

The results are in! See what nearly 90,000 developers picked as their most loved, dreaded, and desired coding languages and more in the 2019 Developer Survey.

Setting values on a copy of a slice from a DataFrame [duplicate]

[Ask Question](#)

18

This question already has an answer here:

[How to deal with SettingWithCopyWarning in Pandas?](#) 11 answers

★
3

I have a small dataframe, say this one :

	Mass32	Mass44
12	0.576703	0.496159
13	0.576658	0.495832
14	0.576703	0.495398
15	0.576587	0.494786
16	0.576616	0.494473
...		

I would like to have a rolling mean of column `Mass32` , so I do this:

```
x['Mass32s'] = pandas.rolling_mean(x.Mass32, 5).shift(-2)
```

It works as in I have a new column named `Mass32s` which contains what I expect it to contain but I also get the warning message:

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 the caveats in the documentation:
<http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

I'm wondering if there's a better way to do it, notably to avoid getting this warning message.

edited Jul 17 '15 at 16:32



firelynx

15.4k 3 65 81

asked Jul 17 '15 at 3:58



vincent

110 1 2 8

marked as duplicate by

coldspeed

pandas

Jan 6 at

17:52

This question has been asked before and already has an answer. If those answers do not fully address your question, please [ask a new question](#).

I don't get the warning message, when I run with your sample code can you check if earlier in your code you've set `x` as a copy of a data frame, something like `x = x[x.Mass32.notnull()]` – [maxymoo](#) Jul 17 '15 at 6:30

there was a couple of NAs in the dataframe that apparently were messing with me here. Fixing them

with `fillna(0)` and `.loc` solved it. Thanks – [vincent](#) Jul 20 '15 at 5:01

1 Answer



43



This warning comes because your dataframe `x` is a copy of a slice. This is not easy to know why, but it has something to do with how you have come to the current state of it.



You can either create a proper dataframe out of `x` by doing

```
x = x.copy()
```

This will remove the warning, but it is **not the proper way**

You should be using the `DataFrame.loc` method, as the warning suggests, like this:

```
x.loc[:, 'Mass32s'] = pandas.rollm
```


answered Jul 17 '15 at 15:47



firelynx

15.4k 3 65 81

messing with me here. Fixing them with `fillna(0)` and `.loc` solved it.
Thanks – [vincent](#) Jul 20 '15 at 5:01

the first method sometimes works...
the second method method gets me the following error `indexer = self._get_setitem_indexer(key)` , in `_get_setitem_indexer` raise `IndexingError(key)` `IndexingError: (slice(None, None, None), 'Mass32s')`
– [Lcat](#) Nov 25 '17 at 1:26 

-
- 1 [@Lcat](#) sounds like you have none values in the index of your dataframe. Can you make a new question with some example data? and pass it on to me? – [firelynx](#) Nov 27 '17 at 15:05

Why is this method better? – [mxbi](#) Aug 25 '18 at 14:19

[@mxbi](#) copying the dataframe makes a copy, thus doubles the memory used. Even if you overwrite the variable as in my example `x = x.copy()` , there will be a spike in memory usage. – [firelynx](#) Aug 25 '18 at 19:04
