How can I select columns by index in a Pandas DataFrame?

Authored by

stats writer

July 2, 2024

Pandas is a popular Python library used for data analysis and manipulation. One of its key features is the ability to select columns in a DataFrame by their index. This means that instead of using the column names, which can be prone to human error, you can select columns by their numerical position. This can be done using the "iloc" function, which stands for "integer location". By specifying the desired index numbers, you can easily retrieve specific columns from the DataFrame. This feature is particularly useful when dealing with large datasets where column names may be difficult to remember, or when performing data analysis tasks that require selecting a subset of columns. Overall, selecting columns by index in a Pandas DataFrame provides a convenient and efficient way to access and manipulate data.

Select Columns by Index in a Pandas DataFrame

Often you may want to select the columns of a pandas DataFrame based on their index value.

If you'd like to select columns based on integer indexing, you can use the .iloc function.

If you'd like to select columns based on label indexing, you can use the .loc function.

This tutorial provides an example of how to use each of these functions in practice.

Example 1: Select Columns Based on Integer Indexing

The following code shows how to create a pandas DataFrame and use .iloc to select the column with an index integer value of 3:

```
import pandas as pd

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

#view DataFrame
df

team points assists rebounds
0 A 11 5 11
1 A 7 7 8
2 A 8 7 10
3 B 10 9 6
4 B 13 12 6
5 B 13 9 5

#select column with index position 3
df.iloc

0 11
1 8
2 10
```

3 6

4 6

5 5

Name: rebounds, dtype: int64

We can use similar syntax to select multiple columns:

#select columns with index positions 1 and 3

df.iloc]

points rebounds

0 11 11

1 7 8

2 8 10

3 10 6

4 13 6

5 13 5

Or we could select all columns in a range:

#select columns with index positions in range 0 through 3

df.iloc

team points assists

0 A 11 5

1 A 7 7

2 A 8 7

3 B 10 9

4 B 13 12

5 B 13 9

Example 2: Select Columns Based on Label Indexing

The following code shows how to create a pandas DataFrame and use .loc to select the column with an index label of 'rebounds':

```
import pandas as pd

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

#view DataFrame
df
```

team points assists rebounds

0 A 11 5 11

1 A 7 7 8

2 A 8 7 10

3 B 10 9 6

4 B 13 12 6

5 B 13 9 5

#select column with index label 'rebounds'

df.loc

0 11

1 8

2 10

3 6

4 6

5 5

Name: rebounds, dtype: int64

We can use similar syntax to select multiple columns with different index labels:

#select the columns with index labels 'points' and 'rebounds'

df.loc]

points rebounds

0 11 11

1 7 8

2 8 10

3 10 6

4 13 6

5 13 5

Or we could select all columns in a range:

#select columns with index labels between 'team' and 'assists'

df.loc

team points assists

0 A 11 5

1 A 7 7

2 A 8 7

3 B 10 9

4 B 13 12

5 B 13 9

Related:

Additional Resources

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

How to Get Row Numbers in a Pandas DataFrame

How to Drop the Index Column in a Pandas DataFrame