ZA - March 23, 1017..

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

FINISHED

Took 34 sec. Last updated by anonymous at March 23 2017, 6:20:05 PM.

```
%pyspark

people = DataFrame(np.random.randn(5,5), columns=['a','b','c','d','e'], index=['Joe','Steve',
people.ix[2:3, ['b','c']] = np.nan
people
```

FINISHED

```
               a         b         c         d         e
Joe     2.059522  0.449785  0.312271 -0.751046  1.007976
Steve  -0.716739  0.971727  0.556467 -1.549082 -0.817007
Wes    -0.373373       NaN       NaN  1.350324 -0.749256
Jim     0.351151 -0.530637  0.946551  0.564245 -0.909751
Travis  1.243299 -0.673996  2.029088 -1.996491 -1.731243
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:20:52 PM.

```
%pyspark

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()
```

FINISHED

```
        blue  red
Joe        2    3
Steve      2    3
Wes        1    2
Jim        2    3
Travis     2    3
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:21:58 PM.

```
%pyspark

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

FINISHED

```
key_list = ['one', 'one', 'one', 'two', 'two']
people.groupby([len, key_list]).min()
```

```
              a         b         c         d         e
3 one -0.373373  0.449785  0.312271 -0.751046 -0.749256
  two  0.351151 -0.530637  0.946551  0.564245 -0.909751
5 one -0.716739  0.971727  0.556467 -1.549082 -0.817007
6 two  1.243299 -0.673996  2.029088 -1.996491 -1.731243
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:23:23 PM.

```
%pyspark

columns = pd.MultiIndex.from_arrays([['US', 'US', 'US', 'JP', 'JP'], [1, 3, 5, 1, 3]], names=
hier_df = DataFrame(np.random.randn(4, 5), columns=columns)
hier_df
```

FINISHED

```
cty          US                            JP          
tenor         1         3         5         1         3
0      0.743275  0.125953  0.641539  0.843928  1.061994
1     -0.658686 -0.618527  0.080921  0.260049  1.305878
2     -0.410266  0.456593  1.265906  1.101358  0.801231
3      0.492608  0.206103  1.074129 -0.181982 -1.107113
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:24:54 PM.

```
%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
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:25:09 PM.

```
%pyspark

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

Took 0 sec. Last updated by anonymous at March 23 2017, 6:28:06 PM.

```
%pyspark

grouped = df.groupby('key1')
grouped['data1'].quantile(0.9)
```

FINISHED

```
key1
a   -0.735039
b    0.030988
Name: data1, dtype: float64
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:28:38 PM.

```
%pyspark

def peak_to_peak(arr): return arr.max() - arr.min()
grouped.agg(peak_to_peak)
```

FINISHED

```
         data1     data2
key1
a     0.946903  0.936521
b     0.037483  0.319820
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:42:03 PM.

```
%pyspark

grouped.describe()
```

FINISHED

```
               data1     data2
key1
a    count  3.000000  3.000000
     mean  -1.122306 -0.140461
     std    0.474882  0.468603
     min   -1.574492 -0.598382
     25%   -1.369665 -0.379762
     50%   -1.164838 -0.161141
     75%   -0.896213  0.088499
     max   -0.627589  0.338139
b    count  2.000000  2.000000
     mean   0.015995  0.227089
     std    0.026504  0.226147
     min   -0.002747  0.067179
     25%    0.006624  0.147134
     50%    0.015995  0.227089
     75%    0.025365  0.307044
     max    0.034736  0.386999
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:42:19 PM.

```
%pyspark

tips = pd.read_csv('/Users/datascienceadmin/Downloads/tips.csv')
```

FINISHED

Took 1 sec. Last updated by anonymous at March 23 2017, 6:40:19 PM.

```
%pyspark

tips['tip_pct'] = tips['tip'] / tips['total_bill']
tips[:6]
```

FINISHED

```
   total_bill   tip     sex smoker  day    time  size   tip_pct
0       16.99  1.01  Female     No  Sun  Dinner     2  0.059447
1       10.34  1.66    Male     No  Sun  Dinner     3  0.160542
2       21.01  3.50    Male     No  Sun  Dinner     3  0.166587
3       23.68  3.31    Male     No  Sun  Dinner     2  0.139780
4       24.59  3.61  Female     No  Sun  Dinner     4  0.146808
5       25.29  4.71    Male     No  Sun  Dinner     4  0.186240
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:42:40 PM.

