How can I group a Pandas DataFrame by a range of values?

Authored by

stats writer

June 24, 2024

Grouping a Pandas DataFrame by a range of values refers to the process of dividing the data into groups based on a specified range of values. This can be achieved by using the "groupby" function in Pandas, which allows for grouping data based on a specific column or set of columns. By specifying a range of values, the data can be grouped into smaller, more manageable subsets, making it easier to analyze and draw conclusions from the data. This method is particularly useful for large datasets, as it helps to organize the data and identify patterns within specific ranges. Overall, grouping a Pandas DataFrame by a range of values is a powerful tool for data analysis and can provide valuable insights into the data.

Pandas: Groupby Range of Values

You can use the following syntax to use the groupby() function in pandas to group a column by a range of values before performing an aggregation:

df.groupby(pd.cut(df,)).sum()

This particular example will group the rows of the DataFrame by the following range of values in the column called my_column:

(0, 25](25, 50](50, 75](75, 100]

It will then calculate the sum of values in all columns of the DataFrame using these ranges of values as the groups.

The following example shows how to use this syntax in

practice.

Example: How to Groupby Range of Values in Pandas

Suppose we have the following pandas DataFrame that contains information about the size of different retail stores and their total sales:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'store_size': ,
'sales': })

#view DataFrame
print(df)
```

```
store_size sales
0 14 15
1 25 18
2 26 24
3 29 25
4 45 20
5 58 35
6 67 34
7 81 49
```

8 90 44

9 98 49

We can use the following syntax to group the DataFrame based on specific ranges of the store_size column and then calculate the sum of every other column in the DataFrame using the ranges as groups:

#group by ranges of store_size and calculate sum of all columns

df.groupby(pd.cut(df,)).sum()

store_size sales

store_size

(0, 25] 39 33

(25, 50] 100 69

(50, 75] 125 69

(75, 100] 269 142

From the output we can see:

For rows with a store_size value between 0 and 25, the sum of store_size is 39 and the sum of sales is 33.For rows with a store_size value between 25 and 50, the sum of store_size is 100 and the sum of sales is 69.

And so on.

If you'd like, you can also calculate just the sum of sales for each range of store_size:

```
#group by ranges of store_size and calculate sum of sales
df.groupby(pd.cut(df, )).sum()
```

```
store_size
(0, 25] 33
(25, 50] 69
(50, 75] 69
(75, 100] 142
Name: sales, dtype: int64
```

You can also use the NumPy arange() function to cut a variable into ranges without manually specifying each cut point:

```
import numpy as np

#group by ranges of store_size and calculate sum of sales
df.groupby(pd.cut(df, np.arange(0, 101, 25))).sum()
```

store_size
(0, 25] 33
(25, 50] 69
(50, 75] 69
(75, 100] 142
Name: sales, dtype: int64

Notice that these results match the previous example.

Note: You can find the complete documentation for the NumPy arange() function .

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