

How can I replicate rows in a Pandas DataFrame?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I replicate rows in a Pandas DataFrame?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=154642>

Pandas DataFrame is a powerful tool for data manipulation and analysis. One useful feature of this library is the ability to replicate rows in a DataFrame. This can be achieved by using the "repeat" method, which takes in an integer value as a parameter to specify the number of times each row should be duplicated. This allows for easy and efficient data augmentation, especially when working with smaller datasets. Additionally, the "repeat" method can also be combined with other DataFrame operations to further enhance data replication and manipulation. Overall, the ability to replicate rows in a Pandas DataFrame provides users with a flexible and efficient solution for data manipulation tasks.

Replicate Rows in a Pandas DataFrame

You can use the following basic syntax to replicate each row in a pandas DataFrame a certain number of times:

#replicate each row 3 times

```
df_new = pd.DataFrame(np.repeat(df.values, 3, axis=0))
```

The number in the second argument of the NumPy repeat() function specifies the number of times to replicate each row.

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

Example: Replicate Rows in a Pandas DataFrame

Suppose we have the following pandas DataFrame that contains information about various basketball players:

```
import pandas as pd

#create dataframe
df = pd.DataFrame({'team': ,
'points': ,
'assists': ,
'rebounds': })
```

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

```
team points assists rebounds
0 A 18 5 11
1 B 20 7 8
2 C 19 7 10
3 D 14 9 6
4 E 14 12 6
5 F 11 5 5
```

We can use the following syntax to replicate each row in the DataFrame three times:

```
import numpy as np
```

```
#define new DataFrame as original DataFrame with each
```

row repeated 3 times

```
df_new = pd.DataFrame(np.repeat(df.values, 3, axis=0))
```

#assign column names of original DataFrame to new DataFrame

```
df_new.columns = df.columns#view new DataFrame
```

```
print(df_new)
```

team points assists rebounds

0 A 18 5 11

1 A 18 5 11

2 A 18 5 11

3 B 20 7 8

4 B 20 7 8

5 B 20 7 8

6 C 19 7 10

7 C 19 7 10

8 C 19 7 10

9 D 14 9 6

10 D 14 9 6

11 D 14 9 6

12 E 14 12 6

13 E 14 12 6

14 E 14 12 6

15 F 11 5 5

16 F 11 5 5

17 F 11 5 5

The new DataFrame contains each of the rows from the original DataFrame, replicated three times each.

Notice that the index values have also been reset.

The values in the index now range from 0 to 17.

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

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