

How can I create a boolean column in Pandas based on a condition?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I create a boolean column in Pandas based on a condition?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=154033>

Creating a boolean column in Pandas based on a condition can be achieved by using the "np.where()" function. This function takes in three arguments - the condition, the value to be assigned if the condition is true, and the value to be assigned if the condition is false. By using this function, a new column can be created with boolean values based on the specified condition. This allows for efficient filtering and analysis of data in a Pandas DataFrame.

Pandas: Create Boolean Column Based on Condition

You can use the following basic syntax to create a boolean column based on a condition in a pandas DataFrame:

```
df = np.where(df > 15, True, False)
```

This particular syntax creates a new boolean column with two possible values:

True if the value in some_column is greater than 15. False if the value in some_column is less than or equal to 15.

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

Example: Create Boolean Column Based on Condition in Pandas

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

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'team': ,  
'points': })
```

```
#view DataFrame
```

```
print(df)
```

```
team points
```

```
0 A 5
```

```
1 A 17
```

```
2 A 7
```

```
3 A 19
```

```
4 B 12
```

```
5 B 13
```

```
6 B 9
```

```
7 B 24
```

We can use the following code to create a new column called `good_player` that returns `True` if the value in the `points` column is greater than 15 or `False` otherwise:

```
import numpy as np
```

#create new boolean column based on value in points column

```
df = np.where(df > 15, True, False)
```

#view updated DataFrame

```
print(df)
```

```
team points good_player
```

```
0 A 5 False
```

```
1 A 17 True
```

```
2 A 7 False
```

```
3 A 19 True
```

```
4 B 12 False
```

```
5 B 13 False
```

```
6 B 9 False
```

```
7 B 24 True
```

Notice that the new column called good_player only contains two values: True or False.

We can use the dtypes() function to verify that the new good_player column is indeed a boolean column:

```
#display data type of good_player column  
df.dtypes
```

```
dtype('bool')
```

The new `good_player` column is indeed a boolean column.

Also note that you could return numeric values such as 1 and 0 instead of True and False if you'd like:

```
import numpy as np
```

```
#create new boolean column based on value in points  
column
```

```
df = np.where(df > 15, 1, 0)
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points good_player
```

```
0 A 5 0
```

```
1 A 17 1
```

```
2 A 7 0
```

```
3 A 19 1
```

```
4 B 12 0
```

```
5 B 13 0
```

```
6 B 9 0
```

7 B 24 1

The `good_player` column now contains a 1 if the corresponding value in the `points` column is greater than 15.

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

ARABPSYCHOLOGY.COM