

3.1 - Concatenated assignment

July 12, 2017

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

```
In [2]: df = pd.DataFrame(index=range(5), columns=range(5))
```

```
In [3]: df
```

```
Out [3]:
```

	0	1	2	3	4
0	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN

```
In [4]: df.loc[3].loc[4]
```

```
Out [4]: nan
```

```
In [5]: df.loc[3].loc[4] = 3
```

```
In [6]: df
```

```
Out [6]:
```

	0	1	2	3	4
0	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	3
4	NaN	NaN	NaN	NaN	NaN

```
In [7]: df.loc[3:5].loc[:, 4] = 3
```

```
/home/pietro/nobackup/repo/pandas/pandas/core/indexing.py:601: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
self.obj[item_labels[indexer[info_axis]]] = value
```

```
In [8]: df
```

```
Out [8]:
```

	0	1	2	3	4
0	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	3
4	NaN	NaN	NaN	NaN	NaN

0.1 More worrying

```
In [9]: mytmps = df.loc[3]
```

```
In [10]: mytmps.loc[0] = 99
```

```
In [11]: df
```

```
Out [11]:
```

	0	1	2	3	4
0	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN
3	99	NaN	NaN	NaN	3
4	NaN	NaN	NaN	NaN	NaN

0.1.1 Solution

```
In [13]: mytmps = df.loc[3].copy()
```