

# How can I drop columns with NaN values in Pandas?

Authored by  
**stats writer**

June 25, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I drop columns with NaN values in Pandas?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152354>

The process of dropping columns with NaN (Not a Number) values in Pandas involves identifying and removing any columns in a dataset that contain missing or invalid values. This can be achieved by using the "dropna" function in Pandas, which allows for the removal of columns that have NaN values. This method helps to ensure that the remaining data is accurate and reliable for analysis.

## **Pandas: Drop Columns with NaN Values**

**You can use the following methods to drop columns from a pandas DataFrame with NaN values:**

### **Method 1: Drop Columns with Any NaN Values**

```
df = df.dropna(axis=1)
```

### **Method 2: Drop Columns with All NaN Values**

```
df = df.dropna(axis=1, how='all')
```

### **Method 3: Drop Columns with Minimum Number of NaN Values**

```
df = df.dropna(axis=1, thresh=2)
```

**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': ,
'position': ,
'points': ,
'rebounds': })

#view DataFrame
print(df)

team position points rebounds
0 A NaN 11 NaN
1 A G 28 NaN
2 A F 10 NaN
3 B F 26 NaN
4 B C 6 NaN
5 B G 25 NaN
```

### Example 1: Drop Columns with Any NaN Values

The following code shows how to drop columns with any NaN values:

```
#drop columns with any NaN values
```

```
df = df.dropna(axis=1)
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points
```

```
0 A 11
```

```
1 A 28
```

```
2 A 10
```

```
3 B 26
```

```
4 B 6
```

```
5 B 25
```

Notice that the position and rebounds columns were dropped since they both had at least one NaN value.

Example 2: Drop Columns with All NaN Values

The following code shows how to drop columns with all NaN values:

```
#drop columns with all NaN values
```

```
df = df.dropna(axis=1, how='all')
```

```
#view updated DataFrame
```

```
print(df)
```

**team position points**

**0 A NaN 11**

**1 A G 28**

**2 A F 10**

**3 B F 26**

**4 B C 6**

**5 B G 25**

**Notice that the rebounds column was dropped since it was the only column with all NaN values.**

**Example 3: Drop Columns with Minimum Number of NaN Values**

**The following code shows how to drop columns with at least two NaN values:**

```
#drop columns with at least two NaN values
```

```
df = df.dropna(axis=1, thresh=2)
```

```
#view updated DataFrame
```

```
print(df)
```

**team position points**

**0 A NaN 11**

**1 A G 28**

**2 A F 10**

**3 B F 26**

**4 B C 6**

**5 B G 25**

**Notice that the rebounds column was dropped since it was the only column with at least two NaN values.**

**Note: You can find the complete documentation for the `dropna()` function in pandas .**

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