

Question: How can I get the group after using the groupby() function in Pandas?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *Question: How can I get the group after using the groupby() function in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153684>

The groupby() function in Pandas allows you to group data based on one or more columns in a DataFrame. After using this function, you can retrieve the grouped data by accessing the specific group using the get_group() method. This method takes in the group name or label as a parameter and returns the corresponding group as a new DataFrame. This allows you to further analyze or manipulate the grouped data as needed.

Pandas: Get Group After Using groupby()

You can use the following methods to get a specific group after using the groupby() function on a pandas DataFrame:

Method 1: Get Group After Using groupby()

```
grouped_df.get_group('A')
```

Method 2: Get Specific Columns of Group After Using groupby()

```
grouped_df].get_group('A')
```

The following examples show how to use each method in practice with the following pandas DataFrame:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'store': ,  
'sales': ,  
'refunds': })
```

```
#view DataFrame  
print(df)
```

```
store sales refunds
```

```
0 A 12 4
```

```
1 A 15 8
```

```
2 A 24 7
```

```
3 A 24 7
```

```
4 B 14 10
```

```
5 B 19 5
```

```
6 B 12 4
```

```
7 B 38 11
```

Example 1: Get Group After Using groupby()

The following code shows how to use the groupby() function to group the rows by store name, then use the get_group() function to retrieve all rows that belong to the group with the group name 'A':

```
#group rows of DataFrame based on value in 'store'
```

column

```
grouped_stores = df.groupby()
```

```
#get all rows that belong to group name 'A'
```

```
grouped_stores.get_group('A')
```

```
store sales refunds
```

```
0 A 12 4
```

```
1 A 15 8
```

```
2 A 24 7
```

```
3 A 24 7
```

Notice that get_group() returns all rows that belong to the group with the group name 'A'.

Example 2: Get Specific Columns of Group After Using groupby()

The following code shows how to use the groupby() function to group the rows by store name, then use the get_group() function to retrieve all rows that belong to the group with the group name 'A' only for the 'sales' and 'refunds' columns:

```
#group rows of DataFrame based on value in 'store'  
column
```

```
grouped_stores = df.groupby()
```

```
#get all rows that belong to group name 'A' for sales  
and refunds columns
```

```
grouped_stores].get_group('A')
```

```
store refunds
```

```
0 A 4
```

```
1 A 8
```

```
2 A 7
```

```
3 A 7
```

Notice that `get_group()` returns all rows that belong to the group with the group name 'A' for the 'sales' and 'refunds' columns only.

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