

# How can I calculate the standard deviation for each row in a Pandas DataFrame?

Authored by  
**stats writer**

June 25, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I calculate the standard deviation for each row in a Pandas DataFrame?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152323>

Calculating the standard deviation for each row in a Pandas DataFrame can be achieved by using the "std" function. This function calculates the standard deviation for each column by default, but by specifying the "axis=1" parameter, it can be applied to each row instead. This will result in a new column being added to the DataFrame, containing the standard deviation values for each row. This method is useful for analyzing the variability within each row of a dataset and can provide valuable insights for data analysis and decision making.

## **Pandas: Calculate Standard Deviation for Each Row**

**You can use the following basic syntax to calculate the standard deviation of values for each row in a pandas DataFrame:**

```
df.std(axis=1, numeric_only=True)
```

**The argument axis=1 tells pandas to perform the calculation for each row (instead of each column) and numeric\_only=True tells pandas to only consider numeric columns when performing the calculation.**

**The following example shows how to use this syntax in practice.**

### **Example: Calculate Standard Deviation for Each Row in Pandas**

**Suppose we have the following pandas DataFrame that contains information about the points scored by various basketball players during four different games:**

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'player': ,  
'game1': ,  
'game2': ,  
'game3': ,  
'game4': })
```

```
#view DataFrame
```

```
print(df)
```

```
player game1 game2 game3 game4  
0 A 18 5 11 9  
1 B 22 7 8 8  
2 C 19 7 10 8  
3 D 14 9 6 9  
4 E 14 12 6 14  
5 F 11 9 5 15  
6 G 20 9 9 10  
7 H 28 4 12 11
```

We can use the following syntax to calculate the standard deviation of points scored by each player:

```
#calculate standard deviation for each row  
df.std(axis=1, numeric_only=True)
```

```
0 5.439056  
1 7.182154  
2 5.477226  
3 3.316625  
4 3.785939  
5 4.163332  
6 5.354126  
7 10.144785  
dtype: float64
```

**Here's how to interpret the output:**

**The standard deviation of points scored by player A is 5.439. The standard deviation of points scored by player B is 7.182. The standard deviation of points scored by player C is 5.477.**

**And so on.**

**Note that the `std()` function calculates the sample standard deviation by default.**

If you would instead like to calculate the population standard deviation, you must use the argument `ddof=0`:

```
#calculate population standard deviation for each row  
df.std(axis=1, ddof=0, numeric_only=True)
```

```
0 4.747351
```

```
1 5.881366
```

```
2 4.807037
```

```
3 3.384910
```

```
4 3.983518
```

```
5 3.915150
```

```
6 4.892772
```

```
7 8.091179
```

```
dtype: float64
```

To assign the standard deviation values to a new column, you can use the following syntax:

```
#add new column to display standard deviation for each  
row
```

```
df = df.std(axis=1, numeric_only=True)
```

```
#view updated DataFrame
```

```
print(df)
```

```
player game1 game2 game3 game4 points_std
0 A 18 5 11 9 5.439056
1 B 22 7 8 8 7.182154
2 C 19 7 10 8 5.477226
3 D 14 9 6 9 3.316625
4 E 14 12 6 14 3.785939
5 F 11 9 5 15 4.163332
6 G 20 9 9 10 5.354126
7 H 28 4 12 11 10.144785
```

The standard deviation of values for each row in the game1, game2, game3 and game4 columns is now shown in the points\_std column.

The following tutorials explain how to perform other common operations in pandas: