In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
# reading data set
data1 = pd.read_csv("sets/records.csv")
data1
```

Out[2]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [ ]:

```
# GroupingBy Function

# Pandas GroupingBy Function  means splitting of data into smaller groups based on smoe cri

# syntax  => DataFrame.groupby()
```

In [ ]:

```
# GroupingBy Function involves some operations
# Splitting the dataset - Apply functions - Combine results
```

In [ ]:

```
# data1.groupby()
'''
Signature:
data1.groupby(

    by=None,
    axis=0,
    level=None,
    as_index=True,
    sort=True,
    group_keys=True,
    squeeze=False,
    observed=False,
    **kwargs,
)

'''
```

In [3]:

```
res1 = data1.groupby(by='Section')
res1
```

Out[3]:

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000216A72D6128>
```

In [7]:

```
for code1,code2 in res1:
    print(code1)
    print(code2)
    print();print();print()
```

```
A
   Student ID Section  Class  Study hrs  Social Media usage hrs  Percentage
0        1001       A     10          2                       3          50
2        1003       A     10          3                       2          60



B
   Student ID Section  Class  Study hrs  Social Media usage hrs  Percentage
1        1002       B     10          6                       2          80



C
   Student ID Section  Class  Study hrs  Social Media usage hrs  Percentage
3        1004       C     11          0                       1          45
4        1005       C     12          5                       2          75
```

In [ ]:

```

```

In [11]:

```
data1
```

Out[11]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [8]:

```
res1 = data1.groupby(by='Section')
res1
```

Out[8]:

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000216A72DF978>
```

In [10]:

```
res1.groups
```

Out[10]:

```
{'A': Int64Index([0, 2], dtype='int64'),
 'B': Int64Index([1], dtype='int64'),
 'C': Int64Index([3, 4], dtype='int64')}
```

In [ ]:

```

```

In [12]:

```
# Grouping Multiple Columns
data1
```

Out[12]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [13]:

```
res2 = data1.groupby(['Section','Class'])
res2
```

Out[13]:

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000216A772E6D8>
```

In [14]:

```
res2.groups
```

Out[14]:

```
{('A', 10): Int64Index([0, 2], dtype='int64'),
 ('B', 10): Int64Index([1], dtype='int64'),
 ('C', 11): Int64Index([3], dtype='int64'),
 ('C', 12): Int64Index([4], dtype='int64')}
```

In [15]:

```
list(data1)
```

Out[15]:

```
['Student ID',
 'Section',
 'Class',
 'Study hrs',
 'Social Media usage hrs',
 'Percentage']
```

In [ ]:

```

```

In [ ]:

```

```

In [16]:

```
data1
```

Out[16]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [17]:

```
data2 = data1.groupby('Class')
data2
```

Out[17]:

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000216A7743828>
```

In [18]:

```
data2.get_group(10)
```

Out[18]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60

In [ ]:

```

```

In [ ]:

```

```

In [19]:

```
data1
```

Out[19]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [20]:

```
data3 = data1.groupby('Section')
data3
```

Out[20]:

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000216A7743978>
```

In [23]:

```
data3.get_group('C')
```

Out[23]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [ ]:

```
```

In [ ]:

```
```

In [24]:

```
data1
```

Out[24]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
0	1001	A	10	2	3	50
1	1002	B	10	6	2	80
2	1003	A	10	3	2	60
3	1004	C	11	0	1	45
4	1005	C	12	5	2	75

In [25]:

```
data1.sum()
```

Out[25]:

```
Student ID                 5015
Section                   ABACC
Class                        53
Study hrs                    16
Social Media usage hrs       10
Percentage                  310
dtype: object
```

In [ ]:

```
```

In [26]:

```
# mean => eg 2,3,4,5,6,7  => 2+3+4+5+6+7 / 6
data1.mean()
```

Out[26]:

```
Student ID                 1003.0
Class                        10.6
Study hrs                     3.2
Social Media usage hrs        2.0
Percentage                   62.0
dtype: float64
```

In [ ]:

```
```

In [27]:

```
data1.describe()
```

Out[27]:

	Student ID	Class	Study hrs	Social Media usage hrs	Percentage
count	5.000000	5.000000	5.000000	5.000000	5.000000
mean	1003.000000	10.600000	3.200000	2.000000	62.000000
std	1.581139	0.894427	2.387467	0.707107	15.247951
min	1001.000000	10.000000	0.000000	1.000000	45.000000
25%	1002.000000	10.000000	2.000000	2.000000	50.000000
50%	1003.000000	10.000000	3.000000	2.000000	60.000000
75%	1004.000000	11.000000	5.000000	2.000000	75.000000
max	1005.000000	12.000000	6.000000	3.000000	80.000000

In [ ]:

```
```

In [ ]:

```
```

In [28]:

```
# multiple Agg Funx

data1.agg(['sum','max','min','mean'])
```

Out[28]:

	Student ID	Section	Class	Study hrs	Social Media usage hrs	Percentage
sum	5015.0	ABACC	53.0	16.0	10.0	310.0
max	1005.0	C	12.0	6.0	3.0	80.0
min	1001.0	A	10.0	0.0	1.0	45.0
mean	1003.0	NaN	10.6	3.2	2.0	62.0

In [ ]:

```
```

In [ ]:

```
```

In [30]:

```
# create new data set
data4 = pd.DataFrame({'X':['B','A','B','A'],'Y':[1,2,3,4]})
data4
```

Out[30]:

	X	Y
0	B	1
1	A	2
2	B	3
3	A	4

In [31]:

```
data4.groupby(['X'])
```

Out[31]:

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000216A774D7F0>
```

In [32]:

```
data4.groupby(['X']).sum()
```

Out[32]:

	Y
X	
A	6
B	4

In [ ]:

```

```

In [33]:

```
# sorting in reverse order
data4.groupby(['X'],sort=False).sum()
```

Out[33]:

	Y
X	
B	4
A	6

In [34]:

```
# # sorting in forword order (By default sort=True)
data4.groupby(['X'],sort=True).sum()
```

Out[34]:

X	Y
A	6
B	4

In [ ]:

```

```

In [ ]:

```

```

In [35]:

```
data4
```

Out[35]:

	X	Y
0	B	1
1	A	2
2	B	3
3	A	4

In [36]:

```
data4.agg(np.size)
```

Out[36]:

```
X    4
Y    4
dtype: int64
```

In [37]:

```
data4.agg([np.sum])
```

Out[37]:

	X	Y
sum	BABA	10

In [38]:

```
data4.agg([np.mean])
```

Out[38]:

	Y
mean	2.5

In [39]:

```
data4.agg([np.std])
```

Out[39]:

	Y
std	1.290994

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [41]:

```
# adv functions
data5 = pd.DataFrame({'X':['A','B','A','C','B','C','A','D','C','E'],'Y':[30,24,45,67,89,32,
data5
```

Out[41]:

	X	Y
0	A	30
1	B	24
2	A	45
3	C	67
4	B	89
5	C	32
6	A	45
7	D	78
8	C	12
9	E	98

In [45]:

```
data7 = data5.groupby(by="X")
data7.groups
```

Out[45]:

```
{'A': Int64Index([0, 2, 6], dtype='int64'),
 'B': Int64Index([1, 4], dtype='int64'),
 'C': Int64Index([3, 5, 8], dtype='int64'),
 'D': Int64Index([7], dtype='int64'),
 'E': Int64Index([9], dtype='int64')}
```

In [46]:

```
data7.get_group('A')
```

Out[46]:

	X	Y
0	A	30
2	A	45
6	A	45

In [47]:

```
data7.get_group('B')
```

Out[47]:

	X	Y
1	B	24
4	B	89

In [48]:

```
data7.get_group('C')
```

Out[48]:

	X	Y
3	C	67
5	C	32
8	C	12

In [49]:

```
data7.get_group('D')
```

Out[49]:

	X	Y
7	D	78

In [50]:

```
data7.get_group('E')
```

Out[50]:

	X	Y
9	E	98

In [ ]:

```

```

In [65]:

```
# get data which scores are greater then 44
data6 = pd.DataFrame({'X':['A','B','A','C','B','C','A','D','C','E'],'Y':[30,24,45,167,89,13
data6
```

Out[65]:

	X	Y
0	A	30
1	B	24
2	A	45
3	C	167
4	B	89
5	C	132
6	A	145
7	D	78
8	C	12
9	E	198

In [71]:

```
data8 = data6.groupby('Y')
```

In [78]:

```
data8.filter(lambda x: print(x))
```

```
   X    Y
0  A   30
2  A   45
6  A  145
   X   Y
1  B  24
4  B  89
   X    Y
3  C  167
5  C  132
8  C   12
   X   Y
7  D  78
   X    Y
9  E  198
```

Out[78]:

	X	Y

In [85]:

```
data8.filter(lambda res4: print(res4['Y']>40))
```

```
0    False
2     True
6     True
Name: Y, dtype: bool
1    False
4     True
Name: Y, dtype: bool
3     True
5     True
8    False
Name: Y, dtype: bool
7    True
Name: Y, dtype: bool
9    True
Name: Y, dtype: bool
```

Out[85]:

X	Y

In [ ]:

```

```