

What is the difference between axis=0 and axis=1 in Pandas?

Authored by
stats writer

May 12, 2024

RECOMMENDED CITATION

stats writer (2024). *What is the difference between axis=0 and axis=1 in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=144015>

Axis=0 and axis=1 are two important parameters in Pandas that indicate the direction of an operation within a DataFrame. In simple terms, axis=0 refers to operations performed vertically, i.e. along the rows of a DataFrame, while axis=1 refers to operations performed horizontally, i.e. along the columns of a DataFrame. This means that the choice of axis can greatly impact the result of an operation, as it determines the direction in which the data is being manipulated. Therefore, understanding the difference between axis=0 and axis=1 is crucial for effectively working with Pandas DataFrames.

The Difference Between axis=0 and axis=1 in Pandas

Many functions in require that you specify an axis along which to apply a certain calculation.

Typically the following rule of thumb applies:

axis=0: Apply the calculation "column-wise"
axis=1: Apply the calculation "row-wise"

The following examples show how to use the axis argument in different scenarios with the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'points': ,
'assists': ,
```

```
'rebounds': })
```

```
#view DataFrame
```

```
df
```

```
team points assists rebounds
```

```
0 A 25 5 11
```

```
1 A 12 7 8
```

```
2 B 15 7 10
```

```
3 B 14 9 6
```

```
4 B 19 12 6
```

```
5 B 23 9 5
```

```
6 C 25 9 9
```

```
7 C 29 4 12
```

Example 1: Find Mean Along Different Axes

We can use axis=0 to find the mean of each column in the DataFrame:

```
#find mean of each column
```

```
df.mean(axis=0)
```

```
points 20.250
```

```
assists 7.750
```

```
rebounds 8.375
```

dtype: float64

The output shows the mean value of each numeric column in the DataFrame.

Notice that pandas automatically avoids calculating the mean of the 'team' column because it's a character column.

We can also use axis=1 to find the mean of each row in the DataFrame:

```
#find mean of each row  
df.mean(axis=1)
```

```
0 13.666667
```

```
1 9.000000
```

```
2 10.666667
```

```
3 9.666667
```

```
4 12.333333
```

```
5 12.333333
```

```
6 14.333333
```

```
7 15.000000
```

dtype: float64

From the output we can see:

The mean value in the first row is 13.667. The mean value in the second row is 9.000. The mean value in the third row is 10.667.

And so on.

Example 2: Find Sum Along Different Axes

We can use axis=0 to find the sum of specific columns in the DataFrame:

```
#find sum of 'points' and 'assists' columns  
df].sum(axis=0)
```

```
points 162  
assists 62  
dtype: int64
```

We can also use axis=1 to find the sum of each row in the DataFrame:

```
#find sum of each row  
df.sum(axis=1)
```

```
0 41
```

1 27

2 32

3 29

4 37

5 37

6 43

7 45

dtype: int64

Example 3: Find Max Along Different Axes

We can use axis=0 to find the max value of specific columns in the DataFrame:

```
#find max of 'points', 'assists', and 'rebounds' columns  
df].max(axis=0)
```

points 29

assists 12

rebounds 12

dtype: int64

We can also use axis=1 to find the max value of each row in the DataFrame:

#find max of each row

df.max(axis=1)

0 25

1 12

2 15

3 14

4 19

5 23

6 25

7 29

dtype: int64

From the output we can see:

The max value in the first row is 25. The max value in the second row is 12. The max value in the third row is 15.

And so on.

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