b	1.0	NaN	NaN
c	NaN	2.0	NaN
d	NaN	3.0	NaN
e	NaN	4.0	NaN
f	NaN	NaN	5.0
g	NaN	NaN	6.0

In [257…
```
s4=pd.concat([s1*5,s3])
print(s4)
```

```
a    0
b    5
f    5
g    6
dtype: int64
```

In [262…
```
pd.concat([s1,s4],axis=1)
```

Out[262]:

	0	1
a	0.0	0
b	1.0	5
f	NaN	5
g	NaN	6

In [260…
```
s4
```

Out[260]:
```
a    0
b    5
f    5
g    6
dtype: int64
```

In [263…
```
pd.concat([s1,s4],axis=1,join='inner')
```

Out[263]:

	0	1
a	0	0
b	1	5

In [265…
```
result=pd.concat([s1,s1,s3],keys=['one','two','three'])
```

In [266…
```
result
```

Out[266]:
```
one    a    0
       b    1
two    a    0
       b    1
three  f    5
       g    6
dtype: int64
```

In [267…
```
result.unstack()
```

Out[267]:

	a	b	f	g
one	0.0	1.0	NaN	NaN
two	0.0	1.0	NaN	NaN
three	NaN	NaN	5.0	6.0

In [272…
```
pd.concat([s1,s1,s3],axis=1,keys=['one','two','three'])
```

Out[272]:

	one	two	three
a	0.0	0.0	NaN
b	1.0	1.0	NaN
f	NaN	NaN	5.0
g	NaN	NaN	6.0

In [276…
```
df1=pd.DataFrame(np.arange(6).reshape(3,2),index=['a','b','c'],columns=['one','two
```

In [277…
```
df1
```

Out[277]:

	one	two
a	0	1
b	2	3
c	4	5

In [278…
```
import numpy as np
```

In [280…
```
df2=pd.DataFrame(5+np.arange(4).reshape(2,2),index=['a','c'],columns=['three','four
```

In [281…
```
df2
```

Out[281]:

	three	four
a	5	6
c	7	8

In [284…
```
pd.concat({'level1':df1,'level2':df2},axis=1)
```

Out[284]:

	level1		level2	
	one	two	three	four
a	0	1	5.0	6.0
b	2	3	NaN	NaN
c	4	5	7.0	8.0

In [ ]:
```
```

In [288…
```
df5=pd.concat([df1,df2],axis=1,keys=['level1','level2'],names=['upper','lower'])
print(df5)
```

```
upper    level1                            level2
lower         a         b         c         d  three four
0     -0.564101  0.214955  0.537704  0.085733    NaN  NaN
1      0.918373 -2.616679  1.300794 -0.047741    NaN  NaN
2     -0.782104  0.562508  0.926161 -0.433202    NaN  NaN
a           NaN       NaN       NaN       NaN    5.0  6.0
c           NaN       NaN       NaN       NaN    7.0  8.0
```

In [289…
```
df1=pd.DataFrame(np.random.randn(3,4),columns=['a','b','c','d'])
print(df1)
```

```
          a         b         c         d
0 -0.386629  0.108284 -0.629828  0.678859
1  2.305422  2.128106 -0.492602 -0.789086
2  1.944952 -0.763417 -0.109988 -1.737148
```

In [290…
```
df2=pd.DataFrame(np.random.randn(2,3),columns=['b','d','a'])
print(df2)
```

```
          b         d         a
0  0.785574  0.585834  1.091273
1 -0.529958  1.319749  0.444525
```

In [291…
```
df5.unstack()
```

Out[291]:
```
upper   lower
level1  a      0   -0.564101
               1    0.918373
               2   -0.782104
               a         NaN
               c         NaN
        b      0    0.214955
               1   -2.616679
               2    0.562508
               a         NaN
               c         NaN
        c      0    0.537704
               1    1.300794
               2    0.926161
               a         NaN
               c         NaN
        d      0    0.085733
               1   -0.047741
               2   -0.433202
               a         NaN
               c         NaN
level2  three  0         NaN
               1         NaN
               2         NaN
               a    5.000000
               c    7.000000
        four   0         NaN
               1         NaN
               2         NaN
               a    6.000000
               c    8.000000
dtype: float64
```

