# Data Aggregation

FINISHED

```
%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
```

```
      data1     data2 key1 key2
0  0.830010  0.098991    a  one
1  0.655552 -0.140033    a  two
2  1.805540 -1.066978    b  one
3 -0.733145  0.282225    b  two
4  0.389095  0.305841    a  one
```

FINISHED

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

```
key1
a    0.624885
b    0.536198
Name: data1, dtype: float64
```

FINISHED

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

```
key1  key2
a     one     0.609552
      two     0.655552
b     one     1.805540
      two    -0.733145
Name: data1, dtype: float64
```

FINISHED

```
%pyspark
```

```
means.unstack()
```

```
key2       one       two
key1
a     0.609552  0.655552
b     1.805540 -0.733145
```

```
%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    0.655552
            2006    1.805540
Ohio        2005    0.048432
            2006    0.389095
Name: data1, dtype: float64
```

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

FINISHED

```
         data1     data2
key1
a     0.624885  0.088267
b     0.536198 -0.392376
```

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

FINISHED

```
              data1     data2
key1 key2
a    one   0.609552  0.202416
     two   0.655552 -0.140033
b    one   1.805540 -1.066978
     two  -0.733145  0.282225
```

```
%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  0.830010  0.098991    a  one
1  0.655552 -0.140033    a  two
4  0.389095  0.305841    a  one
b
      data1     data2 key1 key2
2  1.805540 -1.066978    b  one
3 -0.733145  0.282225    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  0.830010  0.098991    a  one
4  0.389095  0.305841    a  one
a two
      data1     data2 key1 key2
1  0.655552 -0.140033    a  two
b one
     data1     data2 key1 key2
2  1.80554 -1.066978    b  one
b two
      data1     data2 key1 key2
3 -0.733145  0.282225    b  two
```

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

FINISHED

```
pieces['b']
```

```
      data1     data2 key1 key2
2  1.805540 -1.066978    b  one
3 -0.733145  0.282225    b  two
```

```
%pyspark
df.dtypes
```

FINISHED

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

```
%pyspark
grouped = df.groupby(df.dtypes, axis =1)
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  0.830010  0.098991
1  0.655552 -0.140033
2  1.805540 -1.066978
3 -0.733145  0.282225
4  0.389095  0.305841}
```

```
%pyspark

df.groupby('key1')['data1']
df.groupby('key1')[['data2']]
df[['data2']].groupby(df['key1'])
df.groupby(['key1', 'key2'])[['data2']].mean()
```

FINISHED

```
              data2
key1 key2
a    one   0.202416
     two  -0.140033
b    one  -1.066978
     two   0.282225
```

```
%pyspark

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

s_grouped

s_grouped.mean()
```

FINISHED

```
key1  key2
a     one     0.202416
      two    -0.140033
b     one    -1.066978
      two     0.282225
Name: data2, dtype: float64
```

```
%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

people
```

FINISHED

```
               a         b         c         d         e
Joe     0.244025 -0.296461 -0.476127  0.917887 -0.131135
Steve  -0.254700  0.383853  0.569686 -0.017164  1.203755
Wes     1.057538       NaN       NaN  1.204026  1.543216
Jim    -0.808178  0.522520  1.722389  0.286746 -0.082646
Travis -1.030573 -0.967670 -0.508838  2.233119  1.346585
```

```
%pyspark
```

FINISHED

```
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     0.441759 -0.183570
Steve   0.552522  1.332909
Wes     1.204026  2.600754
Jim     2.009136 -0.368305
Travis  1.724281 -0.651658
```

```
%pyspark
map_series = Series(mapping)

map_series
```

FINISHED

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

```
%pyspark
people.groupby(map_series, axis=1).count()

people.groupby(len).sum()
```

FINISHED

```
          a         b         c         d         e
3  0.493385  0.226059  1.246262  2.408658  1.329435
5 -0.254700  0.383853  0.569686 -0.017164  1.203755
6 -1.030573 -0.967670 -0.508838  2.233119  1.346585
```

```
%pyspark
key_list = ['one', 'one', 'one', 'two', 'two']
```

FINISHED

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

```
              a         b         c         d         e
3 one  0.244025 -0.296461 -0.476127  0.917887 -0.131135
  two -0.808178  0.522520  1.722389  0.286746 -0.082646
5 one -0.254700  0.383853  0.569686 -0.017164  1.203755
6 two -1.030573 -0.967670 -0.508838  2.233119  1.346585
```

```
%pyspark
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
```

FINISHED

```
cty          US                            JP
tenor         1         3         5         1         3
0     -0.579134  0.876165  1.918637  0.766555  0.043658
1     -2.175186  1.331887 -1.763658  0.529897 -0.462561
2      0.582438 -0.574404  0.416526 -0.616175  0.287130
3     -2.366853 -0.011979 -0.793241 -1.646191 -0.374064
```

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

FINISHED

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

READY