How can I use Pandas to implement a “Group By Having” function in my data analysis?

Authored by

stats writer

June 26, 2024

Pandas is a popular data analysis library in Python that offers various functions to manipulate and summarize data. One useful feature is the "group by having" function, which allows users to group data based on a specific condition and perform calculations on the resulting groups. This function is especially useful when analyzing large datasets and identifying patterns or trends within subgroups. By using the "group by having" function in Pandas, users can efficiently filter and aggregate data to gain valuable insights and make informed decisions. This functionality makes Pandas a powerful tool for data analysis and enables users to effectively handle complex data manipulations.

Pandas: A Simple Formula for "Group By Having"

You can use the following basic syntax to perform the equivalent of a SQL "GROUP BY HAVING" statement in pandas:

df.groupby('some_column').filter(lambda x: some condition)

The following examples show how to use this syntax in practice 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 30
1 A F 22
2 A F 19
3 B G 14
4 B F 14
5 B F 11
6 C G 20
7 C G 28

Example 1: Pandas Group By Having with Count

The following code shows how to group the rows by the value in the team column, then filter for only the teams that have a count greater than 2:

#group by team and filter for teams with count > 2df.groupby('team').filter(lambda x: len(x) > 2)

team position points
0 A G 30
1 A F 22

2 A F 19

3 B G 14

4 B F 14

5 B F 11

Notice that only the rows with a team value of 'A' or 'B' are returned since these are the two teams that have a count greater than 2.

Example 2: Pandas Group By Having with Mean

The following code shows how to group the rows by the value in the team column, then filter for only the teams that have a mean points value greater than 20:

#group by team and filter for teams with mean points > 20df.groupby('team').filter(lambda x: x.mean() > 20)

team position points

0 A G 30

1 A F 22

2 A F 19

6 C G 20

7 C G 28

Notice that only the rows with a team value of 'A' or 'C' are returned since these are the two teams that have a mean points value greater than 20.

Example 3: Pandas Group By Having with Sum

The following code shows how to group the rows by the value in the team column, then filter for only the teams that have a sum of points equal to exactly 48:

#group by team and filter for teams with sum of points equal to 48df.groupby('team').filter(lambda x: x.sum() == 48)

team position points
6 C G 20
7 C G 28

Notice that only the rows with a team value of 'C' are returned since this is the one team that has a sum of points equal to 48.