In [292…
```
pd.concat([df1,df2])
```

Out[292]:

	a	b	c	d
0	-0.386629	0.108284	-0.629828	0.678859
1	2.305422	2.128106	-0.492602	-0.789086
2	1.944952	-0.763417	-0.109988	-1.737148
0	1.091273	0.785574	NaN	0.585834
1	0.444525	-0.529958	NaN	1.319749

In [293…
```
a=pd.Series([np.nan,2.5,np.nan,3.5,4.6,np.nan],index=list('fedcba'))
```

In [294…
```
b=pd.Series(np.arange(len(a),dtype=np.float64) ,index=list('fedcba'))
```

In [295…
```
a
```

Out[295]:
```
f    NaN
e    2.5
d    NaN
c    3.5
b    4.6
a    NaN
dtype: float64
```

In [296…
```
b
```

Out[296]:
```
f    0.0
e    1.0
d    2.0
c    3.0
b    4.0
a    5.0
dtype: float64
```

In [297…
```
b[-1]=np.nan
```

In [304…
```
b
```

Out[304]:
```
f    0.0
e    1.0
d    2.0
c    3.0
b    4.0
a    NaN
dtype: float64
```

In [299…
```
np.where(pd.isnull(a),b,a)
```

Out[299]:
```
array([0. , 2.5, 2. , 3.5, 4.6, nan])
```

In [303…
```
b[:-2].combine_first(a[2:])
```

Out[303]:
```
a    NaN
b    4.6
c    3.0
d    2.0
e    1.0
f    0.0
dtype: float64
```

In [300…
```
df1=pd.DataFrame({'a':[1.,np.nan,5.,np.nan],'b':[np.nan,2.,np.nan,6.],'c':range(2,
```

```
print(df1)
```

```
     a    b   c
0  1.0  NaN   2
1  NaN  2.0   6
2  5.0  NaN  10
3  NaN  6.0  14
```

In [301…
```
df2=pd.DataFrame({'a':[5.,4.,np.nan,3.,7.],'b':[np.nan,3.,4.,6.,8.]})
print(df2)
```

```
     a    b
0  5.0  NaN
1  4.0  3.0
2  NaN  4.0
3  3.0  6.0
4  7.0  8.0
```

In [302…
```
df1.combine_first(df2)
```

Out[302]:

	a	b	c
0	1.0	NaN	2.0
1	4.0	2.0	6.0
2	5.0	4.0	10.0
3	3.0	6.0	14.0
4	7.0	8.0	NaN

reshaping and pivoting

In [307…
```
Data=pd.DataFrame(np.arange(6).reshape(2,3),index=pd.Index(['c1','c2'],name='city'
```

In [308…
```
Data
```

Out[308]:

number	one	two	three
city			
c1	0	1	2
c2	3	4	5

In [309…
```
result=Data.stack()
```

In [310…
```
result
```

Out[310]:
```
city  number
c1    one       0
      two       1
      three     2
c2    one       3
      two       4
      three     5
dtype: int32
```

In [311…
```
result.unstack()
```

Out[311]:

number	one	two	three
city			
c1	0	1	2
c2	3	4	5

In [312…
```
result.unstack(0)
```

Out[312]:

city	c1	c2
number		
one	0	3
two	1	4
three	2	5

In [318…
```
result.unstack('city')
```

Out[318]:

city	c1	c2
number		
one	0	3
two	1	4
three	2	5

In [319…
```
s1=pd.Series([0,1,2,3],index=['a','b','c','d'])
```

In [320…
```
s2=pd.Series([4,5,6],index=['c','d','e'])
```

In [322…
```
data2=pd.concat([s1,s2],keys=['one','two'])
```

In [323…
```
data2
```

Out[323]:
```
one  a    0
     b    1
     c    2
     d    3
two  c    4
     d    5
     e    6
dtype: int64
```

In [324…
```
data2.unstack()
```

Out[324]:

	a	b	c	d	e
one	0.0	1.0	2.0	3.0	NaN
two	NaN	NaN	4.0	5.0	6.0

In [325…
```
data2.unstack().stack()
```

Out[325]:
```
one  a    0.0
     b    1.0
     c    2.0
     d    3.0
two  c    4.0
     d    5.0
     e    6.0
dtype: float64
```

