coursera

Discussion Forums

Week 3

This thread is only visible to learners in the December 5 - January 8 session.

← Week 3

JP

Q. 11. The right answer might be wrong !

juan carlos basto pineda Week 3 · a day ago · Edited

Hi Everyone, I have a challenging test here.

I calculated the standard deviation of the populations grouped by continent in 2 different ways, and they do NOT coincide in all the decimal digits. The first solution, which seems more compact and fancy, allowed me to finally get a grade of 100%, after struggling for days trying to find out what was wrong with my initial approach. Strikingly, a manual inspection confirms that my initial method is returning the right number and not the other way around, which implies that the official 'right' answer might be wrong (!)

What is more surprising is that both methods resort to the same function, np.std, so how can they return different answers?

The code chunk below should allow you to reproduce the problem. I am looking forward to your comments, and please let me know if I am overlooking something here.

Thanks!

Ps. Note that for those continents encompassing only one country the standard deviation is not well defined. The method that returned the "right" solution placed NaN's there, while my initial approach put 0's. Yet this is not the source of the conflict, I placed np.NaN by hand where necessary and the grader continued complaining about a mismatch between my numbers and the "right" answer in the column 'std'.

Help Center

```
# This is my merged dataframe, showing only the columns of interest
                Country    population      continent
0                Canada  3.523986e+07  North America
1                Brazil  2.059153e+08  South America
2                 Italy  5.990826e+07         Europe
3                France  6.383735e+07         Europe
4         United States  3.176154e+08  North America
5             Australia  2.331602e+07      Australia
6                  Iran  7.707563e+07           Asia
7                 India  1.276731e+09           Asia
8        United Kingdom  6.387097e+07         Europe
9                 China  1.367645e+09           Asia
10  Russian Federation  1.435000e+08         Europe
11                Japan  1.274094e+08           Asia
12          South Korea  4.980543e+07           Asia
13              Germany  8.036970e+07         Europe
14                Spain  4.644340e+07         Europe


# This is the solution that got a 100% of the points and its output:
>> df.groupby('continent')['population'].agg({'std': np.std})
Asia           6.790979e+08
Australia               NaN
Europe         3.464767e+07
North America  1.996696e+08
South America           NaN

# This was my initial approach and its output:
>> new.groupby('continent').apply(lambda df: np.std(df['population']))
Asia             6.074036e+08
Australia        0.000000e+00
Europe           3.162885e+07
North America    1.411878e+08
South America    0.000000e+00

# This is a manual verification of the answer for 'Asia':
>> np.std(np.array([7.707563e+07,1.276731e+09,1.367645e+09,1.274094e+08
    ,4.980543e+07]))
# It returns the same value as my initial approach, which was graded as
    wrong
607403627.24072623

```

0 Upvote · Follow 0 · Reply to juan carlos basto pineda

Earliest **Top** **Most Recent**

SS

Shijian Su Mentor · 3 minutes ago · Edited

Hi Juan,

Thank you for your finding here!

I also have the wondering after I saw your finding, so I did a small testing to Asia countries in Excel,

	A	B
1	China	1367645000.00
2	India	1276731000.00
3	Iran	77075630.00
4	Japan	127409400.00
5	South Korea	49805430.00
6	STDEV(B1,B2,B3,B4,B5)	679097900.1

It gives me a result about 6.790979e+08.

So I try it in Python, it turns out,

```
import numpy as np

asia_values = np.array([7.707563e+07, 1.276731e+09, 1.367645e+09, 1
    .274094e+08, 4.980543e+07])
print(np.std(asia_values))
# 607403627.241

print(np.std(asia_values, ddof=1))
# 679097900.145
```

I found a similiar discussion in stackoverflow

http://stackoverflow.com/questions/27600207/why-does-numpy-std-give-a-different-result-to-matlab-std

Hope above information would help.

Thank you,

Shijian

0 · Reply

1

SS Reply

Follow this question to get notifications of replies

Reply