

How can I use the dropna() function in Pandas to remove rows with missing values in specific columns?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the dropna() function in Pandas to remove rows with missing values in specific columns?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151589>

The dropna() function in Pandas is a useful tool for removing rows with missing values in specific columns from a dataset. This function allows the user to specify which columns to check for missing values, and then removes any rows that have missing values in those columns. This can be particularly helpful when working with large datasets, as it allows for quick and efficient removal of incomplete data. By using the dropna() function, data analysts and scientists can ensure that their analyses are based on complete and accurate data, leading to more reliable results.

Pandas: Use dropna() with Specific Columns

You can use the dropna() function with the subset argument to drop rows from a pandas DataFrame which contain missing values in specific columns.

Here are the most common ways to use this function in practice:

Method 1: Drop Rows with Missing Values in One Specific Column

```
df.dropna(subset = , inplace=True)
```

Method 2: Drop Rows with Missing Values in One of Several Specific Columns

```
df.dropna(subset = , inplace=True)
```

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)

team points assists rebounds
0 A 18.0 5.0 11.0
1 B NaN NaN 8.0
2 C 19.0 NaN 10.0
3 D 14.0 9.0 6.0
4 E 14.0 12.0 6.0
5 F 11.0 9.0 5.0
6 G 20.0 9.0 9.0
7 H 28.0 4.0 NaN
```

Example 1: Drop Rows with Missing Values in One Specific Column

We can use the following syntax to drop rows with

missing values in the 'assists' column:

#drop rows with missing values in 'assists' column

df.dropna(subset = , inplace=True)

#view updated DataFrame

print(df)

team points assists rebounds

0 A 18.0 5.0 11.0

3 D 14.0 9.0 6.0

4 E 14.0 12.0 6.0

5 F 11.0 9.0 5.0

6 G 20.0 9.0 9.0

7 H 28.0 4.0 NaN

Notice that the two rows with missing values in the 'assists' column have both been removed from the DataFrame.

Also note that the last row in the DataFrame is kept even though it has a missing value because the missing value is not located in the 'assists' column.

Example 2: Drop Rows with Missing Values in One of Several

Specific Columns

We can use the following syntax to drop rows with missing values in the 'points' or 'rebounds' columns:

```
#drop rows with missing values in 'points' or 'rebounds'  
column
```

```
df.dropna(subset = , inplace=True)
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points assists rebounds
```

```
0 A 18.0 5.0 11.0
```

```
2 C 19.0 NaN 10.0
```

```
3 D 14.0 9.0 6.0
```

```
4 E 14.0 12.0 6.0
```

```
5 F 11.0 9.0 5.0
```

```
6 G 20.0 9.0 9.0
```

Notice that the two rows with missing values in the 'points' or 'rebounds' columns have been removed from the DataFrame.

The following tutorials explain how to perform other

common tasks in pandas:

ARABPSYCHOLOGY.COM