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

Authored by

stats writer

April 23, 2024

Pandas DataFrame is a popular data analysis tool used in Python programming. It allows users to easily select and manipulate data in a tabular format. One useful feature of Pandas DataFrame is the ability to select rows by index. This means that users can specify the specific rows they want to extract from their DataFrame based on the index or label assigned to each row. This can be achieved by using the iloc function in Pandas, which allows for efficient and flexible row selection. By specifying the desired index values, users can easily retrieve the corresponding rows from their DataFrame, making data analysis and manipulation more streamlined and efficient.

Select Rows by Index in a Pandas DataFrame

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

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

If you'd like to select rows 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 Rows Based on Integer Indexing

The following code shows how to create a pandas DataFrame and use .iloc to select the row with an index integer value of 4:

import pandas as pd

```
import numpy as np

#make this example reproducible
np.random.seed(0)

#create DataFramedf = pd.DataFrame(np.random.rand(6,2), index=range(0,18,3), columns=)#view DataFrame df

A B
0 0.548814 0.715189
3 0.602763 0.544883
6 0.423655 0.645894
9 0.437587 0.891773
12 0.963663 0.383442
15 0.791725 0.528895

#select the 5th row of the DataFrame
df.iloc]

A B
12 0.963663 0.383442
```

We can use similar syntax to select multiple rows:

#select the 3rd, 4th, and 5th rows of the DataFrame
df.iloc]

A B
6 0.423655 0.645894
9 0.437587 0.891773
12 0.963663 0.383442

Or we could select all rows in a range:

#select the 3rd, 4th, and 5th rows of the DataFrame
df.iloc

A B
6 0.423655 0.645894
9 0.437587 0.891773
12 0.963663 0.383442

Example 2: Select Rows Based on Label Indexing

The following code shows how to create a pandas DataFrame and use .loc to select the row with an index label of 3:

import pandas as pd

```
import numpy as np

#make this example reproducible
np.random.seed(0)

#create DataFramedf =
pd.DataFrame(np.random.rand(6,2),
index=range(0,18,3), columns=)
#view DataFrame
df

A B
0 0.548814 0.715189
3 0.602763 0.544883
6 0.423655 0.645894
9 0.437587 0.891773
12 0.963663 0.383442
15 0.791725 0.528895

#select the row with index label '3'
df.loc]

A B
3 0.602763 0.544883
```

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

#select the rows with index labels '3', '6', and '9'
df.loc]

A B
3 0.602763 0.544883
6 0.423655 0.645894
9 0.437587 0.891773

The Difference Between .iloc and .loc

.iloc selects rows based on an integer index. So, if you want to select the 5th row in a DataFrame, you would use df.iloc] since the first row is at index 0, the second row is at index 1, and so on..loc selects rows based on a labeled index. So, if you want to select the row with an index label of 5, you would directly use df.loc].

How to Get Row Numbers in a Pandas DataFrame
How to Drop Rows with NaN Values in Pandas
How to Drop the Index Column in Pandas