GroupBy

In [167]:
```
import numpy as np
import pandas as pd
```

In [168]:
```
city = ['Kermanshah', 'Hamedan', 'Oromieh', 'Mashad', 'Yazd', 'Kerman','Zabol
```

In [169]:
```
myser = pd.Series([2, 3, 1, 6, 4, 5, 1], index=city)
myser
```

Out[169]:
```
Kermanshah    2
Hamedan       3
Oromieh       1
Mashad        6
Yazd          4
Kerman        5
Zabol         1
dtype: int64
```

In [170]:
```
k = ['W', 'W', 'W', 'E', 'E', 'E', 'E']
myser.groupby(k).max()
```

Out[170]:
```
E    6
W    3
dtype: int64
```

In [171]:
```
myser[['Oromieh', 'Yazd','Zabol']] = np.nan
myser
```

Out[171]:
```
Kermanshah    2.0
Hamedan       3.0
Oromieh       NaN
Mashad        6.0
Yazd          NaN
Kerman        5.0
Zabol         NaN
dtype: float64
```

In [172]:
```
myser.groupby(k).mean()
```

Out[172]:
```
E    5.5
W    2.5
dtype: float64
```

In [173]:

```
f = lambda g: g.fillna(g.mean())
myser.groupby(k).apply(f)
```

Out[173]:

```
Kermanshah    2.0
Hamedan       3.0
Oromieh       2.5
Mashad        6.0
Yazd          5.5
Kerman        5.0
Zabol         5.5
dtype: float64
```

In [174]:

```
f = {'W': 1, 'E': 2}
c = lambda g: g.fillna(f[g.name])
myser.groupby(k).apply(c)
```

Out[174]:

```
Kermanshah    2.0
Hamedan       3.0
Oromieh       1.0
Mashad        6.0
Yazd          2.0
Kerman        5.0
Zabol         2.0
dtype: float64
```

In [175]:

```
#
```

In [176]:

```
df = pd.DataFrame({
        'key1' : ['ali', 'ali', 'ali', 'sara', 'sara', 'sara', 'sara'],
        'key2' : ['one', 'one', 'two', 'one', 'one', 'two', 'two'],
        'data' : [12, 16, 13, 20, 8, 17, 10]
})

df
```

Out[176]:

	key1	key2	data
0	ali	one	12
1	ali	one	16
2	ali	two	13
3	sara	one	20
4	sara	one	8
5	sara	two	17
6	sara	two	10

In [177]:

```
g = df.groupby('key1')
```

In [178]:

```
g.describe()
```

Out[178]:

	data							
	count	mean	std	min	25%	50%	75%	max
key1								
ali	3.0	13.666667	2.081666	12.0	12.5	13.0	14.50	16.0
sara	4.0	13.750000	5.678908	8.0	9.5	13.5	17.75	20.0

In [179]:

```
g.max()
```

Out[179]:

	key2	data
key1		
ali	two	16
sara	two	20

In [180]:

```
g.min()
```

Out[180]:

	key2	data
key1		
ali	one	12
sara	one	8

In [181]:

```
def f(t):
    return t.max() - t.min()
```

In [182]:

```
g.agg(f)
```

Out[182]:

	data
key1	
ali	4
sara	12

In [183]:

```
df
```

Out[183]:

	key1	key2	data
0	ali	one	12
1	ali	one	16
2	ali	two	13
3	sara	one	20
4	sara	one	8
5	sara	two	17
6	sara	two	10

In [184]:

```
d = dict(list(df.groupby('key1')))
d['ali']
```

Out[184]:

	key1	key2	data
0	ali	one	12
1	ali	one	16
2	ali	two	13

In [185]:

```
d['sara']
```

Out[185]:

	key1	key2	data
3	sara	one	20
4	sara	one	8
5	sara	two	17
6	sara	two	10

In [186]:

```
df['data'].groupby(df['key1']).min()
```

Out[186]:

```
key1
ali     12
sara     8
Name: data, dtype: int64
```

In [187]:

```
df
```

Out[187]:

```
   key1  key2  data
0   ali   one    12
1   ali   one    16
2   ali   two    13
3  sara   one    20
4  sara   one     8
5  sara   two    17
6  sara   two    10
```

