How do I select rows without NaN Values in Pandas?

Authored by

stats writer

November 24, 2025

In Pandas, you can use the df.dropna() method to select rows without NaN values. This method will remove all rows that contain any NaN values in any column. You can also use the df.notnull() method to select rows that have no NaN values in the specified columns.

You can use the following methods to select rows without NaN values in pandas:

Method 1: Select Rows without NaN Values in All Columns

df

Method 2: Select Rows without NaN Values in Specific Column

df.isna()]

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

import pandas as pd
import numpy as np

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

#view DataFrame
print(df)

team points assists
0 A NaN 4.0
1 B 12.0 NaN
2 C 15.0 5.0
3 D 25.0 9.0
4 E NaN 12.0
5 F 22.0 14.0
6 G 30.0 10.0

Example 1: Select Rows without NaN Values in All Columns

We can use the following syntax to select rows without NaN values in every column of the

DataFrame:

#create new DataFrame that only contains rows without NaNs
no_nans = df

#view results
print(no_nans)

team points assists
2 C 15.0 5.0
3 D 25.0 9.0
5 F 22.0 14.0
6 G 30.0 10.0

Notice that each row in the resulting DataFrame contains no NaN values in any column.

Example 2: Select Rows without NaN Values in Specific Column

We can use the following syntax to select rows without NaN values in the **points** column of the DataFrame:

#create new DataFrame that only contains rows without NaNs in points column
no_points_nans = df.isna()]

#view results
print(no_points_nans)

team points assists
1 B 12.0 NaN
2 C 15.0 5.0
3 D 25.0 9.0
5 F 22.0 14.0
6 G 30.0 10.0

Notice that each row in the resulting DataFrame contains no NaN values in the **points** column.

There is one row with a NaN value in the **assists** column, but the row is kept in the DataFrame since the value in the **points** column of that row is not NaN.