

How can I calculate the rank of values within a group using Pandas' GroupBy object?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I calculate the rank of values within a group using Pandas' GroupBy object?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=159401>

The Pandas' GroupBy object is a powerful tool for performing group-wise operations on data. One such operation is calculating the rank of values within a group. This can be achieved by first grouping the data using the GroupBy object and then applying the rank() function to the desired column. This function assigns a numerical rank to each value within the group, with the highest value having a rank of 1 and the lowest value having a rank equal to the number of values in the group. This allows for easy comparison and identification of the highest and lowest values within the group. Overall, using the GroupBy object in combination with the rank() function provides an efficient and effective way to calculate the rank of values within a group in a Pandas DataFrame.

Pandas: Calculate Rank in a GroupBy Object

You can use the following syntax to calculate the rank of values in a GroupBy object in pandas:

```
df = df.groupby().rank()
```

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

Example: Calculate Rank in a GroupBy Object

Suppose we have the following pandas DataFrame that shows the points scored by basketball players on various teams:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'team': ,
```

```
'points': })
```

```
#view DataFrame
```

```
print(df)
```

```
team points
```

```
0 A 10
```

```
1 A 10
```

```
2 A 12
```

```
3 A 15
```

```
4 B 19
```

```
5 B 23
```

```
6 C 20
```

```
7 C 20
```

```
8 C 26
```

We can use the following syntax to calculate the rank of the points values for each team:

```
#add ranking column to data frame
```

```
df = df.groupby().rank()
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points points_rank
```

```
0 A 10 1.5
```

```
1 A 10 1.5
```

```
2 A 12 3.0
```

```
3 A 15 4.0
```

```
4 B 19 1.0
```

```
5 B 23 2.0
```

```
6 C 20 1.5
```

```
7 C 20 1.5
```

```
8 C 26 3.0
```

By default, the `rank()` function assigns ranking values in ascending order and uses the average rank when ties are present.

However, we can use the `method` and `ascending` arguments to rank the values in a different manner:

```
#add ranking column to data frame
```

```
df = df.groupby().rank('dense', ascending=False)
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points points_rank
```

0 A 10 3.0
1 A 10 3.0
2 A 12 2.0
3 A 15 1.0
4 B 19 2.0
5 B 23 1.0
6 C 20 2.0
7 C 20 2.0
8 C 26 1.0

This method assigns a value of 1 to the largest value in each group.

You can find a complete list of ranking methods you can use with the rank() function .

Note: You can find the complete documentation for the GroupBy operation in pandas .

Additional Resources

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