How can I rename the rows in a Pandas DataFrame?

Authored by

stats writer

June 25, 2024

Renaming rows in a Pandas DataFrame can be achieved by using the `rename()` function. This function allows users to provide a new index or label for the rows in the DataFrame. The syntax for this function is `df.rename(index=new_index)`, where `df` is the name of the DataFrame and `new_index` is the new index or label for the rows. This process can be useful when working with large datasets and wanting to organize or clarify the row labels. By using the `rename()` function, users can easily change the row names in a Pandas DataFrame to better suit their needs.

Rename the Rows in a Pandas DataFrame

You can use one of the following methods to rename the rows in a pandas DataFrame:

Method 1: Rename Rows Using Values from Existing Column

```
df = df.set_index('some_column', drop=False).rename_axis(None)
```

Method 2: Rename Rows Using Values from Dictionary

```
row_names = {'old_name0':'new_name0',
'old_name1':'new_name1',
'old_name2':'new_name2'}

df = df.rename(index = row_names)
```

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

```
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 22 7 8
2 C 19 7 10
3 D 14 9 6
4 E 14 12 6
5 F 11 9 5
6 G 20 9 9
7 H 28 4 12
```

Example 1: Rename Rows Using Values from Existing Column

Currently the rows of the DataFrame are labeled from 0 to 7.

We can use the following syntax to rename the rows using the values from the team column:

```
import pandas as pd

#rename rows using values in the team column
df = df.set_index('team', drop=False).rename_axis(None)

#view updated DataFrame
print(df)
```

```
  team points assists rebounds
A    A     18       5       11
B    B     22       7        8
C    C     19       7       10
D    D     14       9        6
E    E     14      12        6
F    F     11       9        5
G    G     20       9        9
H    H     28       4       12
```

Notice that the rows are now labeled from A to H, which match the values from the team column.

If you would like to drop the team column from the DataFrame when renaming the rows, you can remove the argument drop=False from the set_index() function:

```
import pandas as pd

#rename rows using values in the team column and drop team column
df = df.set_index('team').rename_axis(None)

#view updated DataFrame
print(df)

points assists rebounds
A 18 5 11
B 22 7 8
C 19 7 10
D 14 9 6
E 14 12 6
F 11 9 5
G 20 9 9
H 28 4 12
```

Notice that the rows are labeled from A to H and the team column has been dropped entirely.

Example 2: Rename Rows Using Values from Dictionary

We could also define a dictionary that specifies the new row labels for the DataFrame:

```
import pandas as pd

#define new row names
row_names = {0:'Zero',
1:'One',
2:'Two',
3:'Three',
4:'Four',
5:'Five',
6:'Six',
7:'Seven'}

#rename values in index using dictionary called row_names
df = df.rename(index = row_names)

#view updated DataFrame
print(df)

team points assists rebounds
Zero A 18 5 11
```

One B 22 7 8

Two C 19 7 10

Three D 14 9 6

Four E 14 12 6

Five F 11 9 5

Six G 20 9 9

Seven H 28 4 12

Notice the row names of the DataFrame now match those that we specified in the dictionary.

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