In [188]:

```
h = df.groupby(['key1', 'key2'])
```

In [189]:

```
h.max()
```

Out[189]:

```
            data
key1  key2
 ali   one    16
       two    13
sara   one    20
       two    17
```

Grouping by Index Levels

In [190]:

```
arr = np.array([[11, 12, 16, 4, 15],
                [17, 2, 18, 19, 10],
                [7, 15, 13, 14, 11],
                [8, 17, 13, 20, 12]])
```

In [191]:

```
mi = pd.MultiIndex.from_arrays([['Ali', 'Ali', 'Ali', 'Sara', 'Sara'],
                                [1, 2, 3, 1, 2]],
                                names=['X', 'Y'])
mi
```

Out[191]:

```
MultiIndex([( 'Ali', 1),
            ( 'Ali', 2),
            ( 'Ali', 3),
            ('Sara', 1),
            ('Sara', 2)],
           names=['X', 'Y'])
```

In [192]:

```
mydf = pd.DataFrame(arr, columns=mi)
mydf
```

Out[192]:

X	Ali			Sara	
Y	1	2	3	1	2
0	11	12	16	4	15
1	17	2	18	19	10
2	7	15	13	14	11
3	8	17	13	20	12

In [193]:

```
mydf.groupby(level='X', axis=1).max()
```

Out[193]:

X	Ali	Sara
0	16	15
1	18	19
2	15	14
3	17	20

In [194]:

```
# cut
```

In [195]:

```
score = [16, 12, 13, 14, 20, 16, 17, 5, 19, 7]
sc = pd.cut(score, 4, labels=['Q1', 'Q2', 'Q3', 'Q4'])
sc
```

Out[195]:

```
['Q3', 'Q2', 'Q3', 'Q3', 'Q4', 'Q3', 'Q4', 'Q1', 'Q4', 'Q1']
Categories (4, object): ['Q1' < 'Q2' < 'Q3' < 'Q4']
```

In [196]:

```
s1 = pd.Series(score)
s2 = pd.Series(sc)
```

In [197]:

```
s1.groupby(s2).agg(['min', 'count']).reset_index()
```

Out[197]:

	index	min	count
0	Q1	5	2
1	Q2	12	1
2	Q3	13	4
3	Q4	17	3

In [198]:

```
#
```

In [199]:

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9]
b = [11, 12, 13, 14, 15, 16, 17, 18, 19]

df = pd.DataFrame({'col1': a, 'col2': b})
df
```

Out[199]:

	col1	col2
0	1	11
1	2	12
2	3	13
3	4	14
4	5	15
5	6	16
6	7	17
7	8	18
8	9	19

In [200]:

```
q = pd.cut(df.col1, 4)
q
```

Out[200]:

```
0    (0.992, 3.0]
1    (0.992, 3.0]
2    (0.992, 3.0]
3      (3.0, 5.0]
4      (3.0, 5.0]
5      (5.0, 7.0]
6      (5.0, 7.0]
7      (7.0, 9.0]
8      (7.0, 9.0]
Name: col1, dtype: category
Categories (4, interval[float64]): [(0.992, 3.0] < (3.0, 5.0] < (5.0, 7.0]
< (7.0, 9.0]]
```

In [201]:

```
def myfunc(g):
    return {
             'max':   g.max(),
             'count': g.count(),
           }
```

In [202]:

```
g = df.col2.groupby(q)
g.apply(myfunc)
```

Out[202]:

```
col1
(0.992, 3.0]  max      13
              count     3
(3.0, 5.0]    max      15
              count     2
(5.0, 7.0]    max      17
              count     2
(7.0, 9.0]    max      19
              count     2
Name: col2, dtype: int64
```

In [203]:

