

How can I use the `groupby()` and `transform()` functions in Pandas to efficiently group data and create transformed versions of that data?

Authored by
stats writer

June 28, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the `groupby()` and `transform()` functions in Pandas to efficiently group data and create transformed versions of that data?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155985>

The `groupby()` and `transform()` functions in Pandas are powerful tools that allow users to efficiently group data and create transformed versions of that data. The `groupby()` function allows users to group data based on specific columns, while the `transform()` function allows users to apply functions to each group separately, creating a new version of the data. This is particularly useful for tasks such as calculating group-specific statistics or applying different transformations to different groups of data. By using these functions, users can easily organize and manipulate large datasets, saving time and increasing overall efficiency.

Use `groupby()` and `transform()` Functions in Pandas

You can use the following methods to use the `groupby()` and `transform()` functions together in a pandas `DataFrame`:

Method 1: Use `groupby()` and `transform()` with built-in function

```
df = df.groupby('group_var').transform('mean')
```

Method 2: Use `groupby()` and `transform()` with custom function

```
df = df.groupby('group_var').transform(lambda x: some function)
```

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({'team': ,  
'points': })
```

```
#view DataFrame
```

```
print(df)
```

```
team points
```

```
0 A 30
```

```
1 A 22
```

```
2 A 19
```

```
3 A 14
```

```
4 B 14
```

```
5 B 11
```

```
6 B 20
```

```
7 B 28
```

Example 1: Use `groupby()` and `transform()` with built-in function

The following code shows how to use the `groupby()` and `transform()` functions to add a new column to the DataFrame called `mean_points`:

```
#create new column called mean_points
```

```
df = df.groupby('team').transform('mean')
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points mean_points
```

```
0 A 30 21.25
```

```
1 A 22 21.25
```

```
2 A 19 21.25
```

```
3 A 14 21.25
```

```
4 B 14 18.25
```

```
5 B 11 18.25
```

```
6 B 20 18.25
```

```
7 B 28 18.25
```

The mean points value for players on team A was 21.25 and the mean points value for players on team B was 18.25, so these values were assigned accordingly to each player in a new column.

Note that we could also use another built-in function such as `sum()` to create a new column that shows the sum of points scored for each team:

```
#create new column called sum_points
```

```
df = df.groupby('team').transform('sum')
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points sum_points
```

```
0 A 30 85
```

```
1 A 22 85
```

```
2 A 19 85
```

```
3 A 14 85
```

```
4 B 14 73
```

```
5 B 11 73
```

```
6 B 20 73
```

```
7 B 28 73
```

The sum of points for players on team A was 85 and the sum of points for players on team B was 73, so these values were assigned accordingly to each player in a new column.

Example 2: Use `groupby()` and `transform()` with custom function

The following code shows how to use the `groupby()` and `transform()` functions to create a custom function that calculates the percentage of total points scored by each

player on their respective teams:

```
#create new column called percent_of_points
```

```
df = df.groupby('team').transform(lambda x: x/x.sum())
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points percent_of_points
```

```
0 A 30 0.352941
```

```
1 A 22 0.258824
```

```
2 A 19 0.223529
```

```
3 A 14 0.164706
```

```
4 B 14 0.191781
```

```
5 B 11 0.150685
```

```
6 B 20 0.273973
```

```
7 B 28 0.383562
```

The first player on team A scored 30 out of 85 total points among team A players. Thus, his percentage of total points scored was $30/85 = 0.352941$. The second player on team A scored 22 out of 85 total points among team A players. Thus, his percentage of total points scored was $22/85 = 0.258824$.

And so on.

Note that we can use the `lambda` argument within the `transform()` function to perform any custom calculation that we'd like.

Additional Resources

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

ARABPSYCHOLOGY.COM