In [326…
```
data2.unstack().stack(dropna=False)
```

Out[326]:
```
one  a    0.0
     b    1.0
     c    2.0
     d    3.0
     e    NaN
two  a    NaN
     b    NaN
     c    4.0
     d    5.0
     e    6.0
dtype: float64
```

In [327…
```
df=pd.DataFrame({'left':result,'right':result+5},columns=pd.Index(['left','right'],
```

In [328…
```
df
```

Out[328]:

		side	left	right
city	number			
c1	one		0	5
	two		1	6
	three		2	7
c2	one		3	8
	two		4	9
	three		5	10

In [329…
```
df.unstack('city')
```

Out[329]:

side	left		right	
city	c1	c2	c1	c2
number				
one	0	3	5	8
two	1	4	6	9
three	2	5	7	10

In [330…
```
df.unstack('city').stack('side')
```

Out[330]:

		city	c1	c2
number	side			
one	left	0	3
	right	5	8
two	left	1	4
	right	6	9
three	left	2	5
	right	7	10

data transformation (removing duplicates)

In [331…
```
data=pd.DataFrame({'k1':['one']*3+['two']*4,'k2':[1,1,2,3,3,4,4]})
```

In [332…
```
data
```

Out[332]:

	k1	k2
0	one	1
1	one	1
2	one	2
3	two	3
4	two	3
5	two	4
6	two	4

In [333…
```
data.duplicated()
```

Out[333]:
```
0    False
1     True
2    False
3    False
4     True
5    False
6     True
dtype: bool
```

In [334…
```
data.drop_duplicates()
```

Out[334]:

	k1	k2
0	one	1
2	one	2
3	two	3
5	two	4

Discretization and binning

In [335…
```
ages=[20,22,25,27,21,23,37,31,61,45,41,32]
```

In [336…
```
bins=[18,25,35,60,100]
```

In [337…
```
catego=pd.cut(ages,bins)
```

In [338…
```
import pandas as pd
```

In [339…
```
catego
```

Out[339]:
```
[(18, 25], (18, 25], (18, 25], (25, 35], (18, 25], ..., (25, 35], (60, 100], (35, 60], (35, 60], (25, 35]]
Length: 12
Categories (4, interval[int64, right]): [(18, 25] < (25, 35] < (35, 60] < (60, 100]]
```

In [340…
```
pd.value_counts(catego)
```

Out[340]:
```
(18, 25]     5
(25, 35]     3
(35, 60]     3
(60, 100]    1
dtype: int64
```

In [341…
```
catego.codes
```

Out[341]:
```
array([0, 0, 0, 1, 0, 0, 2, 1, 3, 2, 2, 1], dtype=int8)
```

In [342…
```
catego.categories
```

Out[342]:
```
IntervalIndex([(18, 25], (25, 35], (35, 60], (60, 100]], dtype='interval[int64, right]')
```

In [345…
```
catego=pd.cut(ages,bins, right=False)
catego
```

Out[345]:
```
[[18, 25), [18, 25), [25, 35), [25, 35), [18, 25), ..., [25, 35), [60, 100), [35, 60), [35, 60), [25, 35)]
Length: 12
Categories (4, interval[int64, left]): [[18, 25) < [25, 35) < [35, 60) < [60, 100)]
```

In [344…
```
catego
```

Out[344]:
```
[[18, 25), [18, 25), [25, 35), [25, 35), [18, 25), ..., [25, 35), [60, 100), [35, 60), [35, 60), [25, 35)]
Length: 12
Categories (4, interval[int64, left]): [[18, 25) < [25, 35) < [35, 60) < [60, 100)]
```

In [346…
```
import numpy as np
data=np.random.rand(20)
```

In [347…
```
data1=pd.cut(data,4,precision=2)
data1
```

Out[347]:
```
[(0.0074, 0.24], (0.0074, 0.24], (0.0074, 0.24], (0.72, 0.95], (0.72, 0.95], ...,
(0.48, 0.72], (0.24, 0.48], (0.48, 0.72], (0.72, 0.95], (0.0074, 0.24]]
Length: 20
Categories (4, interval[float64, right]): [(0.0074, 0.24] < (0.24, 0.48] < (0.48,
0.72] < (0.72, 0.95]]
```

