How can I group and aggregate data by multiple columns in Pandas?

Authored by

stats writer

April 18, 2024

Pandas is a popular data analysis library in Python that allows users to efficiently manipulate and analyze large datasets. One of its key functionalities is the ability to group and aggregate data by multiple columns. This means that users can group the data based on multiple criteria, such as age and gender, and then perform calculations or apply functions on the grouped data. This allows for a more detailed and comprehensive analysis of the data. To achieve this, users can use the "groupby" and "agg" functions in Pandas, which allow for flexible and customizable grouping and aggregation of data. By utilizing these functions, users can easily organize their data and gain valuable insights from their analysis.

Pandas: Group and Aggregate by Multiple Columns

Often you may want to group and aggregate by multiple columns of a pandas DataFrame.

Fortunately this is easy to do using the pandas .groupby() and .agg() functions.

This tutorial explains several examples of how to use these functions in practice.

Example 1: Group by Two Columns and Find Average

Suppose we have the following pandas DataFrame:

```
import pandas as pd

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

'rebounds': })

#view DataFrame
print(df)

team position assists rebounds
0 A G 5 11
1 B G 7 8
2 B F 7 10
3 B G 8 6
4 B F 5 6
5 M F 7 9
6 M C 6 6
7 M C 9 10

The following code shows how to group by columns 'team' and 'position' and find the mean assists:

df.groupby().agg({'assists': }).reset_index()

team position assists
mean
0 A G 5.0
1 B F 6.0
2 B G 7.5

3 M C 7.5
4 M F 7.0

The output tells us:

The mean assists for players in position G on team A is 5.0.The mean assists for players in position F on team B is 6.0.The mean assists for players in position G on team B is 7.5.

And so on.

We can also use the following code to rename the columns in the resulting DataFrame:

```
#group by team and position and find mean assists
new = df.groupby().agg({'assists': }).reset_index()

#rename columns
new.columns =

#view DataFrameprint(new)
```

team pos mean_assists
0 A G 5.0
1 B F 6.0

2 B G 7.5

3 M C 7.5

4 M F 7.0

Example 2: Group by Two Columns and Find Multiple Stats

Assume we use the same pandas DataFrame as the previous example:

```
import pandas as pd

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

The following code shows how to find the median and max number of rebounds, grouped on columns 'team' and 'position':

```
df.groupby().agg({'rebounds': }).reset_index()
```

team position rebounds

median max

0 A G 11 11

1 B F 8 10

2 B G 7 8

3 M C 8 10

4 M F 9 9

The output tells us:

The median rebounds assists for players in position G on team A is 11.The max rebounds for players in position G on team A is 11.The median rebounds for players in position F on team B is 8.The max rebounds for players in position F on team B is 10.

And so on.

How to Filter a Pandas DataFrame on Multiple Conditions

How to Count Missing Values in a Pandas DataFrame

How to Stack Multiple Pandas DataFrames