

1.3 - Operazioni tra Series

April 11, 2017

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

```
In [2]: prezzi = pd.Series({'penna' : 2,  
                           'matita' : 1,  
                           'quaderno' : 2.5})
```

```
In [3]: prezzi
```

```
matita      1.0  
penna       2.0  
quaderno    2.5  
dtype: float64
```

```
In [4]: acquisti = pd.Series({'penna' : 3,  
                             'quaderno' : 13})
```

```
In [5]: acquisti
```

```
penna        3  
quaderno     13  
dtype: int64
```

```
In [6]: acquisti2 = pd.Series([3, 13], index=['penna', 'matita'])
```

```
In [7]: # ?  
        acquisti2 == acquisti
```

ValueError

Traceback (most recent call last)

```
<ipython-input-7-e0a6086e790a> in <module>()  
1 # ?
```

```
----> 2 acquisti2 == acquisti
```

```
      /home/pietro/nobackup/repo/pandas/pandas/core/ops.py in wrapper(self, other)
      815             if not self._indexed_same(other):
      816                 msg = 'Can only compare identically-labeled Series objects'
--> 817                 raise ValueError(msg)
      818             return self._constructor(na_op(self.values, other.values),
      819                                     index=self.index, name=name)
```

ValueError: Can only compare identically-labeled Series objects

```
In [8]: acquisti2.sort_index(inplace=True)
```

```
In [9]: acquisti2 = acquisti2.sort_index()
```

```
In [10]: costo = acquisti * prezzi
```

```
In [11]: costo.sum()
```

```
Out[11]: 38.5
```

0.1 Evitare i NaN

```
In [12]: costo
```

```
matita      NaN
penna       6.0
quaderno    32.5
dtype: float64
```

```
In [13]: costo.isnull()
```

```
matita      True
penna       False
quaderno    False
dtype: bool
```

```
In [14]: costo.dropna()
```

```
penna       6.0
quaderno    32.5
dtype: float64
```

```
In [15]: costo.fillna(0)
```

```
matita      0.0  
penna       6.0  
quaderno   32.5  
dtype: float64
```

```
In [16]: acquisti * prezzi
```

```
matita      NaN  
penna       6.0  
quaderno   32.5  
dtype: float64
```

```
In [17]: # ?  
         acquisti *= prezzi
```

```
In [18]: acquisti # != acquisti * prezzi
```

```
penna       6.0  
quaderno   32.5  
dtype: float64
```

```
In [19]: prezzi2 = prezzi.copy()
```

```
In [20]: prezzi2 *= acquisti
```

```
In [21]: prezzi2
```

```
matita      NaN  
penna      12.00  
quaderno   81.25  
dtype: float64
```

```
In [22]: acquisti2
```

```
matita      13  
penna       3  
dtype: int64
```

```
In [23]: # Funziona nonostante l'ordine diverso!
prezzi * acquisti2
```

```
matita      13.0
penna       6.0
quaderno    NaN
dtype: float64
```

0.2 Indici con doppioni

```
In [24]: prezzi = pd.Series([2,1,2.5,3],
                             index=['penna', 'matita', 'quaderno', 'penna'])
```

```
In [25]: prezzi
```

```
penna      2.0
matita     1.0
quaderno   2.5
penna      3.0
dtype: float64
```

```
In [26]: acquisti * prezzi
```

```
matita      NaN
penna      12.00
penna      18.00
quaderno   81.25
dtype: float64
```

```
In [27]: # Semplicemente non fatelo!
```

```
In [28]: prezzi.index.is_unique
```

```
Out[28]: False
```

```
In [29]: acquisti.index.is_unique
```

```
Out[29]: True
```

```
In [30]: prezzi.loc['penna']
```

```
penna      2.0
penna      3.0
dtype: float64
```

```
In [31]: prezzi.index.drop_duplicates()
```

```
Out[31]: Index(['penna', 'matita', 'quaderno'], dtype='object')
```

0.3 Data type

```
In [32]: acquisti = pd.Series({'penna' : 3,  
                             'matita' : 13})
```

```
In [33]: # Non c'è il dtype "str"!  
         acquisti.astype(str)
```

```
matita    13  
penna      3  
dtype: object
```

```
In [34]: acquisti.astype(float)
```

```
matita    13.0  
penna      3.0  
dtype: float64
```