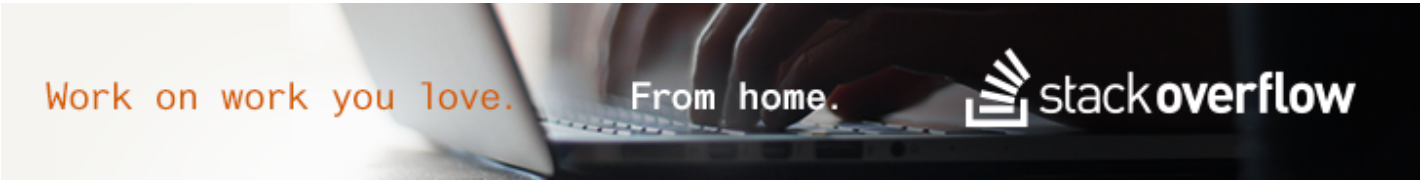
What does axis in pandas mean?



Here is my code to generate a dataframe:

```
import pandas as pd

dff = pd.DataFrame(np.random.randn(1,2),columns=list('AB'))
```

then I got the dataframe:

```
+------------+---------+--------+
|            |  A      |  B     |
+------------+---------+---------
|      0     | 0.626386| 1.52325|
+------------+---------+--------+
```

When I type the commmand :

```
dff.mean(axis=1)
```

I got :

```
0    1.074821
dtype: float64
```

According to the reference of pandas, axis=1 stands for columns and I expect the result of the command to be

```
A    0.626386
B    1.523255
dtype: float64
```

So here is my question: what does axis in pandas mean?

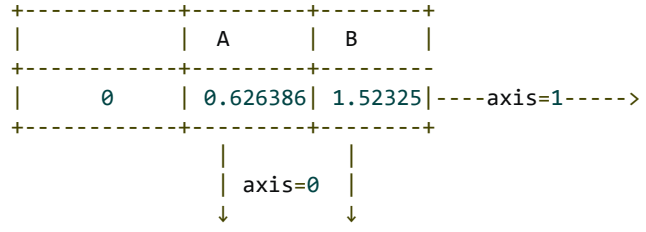
python pandas

edited Mar 3 '14 at 14:43
Peter Westlake
2,853 12 28

asked Mar 3 '14 at 14:41
jerry_sjtu
645 2 13 36

3 Answers

It specifies the axis **along which** the means are computed. By default `axis=0` . This is consistent with the `numpy.mean` usage when `axis` is specified *explicitly* (in `numpy.mean` , axis==None by default, which computes the mean value over the flattened array) , in which `axis=0` along the *rows* (namely, *index* in pandas), and `axis=1` along the *columns*.



edited Mar 3 '14 at 15:43

answered Mar 3 '14 at 14:55

zhangxaochen
9,995 1 21 41

8 Usually axis=0 is said to be "column-wise" (and axis=1 "row-wise"), I think "along the rows" is confusing. (Nice "pic" though :)) – Andy Hayden Mar 3 '14 at 17:43

2 @AndyHayden yeah, but maybe both are a bit confusing, to those who the first time to come across the this ;) – zhangxaochen Mar 4 '14 at 4:13

2 That's certainly true, the picture is **excellent** :) – Andy Hayden Mar 4 '14 at 5:13

6 Also, the reason that `axis=0` indicates aggregating along rows and `axis=1` indicates aggregating along columns is because of how you index into a dataframe. In `df.iloc[row, column]`, `row` is in index position 0 and `column` is in index position 1. Numpy generalizes this to N dimensions, which is where thinking in terms of the axis that the aggregation collapses starts to make more sense than "row-wise" or "column-wise". – Tom Q. Sep 18 '15 at 16:47

The designer of pandas, Wes McKinney, used to work intensively on finance data. Think of columns as stock names and index as daily prices. You can then guess what the default behavior is (i.e., `axis=0`) with respect to this finance data. `axis=1` can be simply thought as 'the other direction'.

For example, the statistics functions, such as `mean()`, `sum()`, `describe()`, `count()` all default to column-wise because it makes more sense to do them for each stock. `sort_index(by=)` also defaults to column. `fillna(method='ffill')` will fill along column because it is the same stock. `dropna()` defaults to row because you probably just want to discard the price on that day instead of throw away all prices of that stock.

Similarly, the square brackets indexing refers to the columns since it's more common to pick a stock instead of picking a day.

edited Jul 15 at 3:37 answered Jul 15 at 2:56

nos

184 9

It means it took the mean based using each column, axis=0 would give you what you think, but axis=1 gives

```
(0.626386+1.52325)/2
1.075
```

answered Mar 3 '14 at 14:47

NightHallow
552 3 12

2 Think if it like you're removing the axis passed to the function. So `dff.mean(axis=1)` removes the `1` axis (the columns) by aggregating the mean function over them. – TomAugspurger Mar 3 '14 at 14:51