```
%pyspark

grouped = tips.groupby(['sex','smoker'])
grouped_pct = grouped['tip_pct']
grouped_pct.agg('mean')
```

FINISHED

```
sex     smoker
Female  No        0.156921
        Yes       0.182150
Male    No        0.160669
        Yes       0.152771
Name: tip_pct, dtype: float64
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:43:03 PM.

```
%pyspark

grouped_pct.agg(['mean','std',peak_to_peak])
```

FINISHED

```
                   mean       std  peak_to_peak
sex    smoker
Female No      0.156921  0.036421      0.195876
       Yes     0.182150  0.071595      0.360233
Male   No      0.160669  0.041849      0.220186
       Yes     0.152771  0.090588      0.674707
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:43:51 PM.

```
%pyspark

grouped_pct.agg([('foo','mean'),('bar',np.std)])
```

FINISHED

```
                    foo       bar
sex    smoker
Female No      0.156921  0.036421
       Yes     0.182150  0.071595
Male   No      0.160669  0.041849
       Yes     0.152771  0.090588
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:45:44 PM.

```
%pyspark

functions = ['count','mean','max']
result = grouped['tip_pct','total_bill'].agg(functions)
result
```

FINISHED

```
              tip_pct                     total_bill
                count      mean       max      count       mean    max
sex    smoker
Female No          54  0.156921  0.252672         54  18.105185  35.83
       Yes         33  0.182150  0.416667         33  17.977879  44.30
Male   No          97  0.160669  0.291990         97  19.791237  48.33
       Yes         60  0.152771  0.710345         60  22.284500  50.81
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:45:59 PM.

```
%pyspark

result['tip_pct']
```

FINISHED

```
               count      mean       max
sex    smoker
Female No         54  0.156921  0.252672
       Yes        33  0.182150  0.416667
Male   No         97  0.160669  0.291990
       Yes        60  0.152771  0.710345
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:46:14 PM.

```
%pyspark

ftuples = [('Durchschnitt', 'mean'), ('Abweichung', np.var)]
grouped['tip_pct','total_bill'].agg(ftuples)
```

FINISHED

```
                  tip_pct              total_bill
              Durchschnitt Abweichung Durchschnitt Abweichung
sex    smoker
Female No         0.156921   0.001327    18.105185  53.092422
       Yes        0.182150   0.005126    17.977879  84.451517
Male   No         0.160669   0.001751    19.791237  76.152961
       Yes        0.152771   0.008206    22.284500  98.244673
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:46:30 PM.

```
%pyspark

grouped.agg({'tip' : np.max, 'size' : 'sum'})
```

FINISHED

```
               size   tip
sex    smoker
Female No       140   5.2
       Yes       74   6.5
Male   No       263   9.0
       Yes      150  10.0
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:48:41 PM.

```
%pyspark

grouped.agg({'tip_pct' : ['min', 'max', 'mean', 'std'], 'size' : 'sum'})
```

FINISHED

```
                tip_pct                                  size
                    min       max      mean       std   sum
sex    smoker
Female No      0.056797  0.252672  0.156921  0.036421   140
       Yes     0.056433  0.416667  0.182150  0.071595    74
Male   No      0.071804  0.291990  0.160669  0.041849   263
       Yes     0.035638  0.710345  0.152771  0.090588   150
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:48:58 PM.

