

2.1 - Indexes

July 12, 2017

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

0.1 Duplicated indexes

```
In [2]: df = pd.concat([pd.DataFrame(np.random.randint(1, 20, size=(100,7)))]*2).index
```

```
In [4]: df.index.is_unique
```

```
Out[4]: False
```

```
In [5]: df.index.is_monotonic
```

```
Out[5]: False
```

```
In [6]: df.tail()
```

```
Out[6]:
```

	0	1	2	3	4	5	6
95	12	11	16	13	16	2	7
96	1	3	10	10	7	15	7
97	11	10	17	1	12	12	19
98	9	3	10	13	13	11	14
99	12	7	13	1	13	19	11

```
In [7]: df.loc[0]
```

```
Out[7]: 0    16
1     3
2     8
3    15
4    12
5     4
6    13
Name: 0, dtype: int64
```

```
In [8]: df.loc[0] = range(7)
```

```
In [11]: df.loc[0]
```

```
Out[11]: 0    0
         1    1
         2    2
         3    3
         4    4
         5    5
         6    6
         Name: 0, dtype: int64
```

```
In [12]: df.loc[99] = range(7)
```

```
-----
ValueError                                Traceback (most recent call last)

<ipython-input-12-968bff81e104> in <module>()
----> 1 df.loc[99] = range(7)

/home/pietro/nobackup/repo/pandas/pandas/core/indexing.py in __setitem__(self, key, value)
    177         key = com._apply_if_callable(key, self.obj)
    178         indexer = self._get_setitem_indexer(key)
--> 179         self._setitem_with_indexer(indexer, value)
    180
    181     def _has_valid_type(self, k, axis):

/home/pietro/nobackup/repo/pandas/pandas/core/indexing.py in _setitem_with_indexer(self, indexer, value)
    617         self.obj._consolidate_inplace()
    618         self.obj._data = self.obj._data.setitem(indexer=indexer,
--> 619                                                  value=value)
    620         self.obj._maybe_update_cacher(clear=True)
    621

/home/pietro/nobackup/repo/pandas/pandas/core/internals.py in setitem(self, i, value)
   3201
   3202     def setitem(self, **kwargs):
-> 3203         return self.apply('setitem', **kwargs)
   3204
   3205     def putmask(self, **kwargs):

/home/pietro/nobackup/repo/pandas/pandas/core/internals.py in apply(self, f, b, f)
   3089
   3090         kwargs['mgr'] = self
-> 3091         applied = getattr(b, f)(**kwargs)
```

```

3092             result_blocks = _extend_blocks(applied, result_blocks)
3093
    /home/pietro/nobackup/repo/pandas/pandas/core/internals.py in setitem(self,
703             indexer.dtype == np.bool_ and
704             len(indexer[indexer]) == len(value)):
--> 705         raise ValueError("cannot set using a list-like indexer
706             "with a different length than the
707

```

ValueError: cannot set using a list-like indexer with a different length than the

```
In [13]: df.loc[99]
```

```
Out[13]:
```

	0	1	2	3	4	5	6
99	12	7	13	1	13	19	11
99	12	7	13	1	13	19	11

```
In [14]: ddf = df.reset_index()
```

```
In [15]: ddf.tail()
```

```
Out[15]:
```

	index	0	1	2	3	4	5	6
145	95	12	11	16	13	16	2	7
146	96	1	3	10	10	7	15	7
147	97	11	10	17	1	12	12	19
148	98	9	3	10	13	13	11	14
149	99	12	7	13	1	13	19	11

```
In [16]: ddf.index.is_unique
```

```
Out[16]: True
```

```
In [17]: ddf.index.is_monotonic
```

```
Out[17]: True
```

```
In [18]: ddf.loc[0]
```

```
Out[18]:
```

index	50
0	7
1	4
2	9
3	8
4	8
5	14
6	14

Name: 0, dtype: int64

```
In [19]: %timeit -n 1 -r 1 [ddf.loc[idx] for idx in range(50)]
```

1 loop, best of 1: 5.41 ms per loop

```
In [20]: %timeit -n 1 -r 1 [df.loc[idx] for idx in range(50)]
```

1 loop, best of 1: 6.33 ms per loop

```
In [21]: %timeit -n 1 -r 1 [ddf.loc[idx] for idx in range(50, 100)]
```

1 loop, best of 1: 5.27 ms per loop

```
In [22]: %timeit -n 1 -r 1 [df.loc[idx] for idx in range(50, 100)]
```

1 loop, best of 1: 20.1 ms per loop

```
In [25]: %timeit -n 1 -r 1 ddf.loc[[40]]
```

1 loop, best of 1: 979 μ s per loop

```
In [26]: %timeit -n 1 -r 1 df.loc[[40]]
```

1 loop, best of 1: 1.3 ms per loop

0.2 Adding elements

```
In [27]: df.tail()
```

```
Out[27]:
```

	0	1	2	3	4	5	6
95	12	11	16	13	16	2	7
96	1	3	10	10	7	15	7
97	11	10	17	1	12	12	19
98	9	3	10	13	13	11	14
99	12	7	13	1	13	19	11

```
In [28]: df.loc[101] = range(7)
```

```
In [29]: df.tail()
```

```
Out[29]:
```

	0	1	2	3	4	5	6
96	1	3	10	10	7	15	7
97	11	10	17	1	12	12	19
98	9	3	10	13	13	11	14
99	12	7	13	1	13	19	11
101	0	1	2	3	4	5	6

```
In [30]: a = np.zeros(shape=(5,4))
```

```
In [31]: a
```

```
Out[31]: array([[ 0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.]])
```

```
In [32]: a[4]
```

```
Out[32]: array([ 0.,  0.,  0.,  0.])
```

```
In [33]: a[5] = [1,2,3,4]
```

```
-----  
IndexError                                Traceback (most recent call last)  
  
<ipython-input-33-9524da7d37d0> in <module>()  
----> 1 a[5] = [1,2,3,4]
```

```
IndexError: index 5 is out of bounds for axis 0 with size 5
```

```
In [34]: def fill_a():  
         s = pd.Series()  
         for i in range(1000):  
             s.loc[i] = i*2  
         return s
```

```
In [35]: %timeit -n 1 -r 1 fill_a()
```

```
1 loop, best of 1: 414 ms per loop
```

```
In [36]: def fill_b():  
         s = pd.Series(index=range(1000))  
         for i in range(1000):  
             s.loc[i] = i*2  
         return s
```

```
In [37]: %timeit -n 1 -r 1 fill_b()
```

```
1 loop, best of 1: 68.2 ms per loop
```

```
In [38]: %timeit -n 1 -r 1 df.loc[202] = range(7)
1 loop, best of 1: 2.17 ms per loop
```

```
In [39]: %timeit -n 1 -r 1 df.loc[202] = range(7)
1 loop, best of 1: 931  $\mu$ s per loop
```

```
In [40]: %timeit -n 1 -r 1 df.loc[203] = range(7)
1 loop, best of 1: 2.18 ms per loop
```

```
In [41]: %timeit -n 1 -r 1 df.loc[203] = range(7)
1 loop, best of 1: 720  $\mu$ s per loop
```