Zeppelin Notebook

2017-03-09 Aggreg...

Search your Notebooks

anonymous

default

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	-0.340748	-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

FINISHED

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

grouped
```

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

FINISHED

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

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

FINISHED

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

means
```

```
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}
```

READY