How can I get the top N rows for each group in a Pandas DataFrame?

Authored by

stats writer

June 27, 2024

One can obtain the top N rows for each group in a Pandas DataFrame by using the "groupby" function to group the data based on a specific column or set of columns. Then, the "apply" function can be used to apply a function, such as "head(N)", which will return the top N rows for each group. This method allows for efficient and organized retrieval of data from a large DataFrame based on specific groupings.

Pandas: Get Top N Rows by Group

You can use the following basic syntax to get the top N rows by group in a pandas DataFrame:

df.groupby('group_column').head(2).reset_index(drop=True)

This particular syntax will return the top 2 rows by group.

Simply change the value inside the head() function to return a different number of top rows.

The following examples show how to use this syntax with the following pandas DataFrame:

import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'position': ,

'points': })

#view DataFrame
print(df)

team position points
0 A G 5
1 A G 7
2 A G 7
3 A F 9
4 A F 12
5 B G 9
6 B G 9
7 B F 4
8 B F 7
9 B F 7

Example 1: Get Top N Rows Grouped by One Column

The following code shows how to return the top 2 rows, grouped by the team variable:

#get top 2 rows grouped by team
df.groupby('team').head(2).reset_index(drop=True)

team position points

0 A G 5

1 A G 7

2 B G 9

3 B G 9

The output displays the top 2 rows, grouped by the team variable.

Example 2: Get Top N Rows Grouped by Multiple Columns

The following code shows how to return the top 2 rows, grouped by the team and position variables:

#get top 2 rows grouped by team and position

df.groupby().head(2).reset_index(drop=True)

team position points

0 A G 5

1 A G 7

2 A F 9

3 A F 12

4 B G 9

5 B G 9

6 B F 4

7 B F 7

The output displays the top 2 rows, grouped by the team and position variables.

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