

How can I use “Is Not Null” in Pandas?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use “Is Not Null” in Pandas?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=153327>

The "Is Not Null" function in Pandas is a data filtering tool that allows users to exclude null or missing values from a data set. This is useful for cleaning and organizing data, as well as performing statistical analysis. By using the "Is Not Null" function, users can easily identify and remove any irrelevant or incomplete data, ensuring that their analysis is accurate and reliable. This function can be applied to individual columns or entire data frames, making it a versatile tool for various data manipulation tasks.

Use "Is Not Null" in Pandas (With Examples)

You can use the `pandas notnull()` function to test whether or not elements in a pandas DataFrame are null.

If an element is equal to NaN or None, then the function will return False.

Otherwise, the function will return True.

Here are several common ways to use this function in practice:

Method 1: Filter for Rows with No Null Values in Any Column

`df`

Method 2: Filter for Rows with No Null Values in Specific Column

```
df].notnull().all(1)]
```

Method 3: Count Number of Non-Null Values in Each Column

```
df.notnull().sum()
```

Method 4: Count Number of Non-Null Values in Entire DataFrame

```
df.notnull().sum().sum()
```

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 22.0 NaN 8.0
```

```
2 C 19.0 7.0 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 NaN
```

```
7 H NaN NaN 12.0
```

Example 1: Filter for Rows with No Null Values in Any Column

The following code shows how to filter the DataFrame to only show rows with no null values in any column:

```
#filter for rows with no null values in any column
```

```
df
```

```
team points assists rebounds
```

```
0 A 18.0 5.0 11.0
```

```
2 C 19.0 7.0 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

Example 2: Filter for Rows with No Null Values in Specific Column

The following code shows how to filter the DataFrame to only show rows with no null values in the assists column:

```
#filter for rows with no null values in the 'assists' column  
df].notnull().all(1)]
```

```
team points assists rebounds
```

```
0 A 18.0 5.0 11.0
```

```
2 C 19.0 7.0 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 NaN
```

Notice that each of the rows in this filtered DataFrame have no null values in the assists column.

Example 3: Count Number of Non-Null Values in Each Column

The following code shows how to count the number of

non-null values in each column of the DataFrame:

#count number of non-null values in each column

```
df.notnull().sum()
```

team 8

points 7

assists 6

rebounds 7

dtype: int64

From the output we can see:

The team column has 8 non-null values. The points column has 7 non-null values. The assists column has 6 non-null values. The rebounds column has 7 non-null values.

Example 4: Count Number of Non-Null Values in Entire DataFrame

The following code shows how to count the number of non-null values in the entire DataFrame:

#count number of non-null values in entire DataFrame

```
df.notnull().sum().sum()
```

28

From the output we can see there are 28 non-null values in the entire DataFrame.

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

ARABPSYCHOLOGY.COM