```
g = df.col2.groupby(q)
g.apply(myfunc).unstack()
```

Out[203]:

	max	count
col1		
(0.992, 3.0]	13	3
(3.0, 5.0]	15	2
(5.0, 7.0]	17	2
(7.0, 9.0]	19	2

transform

In [204]:

```
n = ['ali', 'ali', 'ali', 'ali', 'sara', 'sara', 'sara', 'taha', 'taha']
s = [11, 20, 13, 14, 15, 6, 12, 18, 19]

df = pd.DataFrame({'name': n, 'score': s})
df
```

Out[204]:

	name	score
0	ali	11
1	ali	20
2	ali	13
3	ali	14
4	sara	15
5	sara	6
6	sara	12
7	taha	18
8	taha	19

In [205]:

```
g = df.groupby('name').score
```

In [206]:

```
g.max()
```

Out[206]:

```
name
ali     20
sara    15
taha    19
Name: score, dtype: int64
```

In [207]:

```
g.count()
```

Out[207]:

```
name
ali     4
sara    3
taha    2
Name: score, dtype: int64
```

In [208]:

```
g.transform('max')
```

Out[208]:

```
0    20
1    20
2    20
3    20
4    15
5    15
6    15
7    19
8    19
Name: score, dtype: int64
```

In [209]:

```
g.transform(lambda x: x.max())
```

Out[209]:

```
0    20
1    20
2    20
3    20
4    15
5    15
6    15
7    19
8    19
Name: score, dtype: int64
```

In [210]:

```
g.transform(lambda x: x - 1)
```

Out[210]:

```
0    10
1    19
2    12
3    13
4    14
5     5
6    11
7    17
8    18
Name: score, dtype: int64
```

In [211]:

```
g.transform('mean')
```

Out[211]:

```
0    14.5
1    14.5
2    14.5
3    14.5
4    11.0
5    11.0
6    11.0
7    18.5
8    18.5
Name: score, dtype: float64
```

In [212]:

```
(df['score'] - g.transform('mean')) / g.transform('std')
```

Out[212]:

```
0   -0.903696
1    1.420094
2   -0.387298
3   -0.129099
4    0.872872
5   -1.091089
6    0.218218
7   -0.707107
8    0.707107
Name: score, dtype: float64
```

example

In [213]:

```
df = pd.read_csv('iris.csv')
df
```

Out[213]:

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

In [214]:

```
df.groupby(['variety']).agg('min')
```

Out[214]:

	sepal.length	sepal.width	petal.length	petal.width
variety				
Setosa	4.3	2.3	1.0	0.1
Versicolor	4.9	2.0	3.0	1.0
Virginica	4.9	2.2	4.5	1.4

In [215]:

```
def myfunc(f, n=2):
    return  f.sort_values(by='sepal.length')[:n]
```

In [216]:

```
myfunc(df, 8)
```

Out[216]:

	sepal.length	sepal.width	petal.length	petal.width	variety
13	4.3	3.0	1.1	0.1	Setosa
42	4.4	3.2	1.3	0.2	Setosa
38	4.4	3.0	1.3	0.2	Setosa
8	4.4	2.9	1.4	0.2	Setosa
41	4.5	2.3	1.3	0.3	Setosa
22	4.6	3.6	1.0	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
6	4.6	3.4	1.4	0.3	Setosa

In [217]:

```
df.groupby(['variety']).apply(myfunc)
```

Out[217]:

		sepal.length	sepal.width	petal.length	petal.width	variety
variety						
Setosa	**13**	4.3	3.0	1.1	0.1	Setosa
	8	4.4	2.9	1.4	0.2	Setosa
Versicolor	**57**	4.9	2.4	3.3	1.0	Versicolor
	60	5.0	2.0	3.5	1.0	Versicolor
Virginica	**106**	4.9	2.5	4.5	1.7	Virginica
	121	5.6	2.8	4.9	2.0	Virginica

category

In [218]:

