What are four methods for dropping multiple columns in Pandas?

Authored by

stats writer

June 25, 2024

There are four main methods for dropping multiple columns in Pandas, which is a popular data analysis library in Python. The first method is to use the "drop" function, which allows you to specify a list of columns to be dropped. The second method is to use the "iloc" function, which allows you to specify the index positions of the columns to be dropped. The third method is to use the "filter" function, which allows you to drop columns based on certain criteria, such as column names or data types. Finally, the fourth method is to use the "select_dtypes" function, which allows you to drop columns based on their data types. These methods provide flexibility and efficiency in dropping multiple columns from a Pandas dataframe.

Drop Multiple Columns in Pandas (4 Methods)

You can use the following methods to drop multiple columns from a pandas DataFrame:

Method 1: Drop Multiple Columns by Name

df.drop(columns=, inplace=True)

Method 2: Drop Columns in Range by Name

df.drop(columns=df.loc, inplace=True)

Method 3: Drop Multiple Columns by Index

df.drop(columns=df.columns], inplace=True)

Method 4: Drop Columns in Range by Index

df.drop(columns=df.columns, inplace=True)

Note: The argument inplace=True tells pandas to drop the columns in place without reassigning the DataFrame.

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': ,
'steals': })

#view DataFrame
print(df)

team points assists rebounds steals
0 A 18 5 11 4
1 B 22 7 8 5
2 C 19 7 10 10
```

3 D 14 9 6 12

4 E 14 12 6 4

5 F 11 9 5 8

6 G 20 9 9 7

7 H 28 4 12 2

Example 1: Drop Multiple Columns by Name

The following code shows how to drop the points, rebounds, and steals columns by name:

```
#drop multiple columns by name
df.drop(columns=, inplace=True)

#view updated Dataframe
print(df)
```

team assists

0 A 5

1 B 7

2 C 7

3 D 9

4 E 12

5 F 9

6 G 9

7 H 4

Example 2: Drop Columns in Range by Name

The following code shows how to drop each column between the points and rebounds columns by name:

```
#drop columns in range by name
df.drop(columns=df.loc, inplace=True)

#view updated Dataframe
print(df)
```

```
team steals
0 A 4
1 B 5
2 C 10
3 D 12
4 E 4
5 F 8
6 G 7
7 H 2
```

Example 3: Drop Multiple Columns by Index

The following code shows how to drop the columns in

index positions 0, 3 and 4 from the DataFrame:

#drop multiple columns by index
df.drop(columns=df.columns], inplace=True)

#view updated Dataframe
print(df)

points assists
0 18 5
1 22 7
2 19 7
3 14 9
4 14 12
5 11 9
6 20 9
7 28 4

Example 4: Drop Columns in Range by Index

The following code shows how to drop the columns in index positions 0, 3 and 4 from the DataFrame:

#drop columns by index range
df.drop(columns=df.columns, inplace=True)

#view updated Dataframe
print(df)

team steals
0 A 4
1 B 5
2 C 10
3 D 12
4 E 4
5 F 8
6 G 7
7 H 2

Note that the syntax df.columns specifies columns in index positions 1 *up to* 4.

Thus, this syntax drops the columns in index positions 1, 2 and 3.

Note: You can find the complete documentation for the pandas drop() function .

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