```
%pyspark

tips.groupby(['sex','smoker'], as_index=False).mean()
```

FINISHED

```
     sex smoker  total_bill       tip      size   tip_pct
0  Female     No   18.105185  2.773519  2.592593  0.156921
1  Female    Yes   17.977879  2.931515  2.242424  0.182150
2    Male     No   19.791237  3.113402  2.711340  0.160669
3    Male    Yes   22.284500  3.051167  2.500000  0.152771
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:49:16 PM.

```
%pyspark

df
```

FINISHED

```
      data1     data2 key1 key2
0 -0.627589 -0.161141    a  one
1 -1.164838 -0.598382    a  two
2  0.034736  0.067179    b  one
3 -0.002747  0.386999    b  two
4 -1.574492  0.338139    a  one
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:49:34 PM.

```
%pyspark

k1_means = df.groupby('key1').mean().add_prefix('mean_')
k1_means
```

FINISHED

```
      mean_data1  mean_data2
key1
a      -1.122306   -0.140461
b       0.015995    0.227089
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:49:53 PM.

```
%pyspark

pd.merge(df, k1_means, left_on='key1', right_index=True)
```

FINISHED

```
      data1     data2 key1 key2  mean_data1  mean_data2
0 -0.627589 -0.161141    a  one   -1.122306   -0.140461
1 -1.164838 -0.598382    a  two   -1.122306   -0.140461
4 -1.574492  0.338139    a  one   -1.122306   -0.140461
2  0.034736  0.067179    b  one    0.015995    0.227089
3 -0.002747  0.386999    b  two    0.015995    0.227089
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:50:06 PM.

```
%pyspark

key = ['one', 'two', 'one', 'two', 'one']
people.groupby(key).mean()
```

FINISHED

```
            a         b         c         d         e
one  0.976482 -0.112106  1.170680 -0.465738 -0.490841
two -0.182794  0.220545  0.751509 -0.492419 -0.863379
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:50:18 PM.

```
%pyspark

people.groupby(key).transform(np.mean)
```

FINISHED

```
               a         b         c         d         e
Joe     0.976482 -0.112106  1.170680 -0.465738 -0.490841
Steve  -0.182794  0.220545  0.751509 -0.492419 -0.863379
Wes     0.976482 -0.112106  1.170680 -0.465738 -0.490841
Jim    -0.182794  0.220545  0.751509 -0.492419 -0.863379
Travis  0.976482 -0.112106  1.170680 -0.465738 -0.490841
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:50:34 PM.

```
%pyspark

def demean(arr): return arr - arr.mean()

demeaned = people.groupby(key).transform(demean)
demeaned
```

FINISHED

```
               a         b         c         d         e
Joe     1.083039  0.561891 -0.858408 -0.285308  1.498817
Steve  -0.533945  0.751182 -0.195042 -1.056664  0.046372
Wes    -1.349856       NaN       NaN  1.816062 -0.258415
Jim     0.533945 -0.751182  0.195042  1.056664 -0.046372
Travis  0.266817 -0.561891  0.858408 -1.530753 -1.240402
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:51:03 PM.

```
%pyspark

demeaned.groupby(key).mean()
```

FINISHED

```
                a    b             c    d             e
one -7.401487e-17  0.0 -1.110223e-16  0.0  0.000000e+00
two  0.000000e+00  0.0  0.000000e+00  0.0 -5.551115e-17
```

Took 1 sec. Last updated by anonymous at March 23 2017, 6:51:18 PM.

```
%pyspark

def top(df, n=5, column='tip_pct'): return df.sort_index(by=column)[-n:]

top(tips, n=6)
```

FINISHED

```
/var/folders/g3/8csy0jq52kdf7dwf0g391dk40000gn/T/zeppelin_pyspark-5113699294139885083.py:1: Fu
tureWarning: by argument to sort_index is deprecated, pls use .sort_values(by=...)
  #
     total_bill   tip     sex smoker  day    time  size   tip_pct
109       14.31  4.00  Female    Yes  Sat  Dinner     2  0.279525
183       23.17  6.50    Male    Yes  Sun  Dinner     4  0.280535
232       11.61  3.39    Male     No  Sat  Dinner     2  0.291990
67         3.07  1.00  Female    Yes  Sat  Dinner     1  0.325733
178        9.60  4.00  Female    Yes  Sun  Dinner     2  0.416667
172        7.25  5.15    Male    Yes  Sun  Dinner     2  0.710345
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:51:35 PM.

