

How to Reset Index After Using dropna() in Pandas?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How to Reset Index After Using dropna() in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=154641>

The dropna() function in Pandas is used to eliminate rows or columns with missing values in a dataframe. This may result in changes to the dataframe's index or the presence of gaps. The reset_index() function can be used to reset the index to the default integer index starting from 0. To drop the current index and replace it with a new one, the drop parameter can be used. Additionally, the set_index() function allows for a specific column to be set as the index of the dataframe. This function is useful for reorganizing the index after using dropna() or other functions that may alter it.

Pandas: Reset Index After Using dropna()

You can use the following basic syntax to reset an index of a pandas DataFrame after using the dropna() function to remove rows with missing values:

```
df = df.dropna().reset_index(drop=True)
```

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

Example: Reset Index in Pandas After Using dropna()

Suppose we have the following pandas DataFrame that contains information about various basketball players:

```
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 7.0 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 NaN 5.0
```

```
6 G 20.0 9.0 NaN
```

```
7 H 28.0 4.0 12.0
```

Now suppose we use the `dropna()` function to drop all rows from the DataFrame that have a missing value in any column:

```
#drop rows with nan values in any column
```

```
df = df.dropna()
```

```
#view updated DataFrame
```

```
print(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

7 H 28.0 4.0 12.0

Notice that the index still contains the original index values for each row.

To reset the index after using the dropna() function, we can use the following syntax:

#drop rows with nan values in any column

df = df.dropna().reset_index(drop=True)

#view updated DataFrame

print(df)

team points assists rebounds

0 A 18.0 5.0 11.0

1 C 19.0 7.0 10.0

2 D 14.0 9.0 6.0

3 E 14.0 12.0 6.0

4 H 28.0 4.0 12.0

Notice that each of the rows with missing values have been removed and the index values have been reset.

The index values now range from 0 to 4.

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

ARABPSYCHOLOGY.COM