

6.1 - Altre

April 11, 2017

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

```
In [2]: df = pd.read_csv("tips.csv")
```

```
In [3]: df['sex'].value_counts()
```

```
Male      157
Female     87
Name: sex, dtype: int64
```

```
In [4]: df['size'].value_counts()
```

```
2      156
3       38
4       37
5        5
6        4
1        4
Name: size, dtype: int64
```

```
In [5]: df.describe()
```

	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

```
In [6]: df.quantile(.5)
```

```
total_bill    17.795
tip           2.900
size          2.000
Name: 0.5, dtype: float64
```

```
In [7]: df.std()
```

```
total_bill    8.902412
tip           1.383638
size          0.951100
dtype: float64
```

```
In [8]: df.to_latex()
```

```
Out[8]: '\\begin{tabular}{lrrlllllr}\\n\\toprule\\n{} & total\\_bill & tip & s
```

```
In [9]: df = pd.read_csv("quotes2.csv", parse_dates=['time'])
```

0.1 Funzioni che operano su assi diversi

```
In [10]: df.mean()
```

```
bid    156.903214
ask    156.959464
dtype: float64
```

```
In [11]: df.mean(axis=1)
```

```
0    720.715
1     51.950
2     51.950
3    720.715
4    720.715
5    720.715
6    720.715
7    720.690
8     98.555
9     98.555
10    98.555
11    98.555
12    51.950
13    51.950
```

```
14      51.950
15      51.935
16      51.935
17      98.555
18      98.555
19      98.590
20      98.640
21      98.630
22      98.630
23      98.630
24      98.630
25      98.630
26      98.625
27      98.625
28      98.625
29      98.625
30      98.625
31      98.625
32      98.625
33      98.625
34      98.625
35      98.620
36      98.625
37      98.625
38      98.620
39      51.940
40      51.940
41      98.620
42      98.615
43      98.615
44      51.935
45      98.615
46      98.615
47      98.615
48      98.615
49      98.615
50      98.615
51      98.620
52      98.620
53      98.615
54      98.620
55      98.615
dtype: float64
```

0.2 Funzioni numpy

```
In [12]: import numpy as np
```

```
In [13]: np.where(df.ask > df.bid, df.ask, df.bid)
```

```
Out [13]: array([[ 720.93,   51.95,   51.95,   720.93,   720.93,   720.93,   720.93,
                720.88,   98.56,   98.56,   98.56,   98.56,   51.95,   51.95,
                 51.95,   51.95,   51.95,   98.56,   98.56,   98.63,   98.65,
                 98.63,   98.63,   98.63,   98.63,   98.63,   98.63,   98.63,
                 98.63,   98.63,   98.63,   98.63,   98.63,   98.63,   98.63,
                 98.63,   98.63,   98.63,   98.63,   51.95,   51.95,   98.63,
                 98.62,   98.62,   51.95,   98.62,   98.62,   98.62,   98.62,
                 98.62,   98.62,   98.63,   98.63,   98.63,   98.63,   98.63])
```

```
In [14]: np.diag(df)
```

```
Out [14]: array([Timestamp('2016-05-25 13:30:00.023000'), 'MSFT', 51.95, 720.93], dt
```

0.3 Altro

- categoricals
- to_csv
- to_hdf
- to_excel, to_stata
- join (di cui non ho parlato perché si può vivere senza)
- concatenate
- append

0.4 Altre librerie

- dask
- xarray
- pandas-datareader (mi sono dimenticato di suggerirvelo)
- pandaSDMX