How can I replace all zero values in a Pandas dataframe with NaN?

Authored by

stats writer

June 26, 2024

The process of replacing all zero values in a Pandas dataframe with NaN involves using the replace() function and specifying the value to be replaced as 0 and the replacement value as NaN. This allows for the conversion of all zero values in the dataframe to NaN, which is a more suitable representation for missing or invalid values. This method can be used to clean and manipulate data in a dataframe, ensuring more accurate and consistent analysis.

Pandas: Replace Zero with NaN

You can use the following basic syntax to replace zeros with NaN values in a pandas DataFrame:

df.replace(0, np.nan, inplace=True)

The following example shows how to use this syntax in practice.

Example: Replace Zero with NaN in Pandas

Suppose we have the following pandas DataFrame:

import pandas as pd

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

#view DataFrame

print(df)

points assists rebounds
0 25 5 11
1 0 0 8
2 15 7 10
3 14 0 6
4 19 12 6
5 23 9 0
6 25 9 9
7 29 4 0

We can use the following syntax to replace each zero in the DataFrame with a NaN value:

```
import numpy as np

#replace all zeros with NaN values
df.replace(0, np.nan, inplace=True)

#view updated DataFrame
print(df)
```

points assists rebounds
0 25.0 5.0 11.0

1 NaN NaN 8.0

2 15.0 7.0 10.0

3 14.0 NaN 6.0

4 19.0 12.0 6.0

5 23.0 9.0 NaN

6 25.0 9.0 9.0

7 29.0 4.0 NaN

Notice that each zero in every column of the DataFrame has been replaced with NaN.

Note: We must use the argument inplace=True or else the changes won't be made to the original DataFrame.

Related:

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