

How can I replace NaN values with None in Pandas?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I replace NaN values with None in Pandas?*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152971>

Pandas is a popular library in Python used for data manipulation and analysis. In many datasets, there are missing values represented as NaN (Not a Number). To handle such missing values, a common approach is to replace them with None, a default Python object representing missing data. This can be achieved in Pandas by using the `.fillna()` method and specifying the value to be replaced as None. This allows for easier handling of missing values in data analysis tasks.

Pandas: Replace NaN with None

You can use the following basic syntax to replace NaN values with None in a pandas DataFrame:

```
df = df.replace(np.nan, None)
```

This function is particularly useful when you need to export a pandas DataFrame to a database that uses None to represent missing values instead of NaN.

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

Example: Replace NaN with None in Pandas

Suppose we have the following pandas DataFrame:

```
import pandas as pd
import numpy as np

#create DataFrame
```

```
df = pd.DataFrame({'A': ,  
'B': ,  
'C': ,  
'D': })
```

```
#view DataFrame  
print(df)
```

```
A B C D  
0 5.0 NaN 2.0 5.0  
1 6.0 12.0 7.0 NaN  
2 8.0 NaN 6.0 6.0  
3 NaN 10.0 3.0 15.0  
4 4.0 23.0 2.0 1.0  
5 15.0 6.0 4.0 NaN  
6 13.0 4.0 NaN 4.0
```

Notice that there are several NaN values throughout the DataFrame.

To replace each NaN value with None, we can use the following syntax:

```
#replace all NaN values with None  
df = df.replace(np.nan, None)
```

```
#view updated DataFrameprint(df)
```

```
A B C D
0 5.0 None 2.0 5.0
1 6.0 12.0 7.0 None
2 8.0 None 6.0 6.0
3 None 10.0 3.0 15.0
4 4.0 23.0 2.0 1.0
5 15.0 6.0 4.0 None
6 13.0 4.0 None 4.0
```

Notice that each NaN in every column of the DataFrame has been replaced with None.

Note that if you'd like to only replace NaN values with None in one particular column, you can use the following syntax:

```
#replace NaN values with None in column 'B' only
df = df.replace(np.nan, None)
```

```
#view updated DataFrameprint(df)
```

```
A B C D
0 5.0 None 2.0 5.0
```

1 6.0 12.0 7.0 NaN
2 8.0 None 6.0 6.0
3 NaN 10.0 3.0 15.0
4 4.0 23.0 2.0 1.0
5 15.0 6.0 4.0 NaN
6 13.0 4.0 NaN 4.0

Notice that the NaN values have been replaced with None in column 'B' only.

Related:

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