

## 4.1 - Groupby

April 11, 2017

```
In [1]: import pandas as pd
```

```
In [2]: df = pd.read_csv('quotes2.csv', parse_dates=['time'])
```

```
In [3]: def mia_f(x):  
        return x[0]
```

```
df.apply(mia_f, axis=1)
```

```
0    2016-05-25 13:30:00.023  
1    2016-05-25 13:30:00.023  
2    2016-05-25 13:30:00.041  
3    2016-05-25 13:30:00.048  
4    2016-05-25 13:30:00.048  
5    2016-05-25 13:30:00.048  
6    2016-05-25 13:30:00.048  
7    2016-05-25 13:30:00.072  
8    2016-05-25 13:30:00.075  
9    2016-05-25 13:30:00.076  
10   2016-05-25 13:30:00.076  
11   2016-05-25 13:30:00.076  
12   2016-05-25 13:30:00.078  
13   2016-05-25 13:30:00.078  
14   2016-05-25 13:30:00.078  
15   2016-05-25 13:30:00.078  
16   2016-05-25 13:30:00.079  
17   2016-05-25 13:30:00.080  
18   2016-05-25 13:30:00.084  
19   2016-05-25 13:30:00.086  
20   2016-05-25 13:30:00.088  
21   2016-05-25 13:30:00.089  
22   2016-05-25 13:30:00.104  
23   2016-05-25 13:30:00.104  
24   2016-05-25 13:30:00.104  
25   2016-05-25 13:30:00.104  
26   2016-05-25 13:30:00.104
```

```
27 2016-05-25 13:30:00.105
28 2016-05-25 13:30:00.107
29 2016-05-25 13:30:00.115
30 2016-05-25 13:30:00.115
31 2016-05-25 13:30:00.118
32 2016-05-25 13:30:00.128
33 2016-05-25 13:30:00.128
34 2016-05-25 13:30:00.129
35 2016-05-25 13:30:00.129
36 2016-05-25 13:30:00.129
37 2016-05-25 13:30:00.129
38 2016-05-25 13:30:00.129
39 2016-05-25 13:30:00.130
40 2016-05-25 13:30:00.130
41 2016-05-25 13:30:00.130
42 2016-05-25 13:30:00.131
43 2016-05-25 13:30:00.131
44 2016-05-25 13:30:00.135
45 2016-05-25 13:30:00.135
46 2016-05-25 13:30:00.136
47 2016-05-25 13:30:00.136
48 2016-05-25 13:30:00.144
49 2016-05-25 13:30:00.144
50 2016-05-25 13:30:00.145
51 2016-05-25 13:30:00.145
52 2016-05-25 13:30:00.145
53 2016-05-25 13:30:00.145
54 2016-05-25 13:30:00.145
55 2016-05-25 13:30:00.145
dtype: datetime64[ns]
```

## 0.1 Groupby

```
In [4]: import numpy as np
```

```
df.groupby(['ticker']).aggregate(np.mean)
```

	bid	ask
ticker		
AAPL	98.60525	98.617500
GOOG	720.50000	720.921667
MSFT	51.93700	51.950000

```
In [5]: df.groupby(['time']).aggregate(np.mean)
```

		bid	ask
time			
2016-05-25	13:30:00.023	386.225000	386.440000
2016-05-25	13:30:00.041	51.950000	51.950000
2016-05-25	13:30:00.048	720.500000	720.930000
2016-05-25	13:30:00.072	720.500000	720.880000
2016-05-25	13:30:00.075	98.550000	98.560000
2016-05-25	13:30:00.076	98.550000	98.560000
2016-05-25	13:30:00.078	51.942500	51.950000
2016-05-25	13:30:00.079	51.920000	51.950000
2016-05-25	13:30:00.080	98.550000	98.560000
2016-05-25	13:30:00.084	98.550000	98.560000
2016-05-25	13:30:00.086	98.550000	98.630000
2016-05-25	13:30:00.088	98.650000	98.630000
2016-05-25	13:30:00.089	98.630000	98.630000
2016-05-25	13:30:00.104	98.628000	98.630000
2016-05-25	13:30:00.105	98.620000	98.630000
2016-05-25	13:30:00.107	98.620000	98.630000
2016-05-25	13:30:00.115	98.620000	98.630000
2016-05-25	13:30:00.118	98.620000	98.630000
2016-05-25	13:30:00.128	98.620000	98.630000
2016-05-25	13:30:00.129	98.616000	98.630000
2016-05-25	13:30:00.130	67.490000	67.510000
2016-05-25	13:30:00.131	98.610000	98.620000
2016-05-25	13:30:00.135	75.265000	75.285000
2016-05-25	13:30:00.136	98.610000	98.620000
2016-05-25	13:30:00.144	98.610000	98.620000
2016-05-25	13:30:00.145	98.606667	98.628333

```
In [6]: df.groupby(['time']).mean()
```

