How can I drop rows with NaN values in Pandas?

Authored by

stats writer

April 17, 2024

To drop rows with NaN (Not a Number) values in Pandas, one can use the dropna() function. This function will remove any rows that contain NaN values from a given dataset. By specifying the axis parameter as 0, the function will drop rows with NaN values, while setting the axis parameter to 1 will drop columns with NaN values. This method is useful for handling missing data and ensuring the accuracy of data analysis.

Drop Rows with NaN Values in Pandas

Often you may be interested in dropping rows that contain NaN values in a pandas DataFrame. Fortunately this is easy to do using the pandas function.

This tutorial shows several examples of how to use this function on the following pandas DataFrame:

import numpy as np
import scipy.stats as stats

#create DataFrame with some NaN values
df = pd.DataFrame({'rating': ,
'points': ,
'assists': ,
'rebounds': })

#view DataFrame
df

rating points assists rebounds

0 NaN NaN 5.0 11

1 85.0 25.0 7.0 8

2 NaN 14.0 7.0 10

3 88.0 16.0 NaN 6

4 94.0 27.0 5.0 6

5 90.0 20.0 7.0 9

6 76.0 12.0 6.0 6

7 75.0 15.0 9.0 10

8 87.0 14.0 9.0 10

9 86.0 19.0 5.0 7

Example 1: Drop Rows with Any NaN Values

We can use the following syntax to drop all rows that have *any* NaN values:

df.dropna()

rating points assists rebounds

1 85.0 25.0 7.0 8

4 94.0 27.0 5.0 6

5 90.0 20.0 7.0 9

6 76.0 12.0 6.0 6

7 75.0 15.0 9.0 10

8 87.0 14.0 9.0 10

9 86.0 19.0 5.0 7

Example 2: Drop Rows with All NaN Values

We can use the following syntax to drop all rows that have *all* NaN values in each column:

```
df.dropna(how='all')
```

```
rating points assists rebounds
0 NaN NaN 5.0 11
1 85.0 25.0 7.0 8
2 NaN 14.0 7.0 10
3 88.0 16.0 NaN 6
4 94.0 27.0 5.0 6
5 90.0 20.0 7.0 9
6 76.0 12.0 6.0 6
7 75.0 15.0 9.0 10
8 87.0 14.0 9.0 10
9 86.0 19.0 5.0 7
```

There were no rows with all NaN values in this particular DataFrame, so none of the rows were dropped.

Example 3: Drop Rows Below a Certain Threshold

We can use the following syntax to drop all rows that don't have a certain *at least* a certain number of non-NaN values:

df.dropna(thresh=3)

rating points assists rebounds
1 85.0 25.0 7.0 8
2 NaN 14.0 7.0 10
3 88.0 16.0 NaN 6
4 94.0 27.0 5.0 6
5 90.0 20.0 7.0 9
6 76.0 12.0 6.0 6
7 75.0 15.0 9.0 10
8 87.0 14.0 9.0 10
9 86.0 19.0 5.0 7

The very first row in the original DataFrame did not have at least 3 non-NaN values, so it was the only row that got dropped.

Example 4: Drop Row with Nan Values in a Specific Column

We can use the following syntax to drop all rows that

have a NaN value in a specific column:

df.dropna(subset=)

rating points assists rebounds
0 NaN NaN 5.0 11
1 85.0 25.0 7.0 8
2 NaN 14.0 7.0 10
4 94.0 27.0 5.0 6
5 90.0 20.0 7.0 9
6 76.0 12.0 6.0 6
7 75.0 15.0 9.0 10
8 87.0 14.0 9.0 10
9 86.0 19.0 5.0 7

Example 5: Reset Index After Dropping Rows with NaNs

We can use the following syntax to reset the index of the DataFrame after dropping the rows with the NaN values:

#drop all rows that have any NaN values
df = df.dropna()

#reset index of DataFrame

df = df.reset_index(drop=True)

#view DataFrame

df

rating points assists rebounds

0 85.0 25.0 7.0 8

1 94.0 27.0 5.0 6

2 90.0 20.0 7.0 9

3 76.0 12.0 6.0 6

4 75.0 15.0 9.0 10

5 87.0 14.0 9.0 10

6 86.0 19.0 5.0 77

You can find the complete documentation for the dropna() function .