In [348…
```
pd.value_counts(data1)
```

Out[348]:
```
(0.0074, 0.24]    6
(0.24, 0.48]      5
(0.72, 0.95]      5
(0.48, 0.72]      4
dtype: int64
```

In [349…
```
data_n=pd.cut(np.random.randn(10),4)
```

In [350…
```
data_n
```

Out[350]:
```
[(0.736, 1.367], (-1.158, -0.525], (0.736, 1.367], (0.106, 0.736], (0.106, 0.736],
(-1.158, -0.525], (0.736, 1.367], (0.106, 0.736], (-1.158, -0.525], (0.736, 1.36
7]]
Categories (4, interval[float64, right]): [(-1.158, -0.525] < (-0.525, 0.106] <
(0.106, 0.736] < (0.736, 1.367]]
```

In [351…
```
pd.value_counts(data_n)
```

Out[351]:
```
(0.736, 1.367]      4
(-1.158, -0.525]    3
(0.106, 0.736]      3
(-0.525, 0.106]     0
dtype: int64
```

In [352…
```
data_n.value_counts()
```

Out[352]:
```
(-1.158, -0.525]    3
(-0.525, 0.106]     0
(0.106, 0.736]      3
(0.736, 1.367]      4
dtype: int64
```

In [353…
```
data_n=pd.cut(np.random.randn(1000),4)
```

In [354…
```
pd.value_counts(data_n)
```

Out[354]:
```
(-0.107, 1.402]     456
(-1.616, -0.107]    411
(1.402, 2.912]       78
(-3.132, -1.616]     55
dtype: int64
```

In [355…
```
cat1=pd.qcut(data,4)
```

In [356…
```
cat1
```

Out[356]:
```
[(0.00738, 0.24], (0.00738, 0.24], (0.24, 0.441], (0.687, 0.951], (0.687, 0.951],
..., (0.441, 0.687], (0.441, 0.687], (0.441, 0.687], (0.687, 0.951], (0.00738, 0.2
4]]
Length: 20
Categories (4, interval[float64, right]): [(0.00738, 0.24] < (0.24, 0.441] < (0.44
1, 0.687] < (0.687, 0.951]]
```

In [357…
```
pd.value_counts(cat1)
```

Out[357]:
```
(0.00738, 0.24]    5
(0.24, 0.441]      5
(0.441, 0.687]     5
(0.687, 0.951]     5
dtype: int64
```

In [358…
```
data11=np.arange(20)
```

In [370…
```
cat1_c=pd.cut(data11,6)
```

In [371…
```
pd.value_counts(cat1_c)
```

Out[371]:
```
(-0.019, 3.167]     4
(15.833, 19.0]      4
(3.167, 6.333]      3
(6.333, 9.5]        3
(9.5, 12.667]       3
(12.667, 15.833]    3
dtype: int64
```

In [361…
```
cat1_qc=pd.qcut(data11,4)
```

In [362…
```
pd.value_counts(cat1_qc)
```

Out[362]:
```
(-0.001, 4.75]    5
(4.75, 9.5]       5
(9.5, 14.25]      5
(14.25, 19.0]     5
dtype: int64
```

In [363…
```
cat1_qc1=pd.qcut(data11,[0,0.3,0.5,0.7,1])
```

In [364…
```
pd.value_counts(cat1_qc1)
```

Out[364]:
```
(-0.001, 5.7]    6
(13.3, 19.0]     6
(5.7, 9.5]       4
(9.5, 13.3]      4
dtype: int64
```

In [365…
```
cat1_c1=pd.cut(data11,[0,0.3,0.5,0.7,1])
```

In [366…
```
pd.value_counts(cat1_c1)
```

Out[366]:
```
(0.7, 1.0]    1
(0.0, 0.3]    0
(0.3, 0.5]    0
(0.5, 0.7]    0
dtype: int64
```

In [372…
```
cat1_c1=pd.cut(data11,[0,3,5,7,10,20])
```

In [373…
```
pd.value_counts(cat1_c1)
```

Out[373]:
```
(10, 20]    9
(0, 3]      3
(7, 10]     3
(3, 5]      2
(5, 7]      2
dtype: int64
```

In [ ]: