pandas get column average/mean

Ask Question

93

14

I can't get the average or mean of a column in pandas. A have a dataframe. Neither of things I tried below gives me the average of the column `weight`

```
>>> allDF
         ID           birthyear  weight
0        619040       1962       0.1231231
1        600161       1963       0.981742
2      25602033       1963       1.3123124
3        624870       1987       0.94212
```

The following returns several values, not one:

```
allDF[['weight']].mean(axis=1)
```

So does this:

```
allDF.groupby('weight').mean()
```

python pandas

edited Dec 27 '16 at 13:50

asked Jun 24 '15 at 21:22

PepperoniPizza
3,811 4 37 69

2 You should accept DSM's answer so this question does not remain unanswered – EdChum Jun 25 '15 at 7:46

Possible duplicate of [how to get the average of dataframe column values](https://stackoverflow.com) – Jeru Luke Jul 24 '17 at 10:31

`df.groupby('weight')` wasn't what you wanted, because it split the df into separate columns, each with a distinct value of weight. Instead of just `df['weight'].mean()` – smci Feb

164

✓

If you only want the mean of the `weight` column, select the column (which is a Series) and call `.mean()` :

```
In [479]: df
Out[479]:
         ID  birthyear    weight
0    619040       1962  0.123123
1    600161       1963  0.981742
2  25602033       1963  1.312312
3    624870       1987  0.942120

In [480]: df["weight"].mean()
Out[480]: 0.83982437500000007
```

answered Jun 24 '15 at 21:26

DSM
216k 36 412 381

and what if I wanted to get a mean of each and every column? – Chris Jun 11 '18 at 14:55

2 @Chris df.describe() – Abhishek Poojary Aug 1 '18 at 17:20

@Chris df.mean() gives you the weight of each column and returns it in a series. – emschorsch Feb 22 at 0:41

7

Try `df.mean(axis=0)` , `axis=0` argument calculates the column wise mean of the dataframe so the result will be `axis=1` is row wise mean so you are getting multiple values.

edited Mar 7 at 16:35

Soufiane Sabiri
32 9

answered Aug 8 '18 at 16:38

Chandu
71 1 2

5

Do try to give `print (df.describe())` a shot. I hope it will be very helpful to get an overall description of your dataframe.

`display(df.describe())` is better (in Jupyter Notebooks) because `display` from ipython provides formatted HTML rather than ASCII, which is more visually useful/pleasing. – Zhanwen Chen Apr 5 at 16:28

0

you can use

```
df.describe()
```

you will get basic statistics of the dataframe and to get mean of specific column you can use

```
df["columnname"].mean()
```

answered Nov 28 '18 at 15:41

Arun Singh
21 1

This is a duplicate of the answers mentioned above. – Mehdi Boukhechba Dec 12 '18 at 14:29