How can I use loc to select multiple columns in Pandas?

Authored by

stats writer

June 25, 2024

LOC is a powerful method in Pandas that allows for the selection of specific rows and columns in a DataFrame. Using the LOC function, multiple columns can be selected by specifying the column names within square brackets after the LOC function. This allows for a more efficient and concise way of selecting multiple columns instead of individually specifying each column. Furthermore, LOC also allows for the selection of rows based on specific conditions, making it a versatile tool for data manipulation and analysis in Pandas.

Pandas: Use loc to Select Multiple Columns

You can use the loc function in pandas to select multiple columns in a DataFrame by label.

Here are the most common ways to do so:

Method 1: Select Multiple Columns by Name

df.loc]

Method 2: Select All Columns in Range

df.loc

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': ,

'assists': ,

'rebounds': })

#view DataFrame

print(df)

team points assists rebounds

0 A 5 11 6

1 A 7 8 7

2 A 7 10 7

3 A 9 6 6

4 B 12 6 10

5 B 9 5 12

6 B 9 9 10

7 B 4 12 9

Example 1: Select Multiple Columns by Name

The following code shows how to use the loc function to select the 'points' and 'rebounds' columns from the DataFrame:

#select points and rebounds columns

df.loc]

points rebounds
0 5 6
1 7 7
2 7 7
3 9 6
4 12 10
5 9 12
6 9 10
7 4 9

Notice that each row from the 'points' and 'rebounds' columns are returned.

Also note that the order you specify the columns in the loc function is the order they'll be returned in.

For example, we could return the 'rebounds' column first and then the 'points' column:

#select rebounds and points columns
df.loc]

rebounds points

0 6 5

1 7 7

2 7 7

3 6 9

4 10 12

5 12 9

6 10 9

7 9 4

Example 2: Select All Columns in Range

The following code shows how to use the loc function to select all columns between the 'points' and 'rebounds' columns in the DataFrame:

```
#select all columns between points and rebounds columns
df.loc
```

points assists rebounds

0 5 11 6

1 7 8 7

2 7 10 7

3 9 6 6

4 12 6 10

5 9 5 12

6 9 9 10

7 4 12 9

Notice that all columns between the 'points' and 'rebounds' columns in the DataFrame are returned.

Note: To select columns by index position, use the function instead.

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

How to Select Rows Based on Column Values in Pandas