

How do you slice columns in a Pandas DataFrame?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How do you slice columns in a Pandas DataFrame?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=154383>

Pandas DataFrame is a popular data manipulation and analysis tool in Python. One of its useful functions is the ability to slice columns, allowing users to select and extract specific data from the DataFrame. This can be achieved by using the bracket notation, where the column names are placed inside the brackets, or by using the `loc` and `iloc` methods. The `loc` method uses labels to select columns, while the `iloc` method uses integer indexing. By using these methods, users can easily slice columns based on their specific criteria and perform further analysis on the extracted data. Overall, slicing columns in a Pandas DataFrame provides a flexible and efficient way to work with large datasets and extract relevant information for analysis.

Slice Columns in Pandas DataFrame (With Examples)

You can use the following methods to slice the columns in a pandas DataFrame:

Method 1: Slice by Specific Column Names

```
df_new = df.loc]
```

Method 2: Slice by Column Names in Range

```
df_new = df.loc
```

Method 3: Slice by Specific Column Index Positions

```
df_new = df.iloc]
```

Method 4: Slice by Column Index Position Range

```
df_new = df.iloc
```

Note the subtle difference between loc and iloc in each of these methods:

loc selects rows and columns with specific labels
iloc selects rows and columns at specific integer positions

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

```
import pandas as pd
```

```
#create DataFrame with six columns
```

```
df = pd.DataFrame({'team': ,  
'points': ,  
'assists': ,  
'rebounds': ,  
'steals': ,  
'blocks': })
```

```
#view DataFrame
```

```
print(df)
```

```
team points assists rebounds steals blocks
```

```
0 A 18 5 11 4 1
```

```
1 B 22 7 8 3 0
2 C 19 7 10 3 0
3 D 14 9 6 2 3
4 E 14 12 6 5 2
5 F 11 9 5 4 2
6 G 20 9 9 3 1
7 H 28 4 12 8 5
```

Example 1: Slice by Specific Column Names

We can use the following syntax to create a new DataFrame that only contains the columns team and rebounds:

```
#slice columns team and rebounds
```

```
df_new = df.loc[
```

```
#view new DataFrame
```

```
print(df_new)
```

```
team rebounds
```

```
0 A 11
```

```
1 B 8
```

```
2 C 10
```

```
3 D 6
```

```
4 E 6
5 F 5
6 G 9
7 H 12
```

Example 2: Slice by Column Names in Range

```
#slice columns between team and rebounds
```

```
df_new = df.loc
```

```
#view new DataFrame
```

```
print(df_new)
```

```
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 3: Slice by Specific Column Index Positions

We can use the following syntax to create a new

DataFrame that only contains the columns in the index positions 0 and 3:

```
#slice columns in index positions 0 and 3
```

```
df_new = df.iloc]
```

```
#view new DataFrame
```

```
print(df_new)
```

```
team rebounds
```

```
0 A 11
```

```
1 B 8
```

```
2 C 10
```

```
3 D 6
```

```
4 E 6
```

```
5 F 5
```

```
6 G 9
```

```
7 H 12
```

Example 4: Slice by Column Index Position Range

We can use the following syntax to create a new DataFrame that only contains the columns in the index position range between 0 and 3:

```
#slice columns in index position range between 0 and 3
```

```
df_new = df.iloc
```

```
#view new DataFrame
```

```
print(df_new)
```

```
team points assists
```

```
0 A 18 5
```

```
1 B 22 7
```

```
2 C 19 7
```

```
3 D 14 9
```

```
4 E 14 12
```

```
5 F 11 9
```

```
6 G 20 9
```

```
7 H 28 4
```

Note: When using an index position range, the last index position in the range will not be included. For example, the rebounds column in index position 3 is not included in the new DataFrame.

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