

How can I convert the output of a Pandas GroupBy operation into a DataFrame format?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I convert the output of a Pandas GroupBy operation into a DataFrame format?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=157809>

The process of converting the output of a Pandas GroupBy operation into a DataFrame format involves using the ".agg()" method to aggregate the grouped data and then using the ".reset_index()" method to reset the index and convert it into a DataFrame. This allows for a structured and organized presentation of the grouped data, making it easier to analyze and manipulate. Additionally, the use of column labels can provide further clarity and understanding of the data. This method is useful when working with large datasets and performing complex calculations on grouped data.

Convert Pandas GroupBy Output to DataFrame

This tutorial explains how to convert the output of a pandas GroupBy into a pandas DataFrame.

Example: Convert Pandas GroupBy Output to DataFrame

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': ,
'position': ,
'points': })

#view DataFrameprint(df)

team position points
```

```
0 A G 5
1 A G 7
2 A F 7
3 A C 10
4 B G 12
5 B F 22
6 B F 15
7 B F 10
```

We can use the following syntax to count the number of players, grouped by team and position:

```
#count number of players, grouped by team and
position
group = df.groupby().size()
```

```
#view output
print(group)
```

```
team position
```

```
A C 1
```

```
F 1
```

```
G 2
```

```
B F 3
```

```
G 1
```

dtype: int64

From the output, we can see the total count of players, grouped by team and position.

However, suppose we want our output to display the team name in each row like this:

team position count

0 A C 1

1 A F 1

2 A G 2

3 B F 3

4 B G 1

To achieve this output, we can simply use `reset_index()` when performing the `GroupBy`:

#count number of players, grouped by team and position

```
df_out = df.groupby().size().reset_index(name='count')
```

#view output

```
print(df_out)
```

team position count

0 A C 1

1 A F 1

2 A G 2

3 B F 3

4 B G 1

The output now appears in the format that we wanted.

Note that the name argument within `reset_index()` specifies the name for the new column produced by `GroupBy`.

We can also confirm that the result is indeed a pandas DataFrame:

```
#display object type of df_out  
type(df_out)
```

```
pandas.core.frame.DataFrame
```

Note: You can find the complete documentation for the `GroupBy` operation in pandas .

Additional Resources