How can I use group by with a where condition in Pandas?

Authored by

stats writer

June 25, 2024

Pandas is a popular data analysis library in Python that offers powerful tools for manipulating and organizing data. One useful feature in Pandas is the ability to group data using the "group by" function, which allows for aggregation and analysis of data based on specific criteria. In addition to this, Pandas also offers a "where" condition, which allows for filtering of data based on certain conditions. By combining these two features, users can efficiently analyze their data by grouping it based on a specific condition. This allows for targeted and focused analysis, making it a valuable tool for data scientists and analysts.

Pandas: Use Group By with Where Condition

The easiest way to use group by with a where condition in pandas is to use the query() function:

df.query("team == 'A'").groupby().mean().reset_index()

This particular example example calculates the mean value of points, grouped by position, where team is equal to 'A' in some pandas DataFrame.

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

Example: How to Use Group By with Where Condition in Pandas

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

import pandas as pd

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

#view DataFrame
print(df)

team position points
0 A G 22
1 A G 14
2 A F 15
3 A F 10
4 A F 8
5 B G 29
6 B G 33
7 B F 18

We can use the following code to calculate the mean value of points, grouped by position, where team is equal to 'A':

#calculate mean value of points, grouped by position, where team == 'A'

```
df.query("team == 'A'").groupby().mean().reset_index()

position points
0 F 11.0
1 G 18.0
```

From the output we can see:

The mean points value for players in position 'F' is on team A is 11.The mean points value for players in position 'G' on team A is 18.

Note that we can also use the & operator in the query() function to query for rows where multiple conditions are met.

For example, the following code shows how to calculate the mean value of points, grouped by position, where team is equal to 'A' *and* position is equal to 'G':

```
#calculate mean value of points by position where team is 'A' and position is 'G'
df.query("team=='A' & position=='G'").groupby().mean().reset_index()

position points
```

0 G 18.0

From the output we can see that the mean points value for players in position 'G' on team A is 18.

Since we specified two conditions in the query() function, only the rows that met both conditions were used.