How to Easily Split a Pandas DataFrame by Column Value

Authored by

stats writer

December 2, 2025

Splitting a Pandas DataFrame by column value is a common task in data manipulation. It can be easily done by using the groupby() function combined with the appropriate column name. This will group the DataFrame by the given column, and then we can apply an aggregate function to each group to obtain the desired result. This is a useful way to manipulate a DataFrame in order to gain insights into the data and structure it for further analysis.

You can use the following basic syntax to split a pandas DataFrame by column value:

```
#define value to split on
x = 20

#define df1 as DataFrame where 'column_name' is >= 20
df1 = df >= x]

#define df2 as DataFrame where 'column_name' is < 20
df2 = df < x]
```

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

Example: Split Pandas DataFrame by Column Value

Suppose we have the following pandas DataFrame:

```
import pandas as pd

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

#view DataFrame
print(df)

team points rebounds
0 A 22 11
1 B 24 8
2 C 19 10
3 D 18 6
4 E 14 6
5 F 29 5
6 G 31 9
```

7 H 16 12

We can use the following code to split the DataFrame into two DataFrames where the first contains the rows where 'points' is greater than or equal to 20 and the second contains the rows where 'points' is less than 20:

```
#define value to split on
x = 20

#define df1 as DataFrame where 'points' is >= 20
df1 = df >= x]

print(df1)
```

team points rebounds
0 A 22 11
1 B 24 8
5 F 29 5
6 G 31 9

```
#define df2 as DataFrame where 'points' is < 20
df2 = df < x]

print(df2)
```

team points rebounds
2 C 19 10
3 D 18 6
4 E 14 6
7 H 16 12

Note that we can also use the **reset_index()** function to reset the index values for each resulting DataFrame:

```
#define value to split on
x = 20

#define df1 as DataFrame where 'points' is >= 20
df1 = df >= x].reset_index(drop=True)

print(df1)
```

team points rebounds
0 A 22 11
1 B 24 8
2 F 29 5
3 G 31 9

#define df2 as DataFrame where 'points' is < 20
df2 = df < x].reset_index(drop=True)

print(df2)

team points rebounds
0 C 19 10
1 D 18 6
2 E 14 6
3 H 16 12

Notice that the index for each resulting DataFrame now starts at 0.

The following tutorials explain how to fix other common errors in Python: