

How can I sort a pivot table in Pandas based on the values in a specific column?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I sort a pivot table in Pandas based on the values in a specific column?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153660>

A pivot table in Pandas is a powerful tool for organizing and summarizing data. It allows for easy analysis and comparison of different variables and their relationships. However, to further enhance the functionality of a pivot table, it is often necessary to sort the data based on the values in a specific column. This can be achieved by using the sorting function in Pandas, which allows for the sorting of data in ascending or descending order based on a chosen column. By sorting the pivot table, it becomes easier to identify patterns and trends in the data, making it a valuable tool for data analysis. Additionally, sorting a pivot table can also help with data visualization, as it allows for the creation of more organized and visually appealing charts and graphs. Overall, sorting a pivot table in Pandas based on a specific column is a simple yet effective way to further enhance the usefulness of this data analysis tool.

Pandas: Sort Pivot Table by Values in Column

You can use the following basic syntax to sort a pandas pivot table based on the values in a column:

```
my_pivot_table.sort_values(by=, ascending=False)
```

This particular example sorts the values in a pivot table called `my_pivot_table` based on the values in `some_column` in descending order.

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

Example: Sort Pandas Pivot Table by Values in Column

Suppose we have the following pandas DataFrame that contains information about various basketball players:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'team': ,  
'points': ,  
'assists': })
```

```
#view DataFrame
```

```
print(df)
```

```
team points assists
```

```
0 A 4 2
```

```
1 A 4 2
```

```
2 A 2 5
```

```
3 A 8 5
```

```
4 B 9 4
```

```
5 B 5 7
```

```
6 B 5 5
```

```
7 B 7 3
```

```
8 C 8 9
```

```
9 C 8 8
```

```
10 C 4 4
```

```
11 C 3 4
```

We can use the following code to create a pivot table in

pandas that shows the sum of the values in the points and assists columns for each team:

```
#create pivot table
```

```
df_pivot = df.pivot_table(index=, values=,  
aggfunc='sum')
```

```
#view pivot table
```

```
print(df_pivot)
```

```
assists points
```

```
team
```

```
A 14 18
```

```
B 19 26
```

```
C 25 23
```

By default, pandas sorts the rows of the pivot table in alphabetical order based on the value in the index column, which happens to be the team column.

However, we can use the `sort_values()` function to instead sort the rows of the pivot table based on the values in the points column:

```
#sort pivot table by value in 'points' column in
```

descending order

```
sorted_df_pivot = df_pivot.sort_values(by=,  
ascending=False)
```

#view sorted pivot table

```
print(sorted_df_pivot)
```

assists points

team

B 19 26

C 25 23

A 14 18

Notice that the rows of the pivot table are now sorted based on the values in the points column.

Note that if you leave off the ascending=False argument, the rows will be sorted by the values in the points column in ascending order instead:

#sort pivot table by value in 'points' column in ascending order

```
sorted_df_pivot = df_pivot.sort_values(by=)
```

#view sorted pivot table

```
print(sorted_df_pivot)
```

```
assists points
```

```
team
```

```
A 14 18
```

```
C 25 23
```

```
B 19 26
```

Notice that the rows in the pivot table are now sorted by the values in the points column in ascending (smallest to largest) order.

Note #1: You can also sort by multiple columns in the pivot table by passing multiple values to the `by` argument within the `sort_values()` function.

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