AP_Class8_3.9.17 |Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|U

Zeppelin

AP_Class8_3.9.17

default

FINISHED

```
%pyspark
import pandas as pd
import numpy as np
```

Took 28 sec. Last updated by anonymous at March 09 2017, 11:15:28 PM.

FINISHED

```
%pyspark
df = pd.DataFrame({'Key1':['a','a','b','b','a'],
                'Key2':['one','two','one','two','one'],
                'data1':np.random.randn(5),
                'data2':np.random.randn(5)
})
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:15:39 PM.

FINISHED

```
%pyspark
df
```

```
  Key1 Key2     data1     data2
0    a  one -0.432396 -0.350729
1    a  two -0.334794  0.500221
2    b  one  0.142265  0.743638
3    b  two  1.006378 -0.245366
4    a  one -1.329591 -2.275981
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:15:45 PM.

FINISHED

```
%pyspark
grouped = df['data1'].groupby(df['Key1'])
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:15:50 PM.

FINISHED

```
%pyspark
grouped
```

```
<pandas.core.groupby.SeriesGroupBy object at 0x000002604DB58400>
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:15:55 PM.

FINISHED

```
%pyspark
grouped.mean()
```

```
Key1
a   -0.698927
b    0.574321
Name: data1, dtype: float64
```

AP_Class8_3.9.17

FINISHED

```
%pyspark
means = df['data1'].groupby([df['Key1'],df['Key2']]).mean()
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:16:03 PM.

FINISHED

```
%pyspark
means
```

```
Key1  Key2
a     one    -0.880994
      two    -0.334794
b     one     0.142265
      two     1.006378
Name: data1, dtype: float64
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:16:08 PM.

FINISHED

```
%pyspark
means.unstack()
```

```
Key2       one       two
Key1
a    -0.880994 -0.334794
b     0.142265  1.006378
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:16:10 PM.

FINISHED

```
%pyspark
states = np.array(['ohio','california','california','ohio','ohio'])
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:16:14 PM.

FINISHED

```
%pyspark
years = np.array([2005,2005,2006,2005,2006])
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:16:19 PM.

FINISHED

```
%pyspark
df['data1'].groupby([states,years]).mean()
```

```
california  2005   -0.334794
            2006    0.142265
ohio        2005    0.286991
            2006   -1.329591
Name: data1, dtype: float64
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:16:23 PM.

FINISHED

```
groupby('Key1').mean()

            data1     data2
Key1
a       -0.698927 -0.708829
b        0.574321  0.249136
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:16:26 PM.

FINISHED

```
%pyspark
df.groupby(['Key1','Key2']).mean()
```

```
              data1     data2
Key1 Key2
a    one  -0.880994 -1.313355
     two  -0.334794  0.500221
b    one   0.142265  0.743638
     two   1.006378 -0.245366
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:16:29 PM.

FINISHED

```
%pyspark
df.groupby(['Key1','Key2']).size()
```

```
Key1  Key2
a     one     2
      two     1
b     one     1
      two     1
dtype: int64
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:16:31 PM.

FINISHED

```
%pyspark
for name,group in df.groupby('Key1'):
    print (name)
    print (group)
```

```
a
  Key1 Key2     data1     data2
0    a  one -0.432396 -0.350729
1    a  two -0.334794  0.500221
4    a  one -1.329591 -2.275981
b
  Key1 Key2     data1     data2
2    b  one  0.142265  0.743638
3    b  two  1.006378 -0.245366
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:18:36 PM.

FINISHED

```
%pyspark
for (K1,K2),group in df.groupby(['Key1','Key2']):
    print (K1,K2)
    print (group)
```

```
a one
  Key1 Key2     data1     data2
0    a  one -0.432396 -0.350729
4    a  one -1.329591 -2.275981
a two
  Key1 Key2     data1     data2
1    a  two -0.334794  0.500221
b one
  Key1 Key2     data1     data2
2    b  one  0.142265  0.743638
b two
  Key1 Key2     data1     data2
3    b  two  1.006378 -0.245366
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:19:05 PM.

FINISHED

```
%pyspark
pieces = dict(list(df.groupby('Key1')))
```

Took 1 sec. Last updated by anonymous at March 09 2017, 11:19:09 PM.

FINISHED

```
%pyspark
pieces['b']
```

```
  Key1 Key2     data1     data2
2    b  one  0.142265  0.743638
3    b  two  1.006378 -0.245366
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:19:11 PM.

FINISHED

```
%pyspark
df.dtypes
```

```
Key1      object
Key2      object
data1    float64
data2    float64
dtype: object
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:19:14 PM.

FINISHED

```
%pyspark
grouped = df.groupby(df.dtypes,axis=1)
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:19:19 PM.

FINISHED

```
%pyspark
dict(list(grouped))
```

```
{dtype('float64'):       data1     data2
0 -0.432396 -0.350729
1 -0.334794  0.500221
2  0.142265  0.743638
3    1.000378  0.245366
     3295   0.271981, dtype('O'):   Key1 Key2
0     a  one
1     a  two
2     b  one
3     b  two
4     a  one}
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:19:21 PM.

FINISHED

```
%pyspark
df.groupby('key1')['data1']

df.groupby('key1')[['data2']]

df['data1'].groupby(df['key1'])

df[['data2']].groupby(df['key1'])

df.groupby(['key1', 'key2'])[['data2']].mean()

s_grouped = df.groupby(['key1', 'key2'])['data2']

s_grouped

s_grouped.mean()

people = DataFrame(np.random.randn(5, 5),
 columns=['a', 'b', 'c', 'd', 'e'],
 index=['Joe', 'Steve', 'Wes', 'Jim', 'Travis'])

people.ix[2:3, ['b', 'c']] = np.nan # Add a few NA values

people

mapping = {'a': 'red', 'b': 'red', 'c': 'blue',
 'd': 'blue', 'e': 'red', 'f' : 'orange'}

by_column = people.groupby(mapping, axis=1)

by_column.sum()

map_series = Series(mapping)

map_series

people.groupby(map_series, axis=1).count()

people.groupby(len).sum()

key_list = ['one', 'one', 'one', 'two', 'two']

people.groupby([len, key_list]).min()

columns = pd.MultiIndex.from_arrays([['US', 'US', 'US', 'JP', 'JP'],
 [1, 3, 5, 1, 3]], names=['cty', 'tenor'])

hier_df = DataFrame(np.random.randn(4, 5), columns=columns)

hier_df

hier_df.groupby(level='cty', axis=1).count()
```

Took 0 sec. Last updated by anonymous at March 09 2017, 11:29:53 PM. (outdated)

Zeppelin

FINISHED

Took 0 sec. Last updated by anonymous at March 09 2017, 11:19:41 PM.

AP_Class8_3.9.17

default