```
t = df['variety']
t
```

Out[218]:

```
0         Setosa
1         Setosa
2         Setosa
3         Setosa
4         Setosa
         ...    
145    Virginica
146    Virginica
147    Virginica
148    Virginica
149    Virginica
Name: variety, Length: 150, dtype: object
```

In [219]:

```
c = t.astype('category')
c
```

Out[219]:

```
0         Setosa
1         Setosa
2         Setosa
3         Setosa
4         Setosa
         ...    
145    Virginica
146    Virginica
147    Virginica
148    Virginica
149    Virginica
Name: variety, Length: 150, dtype: category
Categories (3, object): ['Setosa', 'Versicolor', 'Virginica']
```

In [220]:

```
c.values.codes
```

Out[220]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2], dtype=int8)
```

In [221]:

```
c.value_counts()
```

Out[221]:

```
Setosa        50
Versicolor    50
Virginica     50
Name: variety, dtype: int64
```

In [222]:

```
c.values.categories
```

Out[222]:

```
Index(['Setosa', 'Versicolor', 'Virginica'], dtype='object')
```

In [223]:

```
c.isin(['Setosa'])
```

Out[223]:

```
0       True
1       True
2       True
3       True
4       True
       ...  
145    False
146    False
147    False
148    False
149    False
Name: variety, Length: 150, dtype: bool
```

In [224]:

```
x = c[c.isin(['Setosa'])]
x
```

Out[224]:

```
0     Setosa
1     Setosa
2     Setosa
3     Setosa
4     Setosa
5     Setosa
6     Setosa
7     Setosa
8     Setosa
9     Setosa
10    Setosa
11    Setosa
12    Setosa
13    Setosa
14    Setosa
15    Setosa
16    Setosa
17    Setosa
18    Setosa
19    Setosa
20    Setosa
21    Setosa
22    Setosa
23    Setosa
24    Setosa
25    Setosa
26    Setosa
27    Setosa
28    Setosa
29    Setosa
30    Setosa
31    Setosa
32    Setosa
33    Setosa
34    Setosa
35    Setosa
36    Setosa
37    Setosa
38    Setosa
39    Setosa
40    Setosa
41    Setosa
42    Setosa
43    Setosa
44    Setosa
45    Setosa
46    Setosa
47    Setosa
48    Setosa
49    Setosa
Name: variety, dtype: category
Categories (3, object): ['Setosa', 'Versicolor', 'Virginica']
```

In [225]:

```
y = x.cat.remove_unused_categories()
y
```

Out[225]:

```
0     Setosa
1     Setosa
2     Setosa
3     Setosa
4     Setosa
5     Setosa
6     Setosa
7     Setosa
8     Setosa
9     Setosa
10    Setosa
11    Setosa
12    Setosa
13    Setosa
14    Setosa
15    Setosa
16    Setosa
17    Setosa
18    Setosa
19    Setosa
20    Setosa
21    Setosa
22    Setosa
23    Setosa
24    Setosa
25    Setosa
26    Setosa
27    Setosa
28    Setosa
29    Setosa
30    Setosa
31    Setosa
32    Setosa
33    Setosa
34    Setosa
35    Setosa
36    Setosa
37    Setosa
38    Setosa
39    Setosa
40    Setosa
41    Setosa
42    Setosa
43    Setosa
44    Setosa
45    Setosa
46    Setosa
47    Setosa
48    Setosa
49    Setosa
Name: variety, dtype: category
Categories (1, object): ['Setosa']
```

دانشگاه شهید مدنی آذربایجان
برنامه نویسی پیشرفته با پایتون
امین گلزاری اسکوئی
١٤٠٠-١٤٠١

Codes and Projects (click here) (https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Advanced-2021) slides and videos (click here) (https://drive.google.com/drive/folders/1Dx3v7fD1QBWL-MNP2hd7iIxaRbeALkkA)