How can I select rows with NaN values in Pandas, and what are some examples of using this method?

Authored by

stats writer

June 26, 2024

The "select rows with NaN values" method in Pandas allows users to filter and extract rows from a dataset that contain missing or null values. This method is commonly used in data analysis and manipulation tasks to identify and handle missing data. By using this method, users can easily identify and remove rows with missing values or perform further analysis on them. For example, this method can be used to remove incomplete data from a dataset before performing statistical analysis, or to fill in missing values with appropriate estimates. Overall, the "select rows with NaN values" method in Pandas offers a convenient and efficient way to handle missing data in a dataset.

Select Rows with NaN Values in Pandas (With Examples)

You can use the following methods to select rows with NaN values in pandas:

Method 1: Select Rows with NaN Values in Any Column

df.loc

Method 2: Select Rows with NaN Values in Specific Column

df.loc.isnull()]

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': ,
'rebounds': })

#view DataFrame
print(df)
```

Example 1: Select Rows with NaN Values in Any Column

We can use the following syntax to select rows with NaN values in any column of the DataFrame:

```
#create new DataFrame that only contains rows with NaNs in any column
df_nan_rows = df.loc

#view results
print(df_nan_rows)

team points assists rebounds
1 B NaN 7.0 8.0
4 E 14.0 NaN 6.0
```

7 H 28.0 NaN NaN

Notice that each row in the resulting DataFrame contains a NaN value in at least one column.

Example 2: Select Rows with NaN Values in Specific Column

We can use the following syntax to select rows with NaN values in the assists column of the DataFrame:

```
#create new DataFrame that only contains rows with NaNs in assists column
df_assists_nans = df.loc.isnull()]
#view results
print(df_assists_nans)
```

```
team points assists rebounds
4 E 14.0 NaN 6.0
7 H 28.0 NaN NaN
```

Notice that each row in the resulting DataFrame contains a NaN value in the assists column.

There is one row with a NaN value in the points column, but this row is not selected since it doesn't have a NaN

value in the assists column as well.