```
%pyspark

tips.groupby('smoker').apply(top)
```

FINISHED

```
            total_bill   tip     sex smoker   day    time  size   tip_pct
smoker
No     88        24.71  5.85    Male     No  Thur   Lunch     2  0.236746
       185       20.69  5.00    Male     No   Sun  Dinner     5  0.241663
       51        10.29  2.60  Female     No   Sun  Dinner     2  0.252672
       149        7.51  2.00    Male     No  Thur   Lunch     2  0.266312
       232       11.61  3.39    Male     No   Sat  Dinner     2  0.291990
Yes    109       14.31  4.00  Female    Yes   Sat  Dinner     2  0.279525
       183       23.17  6.50    Male    Yes   Sun  Dinner     4  0.280535
       67         3.07  1.00  Female    Yes   Sat  Dinner     1  0.325733
       178        9.60  4.00  Female    Yes   Sun  Dinner     2  0.416667
       172        7.25  5.15    Male    Yes   Sun  Dinner     2  0.710345
```

Took 1 sec. Last updated by anonymous at March 23 2017, 6:51:49 PM.

```
%pyspark

tips.groupby(['smoker','day']).apply(top, n=1, column='total_bill')
```

FINISHED

```
                 total_bill    tip     sex smoker   day    time  size  \
smoker day
No     Fri  94        22.75   3.25  Female     No   Fri  Dinner     2
       Sat  212       48.33   9.00    Male     No   Sat  Dinner     4
       Sun  156       48.17   5.00    Male     No   Sun  Dinner     6
       Thur 142       41.19   5.00    Male     No  Thur   Lunch     5
Yes    Fri  95        40.17   4.73    Male    Yes   Fri  Dinner     4
       Sat  170       50.81  10.00    Male    Yes   Sat  Dinner     3
       Sun  182       45.35   3.50    Male    Yes   Sun  Dinner     3
       Thur 197       43.11   5.00  Female    Yes  Thur   Lunch     4
                  tip_pct
smoker day
No     Fri  94   0.142857
       Sat  212  0.186220
       Sun  156  0.103799
       Thur 142  0.121389
Yes    Fri  95   0.117750
       Sat  170  0.196812
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:52:02 PM.

```
%pyspark

result = tips.groupby('smoker')['tip_pct'].describe()
result
```

FINISHED

```
smoker
No      count    151.000000
        mean       0.159328
        std        0.039910
        min        0.056797
        25%        0.136906
        50%        0.155625
        75%        0.185014
        max        0.291990
Yes     count     93.000000
        mean       0.163196
        std        0.085119
        min        0.035638
        25%        0.106771
        50%        0.153846
        75%        0.195059
        max        0.710345
Name: tip_pct, dtype: float64
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:52:20 PM.

```
%pyspark

result.unstack('smoker')
```

FINISHED

```
smoker           No        Yes
count   151.000000  93.000000
mean      0.159328   0.163196
std       0.039910   0.085119
min       0.056797   0.035638
25%       0.136906   0.106771
50%       0.155625   0.153846
75%       0.185014   0.195059
max       0.291990   0.710345
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:52:36 PM.

```
%pyspark

f = lambda x: x.describe()
grouped.apply(f)
```