		bid	ask
time			
2016-05-25	13:30:00.023	386.225000	386.440000
2016-05-25	13:30:00.041	51.950000	51.950000
2016-05-25	13:30:00.048	720.500000	720.930000
2016-05-25	13:30:00.072	720.500000	720.880000
2016-05-25	13:30:00.075	98.550000	98.560000
2016-05-25	13:30:00.076	98.550000	98.560000
2016-05-25	13:30:00.078	51.942500	51.950000
2016-05-25	13:30:00.079	51.920000	51.950000
2016-05-25	13:30:00.080	98.550000	98.560000
2016-05-25	13:30:00.084	98.550000	98.560000
2016-05-25	13:30:00.086	98.550000	98.630000
2016-05-25	13:30:00.088	98.650000	98.630000
2016-05-25	13:30:00.089	98.630000	98.630000

2016-05-25	13:30:00.104	98.628000	98.630000
2016-05-25	13:30:00.105	98.620000	98.630000
2016-05-25	13:30:00.107	98.620000	98.630000
2016-05-25	13:30:00.115	98.620000	98.630000
2016-05-25	13:30:00.118	98.620000	98.630000
2016-05-25	13:30:00.128	98.620000	98.630000
2016-05-25	13:30:00.129	98.616000	98.630000
2016-05-25	13:30:00.130	67.490000	67.510000
2016-05-25	13:30:00.131	98.610000	98.620000
2016-05-25	13:30:00.135	75.265000	75.285000
2016-05-25	13:30:00.136	98.610000	98.620000
2016-05-25	13:30:00.144	98.610000	98.620000
2016-05-25	13:30:00.145	98.606667	98.628333

## 0.2 Altri tipi di aggregazione

```
In [7]: mdf = df.groupby(['ticker', 'time']).mean()
```

```
In [8]: mdf.groupby('ticker').transform(lambda g : g - g.mean())
```

			bid	ask
ticker	time			
AAPL	2016-05-25	13:30:00.075	-0.051533	-5.391667e-02
	2016-05-25	13:30:00.076	-0.051533	-5.391667e-02
	2016-05-25	13:30:00.080	-0.051533	-5.391667e-02
	2016-05-25	13:30:00.084	-0.051533	-5.391667e-02
	2016-05-25	13:30:00.086	-0.051533	1.608333e-02
	2016-05-25	13:30:00.088	0.048467	1.608333e-02
	2016-05-25	13:30:00.089	0.028467	1.608333e-02
	2016-05-25	13:30:00.104	0.026467	1.608333e-02
	2016-05-25	13:30:00.105	0.018467	1.608333e-02
	2016-05-25	13:30:00.107	0.018467	1.608333e-02
	2016-05-25	13:30:00.115	0.018467	1.608333e-02
	2016-05-25	13:30:00.118	0.018467	1.608333e-02
	2016-05-25	13:30:00.128	0.018467	1.608333e-02
	2016-05-25	13:30:00.129	0.014467	1.608333e-02
	2016-05-25	13:30:00.130	0.008467	1.608333e-02
	2016-05-25	13:30:00.131	0.008467	6.083333e-03
	2016-05-25	13:30:00.135	0.008467	6.083333e-03
	2016-05-25	13:30:00.136	0.008467	6.083333e-03
	2016-05-25	13:30:00.144	0.008467	6.083333e-03
	2016-05-25	13:30:00.145	0.005133	1.441667e-02
GOOG	2016-05-25	13:30:00.023	0.000000	1.666667e-02
	2016-05-25	13:30:00.048	0.000000	1.666667e-02
	2016-05-25	13:30:00.072	0.000000	-3.333333e-02
MSFT	2016-05-25	13:30:00.023	0.014583	7.105427e-15

```

2016-05-25 13:30:00.041  0.014583  7.105427e-15
2016-05-25 13:30:00.078  0.007083  7.105427e-15
2016-05-25 13:30:00.079 -0.015417  7.105427e-15
2016-05-25 13:30:00.130 -0.005417  7.105427e-15
2016-05-25 13:30:00.135 -0.015417  7.105427e-15

