Zeppelin

FINISHED

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

import numpy as np
```

FINISHED

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

import numpy as np



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

FINISHED

```
%pyspark

df
```

```
      data1     data2 key1 key2
0 -0.964873 -0.699508    a  one
1 -0.644564 -0.135251    a  two
2 -0.249916 -1.164450    b  one
3 -0.957116 -0.346726    b  two
4 -0.732494  0.359041    a  one
```

FINISHED

```
%pyspark

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

FINISHED

```
%pyspark

grouped
```

```
<pandas.core.groupby.SeriesGroupBy object at 0x10b822390>
```

FINISHED

```
%pyspark

grouped.mean()
```

```
key1
a   -0.780643
b   -0.603516
Name: data1, dtype: float64
```

FINISHED

```
%pyspark

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

FINISHED

```
%pyspark

means
```

```
key1  key2
a     one    -0.848683
      two    -0.644564
b     one    -0.249916
      two    -0.957116
Name: data1, dtype: float64
```

FINISHED

```
%pyspark

means.unstack()
```

```
key2       one       two
key1
a    -0.848683 -0.644564
b    -0.249916 -0.957116
```

FINISHED

```
%pyspark

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

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

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

```
California  2005   -0.644564
            2006   -0.249916
Ohio        2005   -0.960994
            2006   -0.732494
Name: data1, dtype: float64
```

FINISHED

```
%pyspark

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

```
         data1     data2
key1
a    -0.780643 -0.158572
b    -0.603516 -0.755588
```

FINISHED

```
%pyspark

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

```
              data1     data2
key1 key2
a    one  -0.848683 -0.170233
     two  -0.644564 -0.135251
b    one  -0.249916 -1.164450
     two  -0.957116 -0.346726
```

FINISHED

```
%pyspark

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

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

FINISHED

```
%pyspark

for name, group in df.groupby('key1'):

   print name
   print group
```

```
a
      data1     data2 key1 key2
0 -0.964873 -0.699508    a  one
1 -0.644564 -0.135251    a  two
4 -0.732494  0.359041    a  one
b
      data1     data2 key1 key2
2 -0.249916 -1.164450    b  one
3 -0.957116 -0.346726    b  two
```

FINISHED

```
%pyspark

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

    print k1, k2
    print group
```

```
a one
      data1     data2 key1 key2
0 -0.964873 -0.699508    a  one
4 -0.732494  0.359041    a  one
a two
      data1     data2 key1 key2
1 -0.644564 -0.135251    a  two
b one
      data1    data2 key1 key2
2 -0.249916 -1.16445    b  one
b two
      data1     data2 key1 key2
3 -0.957116 -0.346726    b  two
```

FINISHED

```
%pyspark

pieces = dict(list(df.groupby('key1')))

pieces['b']
```

```
      data1     data2 key1 key2
2 -0.249916 -1.164450    b  one
3 -0.957116 -0.346726    b  two
```

FINISHED

```
%pyspark

df.dtypes
```

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

FINISHED

```
%pyspark

grouped = df.groupby(df.dtypes, axis=1)

dict(list(grouped))
```

```
{dtype('O'):   key1 key2
0    a  one
1    a  two
2    b  one
3    b  two
4    a  one, dtype('float64'):       data1     data2
0 -0.964873 -0.699508
1 -0.644564 -0.135251
2 -0.249916 -1.164450
3 -0.957116 -0.346726
4 -0.732494  0.359041}
```

READY