What is the difference between loc and iloc in Pandas?

Authored by

stats writer

July 2, 2024

Loc and iloc are two methods used in Pandas, a popular Python library for data manipulation and analysis. These methods are used to retrieve data from a Pandas DataFrame, which is a two-dimensional tabular data structure.

The main difference between loc and iloc is the way they handle the selection of data. Loc is used for label-based indexing, which means it selects data based on the labels (row and column names) of the DataFrame. On the other hand, iloc is used for integer-based indexing, which means it selects data based on the integer index of the rows and columns.

Another key difference is that loc is inclusive of the endpoints, while iloc is exclusive of the endpoints. This means that when using loc, the last row or column specified in the selection will be included, while with iloc, it will be excluded.

In summary, the main difference between loc and iloc is their approach to selecting data from a DataFrame. Loc uses labels, while iloc uses integer indexes. It is important to understand the differences between these methods to effectively manipulate and analyze data in Pandas.

Pandas loc vs. iloc: What's the Difference?

When it comes to selecting rows and columns of a pandas DataFrame, loc and iloc are two commonly used functions.

Here is the subtle difference between the two functions:

loc selects rows and columns with specific labelsiloc selects rows and columns at specific integer positions

The following examples show how to use each function in practice.

Example 1: How to Use loc in Pandas

Suppose we have the following pandas DataFrame:

import pandas as pd

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

#view DataFrame
df

team points assists
A A 5 11
B A 7 8
C A 7 10
D A 9 6
E B 12 6
F B 9 5
G B 9 9
H B 4 12

We can use loc to select specific rows of the DataFrame based on their index labels:

#select rows with index labels 'E' and 'F'
df.loc]

team points assists
E B 12 6
F B 9 5

We can use loc to select specific rows and specific columns of the DataFrame based on their labels:

#select 'E' and 'F' rows and 'team' and 'assists' columns
df.loc,]

team assists
E B 12
F B 9

We can use loc with the : argument to select ranges of rows and columns based on their labels:

#select 'E' and 'F' rows and 'team' and 'assists' columns
df.loc

team points assists
E B 12 6

F B 9 5

G B 9 9

H B 4 12

Example 2: How to Use iloc in Pandas

Suppose we have the following pandas DataFrame:

```
import pandas as pd

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

#view DataFrame
df
```

team points assists

A A 5 11

B A 7 8

C A 7 10

D A 9 6

E B 12 6

F B 9 5

G B 9 9
H B 4 12

We can use iloc to select specific rows of the DataFrame based on their integer position:

#select rows in index positions 4 through 6 (not including 6)
df.iloc

team points assists
E B 12 6
F B 9 5

We can use iloc to select specific rows and specific columns of the DataFrame based on their index positions:

#select rows in range 4 through 6 and columns in range 0 through 2
df.iloc

team assists
E B 12
F B 9

We can use loc with the : argument to select ranges of rows and columns based on their labels:

#select rows from 4 through end of rows and columns up to third column
df.iloc

team points assists
E B 12 6
F B 9 5
G B 9 9
H B 4 12

Additional Resources

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

How to Select Rows Based on Column Values in Pandas