ZA - March 30, 2017

```
%pyspark
```

FINISHED

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

CCDS = ChicagoCrimes.csv ('Users/DataScienceAdmin/Downloads/')
```

Took 0 sec. Last updated by anonymous at March 30 2017, 6:53:05 PM.

```
%pyspark
```

FINISHED

```
frame = DataFrame({'data1': np.random.randn(1000), 'data2': np.random.randn(1000)})
factor = pd.cut(frame.data1,4)
factor[:10]
```

```
0    (-1.405, 0.376]
1    (-1.405, 0.376]
2     (0.376, 2.157]
3    (-1.405, 0.376]
4    (-1.405, 0.376]
5     (0.376, 2.157]
6     (0.376, 2.157]
7    (-1.405, 0.376]
8     (0.376, 2.157]
9     (0.376, 2.157]
Name: data1, dtype: category
Categories (4, object): [(-3.193, -1.405] < (-1.405, 0.376] < (0.376, 2.157] < (2.157, 3.938]]
```

Took 0 sec. Last updated by anonymous at March 30 2017, 6:53:13 PM.

```
%pyspark
```

FINISHED

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

Took 0 sec. Last updated by anonymous at March 30 2017, 6:43:19 PM.

```
%pyspark
```

FINISHED

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

```
                    count       max      mean       min
data1
(-3.251, -1.624]     59.0  2.583277  0.172301 -1.836291
(-1.624, -0.00248]  425.0  2.812056 -0.093097 -3.255579
(-0.00248, 1.619]   451.0  3.156449  0.050622 -3.342087
(1.619, 3.24]        65.0  2.409241 -0.031057 -2.136424
```

Took 0 sec. Last updated by anonymous at March 30 2017, 6:44:02 PM.

```
%pyspark

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

FINISHED

Took 0 sec. Last updated by anonymous at March 30 2017, 6:44:30 PM.

```
%pyspark

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

FINISHED

```
       count       max      mean       min
data1
0      100.0  2.583277  0.094771 -2.059120
1      100.0  2.391859 -0.157776 -2.925234
2      100.0  2.764277 -0.103743 -2.528881
3      100.0  2.363338 -0.154678 -2.599154
4      100.0  2.812056 -0.020064 -3.255579
5      100.0  2.869866 -0.134605 -3.342087
6      100.0  2.591914  0.117592 -2.105518
7      100.0  2.372226  0.092537 -3.189796
8      100.0  3.156449  0.015267 -2.223778
9      100.0  2.861311  0.164812 -2.189721
```

Took 0 sec. Last updated by anonymous at March 30 2017, 6:44:36 PM.

```
%pyspark

s = Series(np.random.randn(6))
s[::2] = np.nan
s
```

FINISHED

```
0         NaN
1   -2.130022
2         NaN
3    0.015828
4         NaN
5   -0.711942
dtype: float64
```

Took 0 sec. Last updated by anonymous at March 30 2017, 7:04:55 PM.

```
%pyspark
```

READY