How can I use the GroupBy function in Pandas to calculate the number of values falling within specific bins?

Authored by

stats writer

June 29, 2024

The GroupBy function in Pandas is a useful tool for organizing and analyzing data. This function allows you to group data by a specific column or label, and then perform calculations on the individual groups. One application of the GroupBy function is to calculate the number of values falling within specific bins. This can be achieved by first creating the bins using the Pandas cut function, which assigns each value to a specific bin based on a given range. Then, using the GroupBy function, you can group the data by the bins and count the number of values in each group. This allows for a quick and efficient way to analyze data and gain insights into the distribution of values.

Pandas: Use GroupBy with Bin Counts

You can use the following syntax to calculate the bin counts of one variable grouped by another variable in pandas:

#define bins
groups = df.groupby()

#display bin count by group variable
groups.size().unstack()

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

Example: Use GroupBy with Bin Counts in Pandas

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 4
1 A 7
2 A 7
3 A 11
4 A 12
5 A 15
6 A 19
7 A 19
8 B 5
9 B 5
10 B 11
11 B 12
12 B 14
13 B 14
14 B 15
```

15 B 15

We can use the following syntax to calculate the frequency of points for each team, grouped into specific bins:

#define groups
groups = df.groupby()])

#display bin count grouped by team
groups.size().unstack()

points (0, 10] (10, 15] (15, 20]
team
A 3 3 2
B 2 6 0

Here's how to interpret the output:

A total of 3 players on team A scored between 0 and 10 points.A total of 3 players on team A scored between 10 and 15 points.A total of 2 players on team A scored between 15 and 20 points.

And so on.

Note that we can specify whichever bins we'd like within the pd.cut() function.

For example, we could define just two bins:

```
#define groups
groups = df.groupby()])

#display bin count grouped by team
groups.size().unstack()
```

```
points (0, 10] (10, 20]
team
A 3 5
B 2 6
```

Here's how to interpret the output:

A total of 3 players on team A scored between 0 and 10 points.A total of 5 players on team A scored between 10 and 20 points.A total of 2 players on team B scored between 0 and 10 points.A total of 6 players on team B scored between 10 and 20 points.

Note 1: You can find the complete documentation for the GroupBy function .

Additional Resources

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