```
%pyspark
from pandas import Series, DataFrame
import numpy as np, pandas as pd

df = DataFrame({'key1' : ['a','a','b','b','a'],
                'key2' : ['one','two','one','two','one'],
                'data1' : np.random.randn(5),
                'data2' : np.random.randn(5)})


df
```

FINISHED

```
      data1     data2 key1 key2
0 -0.340748 -0.641836    a  one
1 -1.283440  0.145508    a  two
2  0.977394  1.717586    b  one
3  2.486059 -0.663007    b  two
4 -0.259208 -0.621510    a  one
```

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

grouped
```

FINISHED

```
<pandas.core.groupby.SeriesGroupBy object at 0x1130e7d10>
```

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

FINISHED

```
key1
a   -0.627799
b    1.731727
Name: data1, dtype: float64
```

```
%pyspark
means = df['data1'].groupby([df['key1'], df['key2']]).mean()

means
```

FINISHED

```
key1  key2
a     one    -0.299978
      two    -1.283440
b     one     0.977394
      two     2.486059
Name: data1, dtype: float64
```

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

FINISHED

```
key2       one       two
key1
a    -0.299978 -1.283440
b     0.977394  2.486059
```

```
%pyspark
states = np.array(['Ohio','California','California','Ohio','Ohio'])

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

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

FINISHED

```
California  2005   -1.283440
            2006    0.977394
Ohio        2005    1.072655
            2006   -0.259208
Name: data1, dtype: float64
```

```
%pyspark
df.groupby('key1').mean()
```

FINISHED

```
         data1     data2
key1
a    -0.627799 -0.372613
b     1.731727  0.527290
```

```
%pyspark
df.groupby(['key1','key2']).mean()
```

FINISHED

```
              data1     data2
key1 key2
a    one  -0.299978 -0.631673
     two  -1.283440  0.145508
b    one   0.977394  1.717586
     two   2.486059 -0.663007
```

```
%pyspark
df.groupby(['key1', 'key2']).size()
```

FINISHED

```
key1  key2
a     one     2
      two     1
b     one     1
      two     1
dtype: int64
```

```
%pyspark
for name, group in df.groupby('key1'):
    print name
    print group
```

FINISHED

```
a
      data1     data2 key1 key2
0 -0.340748 -0.641836    a  one
1 -1.283440  0.145508    a  two
4 -0.259208 -0.621510    a  one
b
      data1     data2 key1 key2
2  0.977394  1.717586    b  one
3  2.486059 -0.663007    b  two
```

```
%pyspark
for (k1, k2), group in df.groupby(['key1','key2']):
    print k1, k2
    print group
```

FINISHED

```
a one
      data1     data2 key1 key2
0 -0.340748 -0.641836    a  one
4 -0.259208 -0.621510    a  one
a two
     data1     data2 key1 key2
1 -1.28344  0.145508    a  two
b one
      data1     data2 key1 key2
2  0.977394  1.717586    b  one
b two
      data1     data2 key1 key2
3  2.486059 -0.663007    b  two
```

```
%pyspark
pieces = dict(list(df.groupby('key1')))

pieces['b']
```

FINISHED

```
      data1     data2 key1 key2
2  0.977394  1.717586    b  one
3  2.486059 -0.663007    b  two
```

```
%pyspark
df.d%pysparktypes
```

FINISHED

```
data1    float64
data2    float64
key1      object
key2      object
dtype: object
```

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

FINISHED

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

FINISHED

```
{dtype('O'):   key1 key2
0    a  one
1    a  two
2    b  one
3    b  two
4    a  one, dtype('float64'):       data1     data2
0 -0.340748 -0.641836
1 -1.283440  0.145508
2  0.977394  1.717586
3  2.486059 -0.663007
4 -0.259208 -0.621510}
```

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

FINISHED

```
<pandas.core.groupby.SeriesGroupBy object at 0x1132204d0>
```

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

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

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

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

FINISHED

```
<pandas.core.groupby.DataFrameGroupBy object at 0x113220ad0>
```

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

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

s_grouped

s_grouped.mean()
```

FINISHED

```
key1  key2
a     one    -0.631673
      two     0.145508
b     one     1.717586
      two    -0.663007
Name: data2, dtype: float64
```

FINISHED

```
%pyspark
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
```

```
              a         b         c         d         e
Joe     1.047168  1.332765 -0.734140 -1.866998 -0.162042
Steve   0.709255  1.942420  0.060512  2.230149 -0.281080
Wes     1.288871       NaN       NaN -0.715175  1.506722
Jim     0.312421 -0.330915 -0.886431 -1.028511 -1.076048
Travis -0.052692 -1.416666  0.163449  0.017301 -0.141440
```

FINISHED

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

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

```
            blue       red
Joe    -2.601138  2.217891
Steve   2.290661  2.370595
Wes    -0.715175  2.795593
Jim    -1.914942 -1.094542
Travis  0.180750 -1.610798
```

FINISHED

```
%pyspark
map_series = Series(mapping)

map_series
```

```
a       red
b       red
c      blue
d      blue
e       red
f    orange
dtype: object
```

FINISHED

```
%pyspark
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'])
```

FINISHED

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

hier_df

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

```
cty  JP  US
0     2   3
1     2   3
2     2   3
3     2   3
```

READY