FINISHED

```
                     total_bill        tip       size    tip_pct
sex    smoker
Female No     count   54.000000  54.000000  54.000000  54.000000
              mean    18.105185   2.773519   2.592593   0.156921
              std      7.286455   1.128425   1.073146   0.036421
              min      7.250000   1.000000   1.000000   0.056797
              25%     12.650000   2.000000   2.000000   0.139708
              50%     16.690000   2.680000   2.000000   0.149691
              75%     20.862500   3.437500   3.000000   0.181630
              max     35.830000   5.200000   6.000000   0.252672
       Yes    count   33.000000  33.000000  33.000000  33.000000
              mean    17.977879   2.931515   2.242424   0.182150
              std      9.189751   1.219916   0.613917   0.071595
              min      3.070000   1.000000   1.000000   0.056433
              25%     12.760000   2.000000   2.000000   0.152439
              50%     16.270000   2.880000   2.000000   0.173913
              75%     22.120000   3.500000   2.000000   0.198216
              max     44.300000   6.500000   4.000000   0.416667
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:52:47 PM.

```
%pyspark

tips.groupby('smoker', group_keys=False).apply(top)
```

FINISHED

```
     total_bill   tip     sex smoker   day    time  size   tip_pct
88        24.71  5.85    Male     No  Thur   Lunch     2  0.236746
185       20.69  5.00    Male     No   Sun  Dinner     5  0.241663
51        10.29  2.60  Female     No   Sun  Dinner     2  0.252672
149        7.51  2.00    Male     No  Thur   Lunch     2  0.266312
232       11.61  3.39    Male     No   Sat  Dinner     2  0.291990
109       14.31  4.00  Female    Yes   Sat  Dinner     2  0.279525
183       23.17  6.50    Male    Yes   Sun  Dinner     4  0.280535
67         3.07  1.00  Female    Yes   Sat  Dinner     1  0.325733
178        9.60  4.00  Female    Yes   Sun  Dinner     2  0.416667
172        7.25  5.15    Male    Yes   Sun  Dinner     2  0.710345
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:52:59 PM.

```
%pyspark                                                                          FINISHED

frame = DataFrame({'data1': np.random.randn(1000), 'data2': np.random.randn(1000)})
factor = pd.cut(frame.data1, 4)
```

```
0    (-1.477, 0.149]
1    (-1.477, 0.149]
2    (-3.11, -1.477]
3    (-3.11, -1.477]
4     (0.149, 1.776]
5    (-1.477, 0.149]
6    (-1.477, 0.149]
7    (-1.477, 0.149]
8     (0.149, 1.776]
9    (-1.477, 0.149]
Name: data1, dtype: category
Categories (4, object): [(-3.11, -1.477] < (-1.477, 0.149] < (0.149, 1.776] < (1.776, 3.402]]
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:53:20 PM.

```
%pyspark                                                                          FINISHED

def get_stats(group): return {'min': group.min(), 'max': group.max(), 'count': group.count(),

grouped = frame.data2.groupby(factor)
grouped.apply(get_stats).unstack()
```

```
                 count       max      mean       min
data1
(-3.11, -1.477]   77.0  2.693274  0.031309 -1.757657
(-1.477, 0.149]  501.0  3.163908 -0.011432 -2.872045
(0.149, 1.776]   379.0  2.612984  0.028904 -2.657110
(1.776, 3.402]    43.0  2.037545 -0.153709 -2.417015
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:53:49 PM.

```
%pyspark                                                                          FINISHED

grouping = pd.qcut(frame.data1, 10, labels=False)

grouped = frame.data2.groupby(grouping)
grouped.apply(get_stats).unstack()
```

```
       count       max      mean       min
data1
0      100.0  2.693274 -0.045058 -1.962398
1      100.0  3.163908  0.069553 -2.088065
2      100.0  2.093690 -0.131870 -2.872045
3      100.0  2.265293 -0.034330 -2.470415
4      100.0  1.867187  0.039112 -2.640028
5      100.0  2.376244  0.041758 -2.779234
6      100.0  2.598747 -0.053582 -2.226666
7      100.0  2.586880  0.085379 -2.466933
8      100.0  2.612984  0.052436 -2.657110
9      100.0  2.118315 -0.013115 -2.417015
```

Took 0 sec. Last updated by anonymous at March 23 2017, 6:54:08 PM.

```
%pyspark
```

READY