

How can I split a Pandas DataFrame based on a specific column value?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I split a Pandas DataFrame based on a specific column value?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165398>

The process of splitting a Pandas DataFrame based on a specific column value involves dividing the original DataFrame into separate smaller DataFrames based on the unique values in the specified column. This can be achieved by using the "groupby" function in Pandas, which allows for grouping the data by a specific column. The resulting DataFrames will contain only the rows that have the same value in the specified column. This method is useful for organizing and analyzing data that may have distinct categories or groups within a larger dataset.

Pandas: Split DataFrame By Column Value

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 [df[column_name] >= x]

#define df2 as DataFrame where 'column_name' is < 20

df2 = df [df[column_name] < 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 [df['points'] >= 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 [df['points'] < 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 [df['points'] >= 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 [df['points'] < 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.

Additional Resources

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

ARABPSYCHOLOGY.COM