```

```
In [9]: mdf.groupby(['ticker']).last()
```

```

          bid      ask
ticker
AAPL    98.606667  98.628333
GOOG   720.500000  720.880000
MSFT    51.920000  51.950000

```

```
In [10]: mdf.groupby(['time']).max()
```

```

          bid      ask
time
2016-05-25 13:30:00.023  720.500000  720.930000
2016-05-25 13:30:00.041    51.950000  51.950000
2016-05-25 13:30:00.048  720.500000  720.930000
2016-05-25 13:30:00.072  720.500000  720.880000
2016-05-25 13:30:00.075    98.550000  98.560000
2016-05-25 13:30:00.076    98.550000  98.560000
2016-05-25 13:30:00.078    51.942500  51.950000
2016-05-25 13:30:00.079    51.920000  51.950000
2016-05-25 13:30:00.080    98.550000  98.560000
2016-05-25 13:30:00.084    98.550000  98.560000
2016-05-25 13:30:00.086    98.550000  98.630000
2016-05-25 13:30:00.088    98.650000  98.630000
2016-05-25 13:30:00.089    98.630000  98.630000
2016-05-25 13:30:00.104    98.628000  98.630000
2016-05-25 13:30:00.105    98.620000  98.630000
2016-05-25 13:30:00.107    98.620000  98.630000
2016-05-25 13:30:00.115    98.620000  98.630000
2016-05-25 13:30:00.118    98.620000  98.630000
2016-05-25 13:30:00.128    98.620000  98.630000
2016-05-25 13:30:00.129    98.616000  98.630000
2016-05-25 13:30:00.130    98.610000  98.630000
2016-05-25 13:30:00.131    98.610000  98.620000
2016-05-25 13:30:00.135    98.610000  98.620000
2016-05-25 13:30:00.136    98.610000  98.620000
2016-05-25 13:30:00.144    98.610000  98.620000
2016-05-25 13:30:00.145    98.606667  98.628333

```

```
In [11]: # Identico
         mdf.groupby(['time']).apply(lambda g : g.max())
```

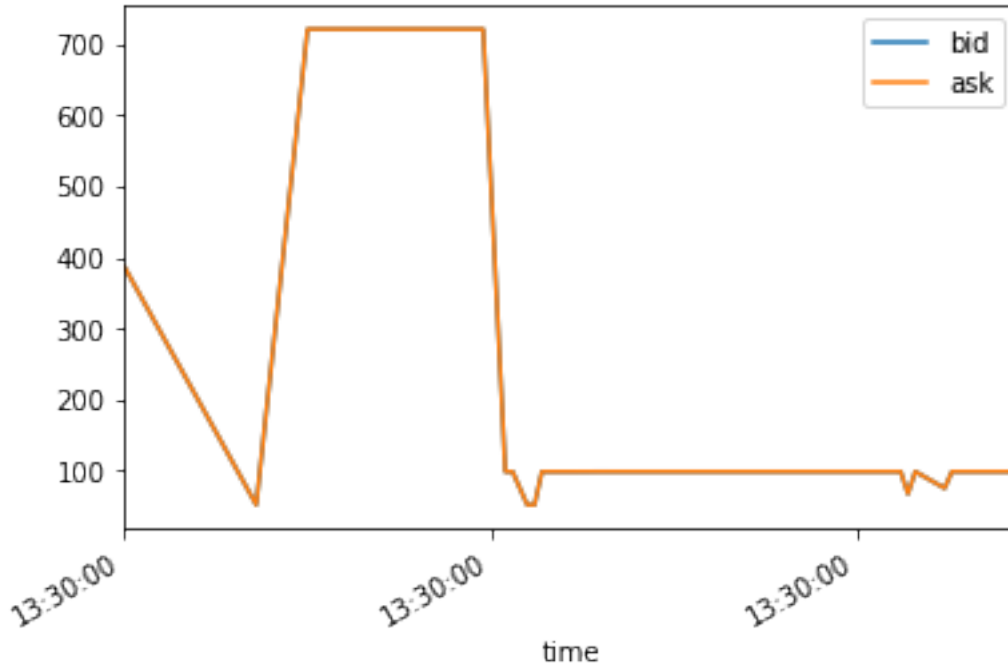
time		bid	ask
2016-05-25 13:30:00.023		720.500000	720.930000
2016-05-25 13:30:00.041		51.950000	51.950000
2016-05-25 13:30:00.048		720.500000	720.930000
2016-05-25 13:30:00.072		720.500000	720.880000
2016-05-25 13:30:00.075		98.550000	98.560000
2016-05-25 13:30:00.076		98.550000	98.560000
2016-05-25 13:30:00.078		51.942500	51.950000
2016-05-25 13:30:00.079		51.920000	51.950000
2016-05-25 13:30:00.080		98.550000	98.560000
2016-05-25 13:30:00.084		98.550000	98.560000
2016-05-25 13:30:00.086		98.550000	98.630000
2016-05-25 13:30:00.088		98.650000	98.630000
2016-05-25 13:30:00.089		98.630000	98.630000
2016-05-25 13:30:00.104		98.628000	98.630000
2016-05-25 13:30:00.105		98.620000	98.630000
2016-05-25 13:30:00.107		98.620000	98.630000
2016-05-25 13:30:00.115		98.620000	98.630000
2016-05-25 13:30:00.118		98.620000	98.630000
2016-05-25 13:30:00.128		98.620000	98.630000
2016-05-25 13:30:00.129		98.616000	98.630000
2016-05-25 13:30:00.130		98.610000	98.630000
2016-05-25 13:30:00.131		98.610000	98.620000
2016-05-25 13:30:00.135		98.610000	98.620000
2016-05-25 13:30:00.136		98.610000	98.620000
2016-05-25 13:30:00.144		98.610000	98.620000
2016-05-25 13:30:00.145		98.606667	98.628333

### 0.3 Plot

```
In [13]: %matplotlib inline
```

```
df.groupby(['time']).mean().plot()
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7fad83d7ab00>
```



```
In [14]: mdf = df.groupby(['ticker', 'time']).mean()
```

```
In [15]: gb = df.groupby(['ticker', 'time'])
```

```
In [16]: mdf
```

ticker	time	bid	ask
AAPL	2016-05-25 13:30:00.075	98.550000	98.560000
	2016-05-25 13:30:00.076	98.550000	98.560000
	2016-05-25 13:30:00.080	98.550000	98.560000
	2016-05-25 13:30:00.084	98.550000	98.560000
	2016-05-25 13:30:00.086	98.550000	98.630000
	2016-05-25 13:30:00.088	98.650000	98.630000
	2016-05-25 13:30:00.089	98.630000	98.630000
	2016-05-25 13:30:00.104	98.628000	98.630000
	2016-05-25 13:30:00.105	98.620000	98.630000
	2016-05-25 13:30:00.107	98.620000	98.630000
	2016-05-25 13:30:00.115	98.620000	98.630000
	2016-05-25 13:30:00.118	98.620000	98.630000
	2016-05-25 13:30:00.128	98.620000	98.630000
	2016-05-25 13:30:00.129	98.616000	98.630000
	2016-05-25 13:30:00.130	98.610000	98.630000
	2016-05-25 13:30:00.131	98.610000	98.620000

	2016-05-25	13:30:00.135	98.610000	98.620000
	2016-05-25	13:30:00.136	98.610000	98.620000
	2016-05-25	13:30:00.144	98.610000	98.620000
	2016-05-25	13:30:00.145	98.606667	98.628333
GOOG	2016-05-25	13:30:00.023	720.500000	720.930000
	2016-05-25	13:30:00.048	720.500000	720.930000
	2016-05-25	13:30:00.072	720.500000	720.880000
MSFT	2016-05-25	13:30:00.023	51.950000	51.950000
	2016-05-25	13:30:00.041	51.950000	51.950000
	2016-05-25	13:30:00.078	51.942500	51.950000
	2016-05-25	13:30:00.079	51.920000	51.950000
	2016-05-25	13:30:00.130	51.930000	51.950000
	2016-05-25	13:30:00.135	51.920000	51.950000

```
In [17]: mdf.groupby(level='ticker').plot()
```

```
ticker
AAPL      Axes(0.125,0.125;0.775x0.755)
GOOG      Axes(0.125,0.125;0.775x0.755)
MSFT      Axes(0.125,0.125;0.775x0.755)
dtype: object
```

