

# How can I filter a Pandas DataFrame using boolean columns?

Authored by  
**stats writer**

June 24, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I filter a Pandas DataFrame using boolean columns?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151509>

Filtering a Pandas DataFrame using boolean columns refers to the process of selecting specific rows or columns based on boolean values in a DataFrame. This can be achieved by using logical operators such as "and", "or", and "not" to create boolean expressions and applying them to the DataFrame. The resulting DataFrame will only contain rows or columns that meet the specified conditions, allowing for efficient data manipulation and analysis.

## Filter Pandas DataFrame Using Boolean Columns

You can use the following methods to filter the rows of a pandas DataFrame based on the values in Boolean columns:

### Method 1: Filter DataFrame Based on One Boolean Column

```
#filter for rows where value in 'my_column' is True  
df.loc
```

### Method 2: Filter DataFrame Based on Multiple Boolean Columns

```
#filter for rows where value in 'column1' or 'column2' is  
True  
df.loc
```

The following examples show how to use each method in practice with the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'points': ,
'all_star': ,
'starter': })
```

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

```
team points all_star starter
0 A 18 True False
1 B 20 False True
2 C 25 True True
3 D 40 True True
4 E 34 True False
5 F 32 False False
6 G 19 False False
```

### Example 1: Filter DataFrame Based on One Boolean Column

We can use the following syntax to filter the pandas DataFrame to only contain rows where the value in the all\_star column is True:

```
#filter for rows where 'all_star' is True  
df.loc
```

```
team points all_star starter
```

```
0 A 18 True False
```

```
2 C 25 True True
```

```
3 D 40 True True
```

```
4 E 34 True False
```

Notice that the DataFrame has been filtered to only contain rows where the value in the all\_star column is True.

If you would instead like to filter for rows where all\_star is False, simply type a tilde (~) in front of the column name:

```
#filter for rows where 'all_star' is False  
df.loc
```

```
team points all_star starter
```

```
1 B 20 False True
```

```
5 F 32 False False
```

```
6 G 19 False False
```

Now the DataFrame has been filtered to only contain rows where the value in the `all_star` column is False.

### Example 2: Filter DataFrame Based on Multiple Boolean Columns

We can use the following syntax to filter the pandas DataFrame to only contain rows where the value in the `all_star` column *or* the `starter` column is True:

```
#filter for rows where 'all_star' or 'starter' is True  
df.loc
```

```
team points all_star starter
```

```
0 A 18 True False
```

```
1 B 20 False True
```

```
2 C 25 True True
```

```
3 D 40 True True
```

```
4 E 34 True False
```

If you would like to filter for rows where the value in *both* the `all_star` and `starter` column is True, you can use the `&` operator instead of the `|` operator:

```
#filter for rows where 'all_star' and 'starter' is True  
df.loc
```

**team points all\_star starter**

**2 C 25 True True**

**3 D 40 True True**

**Now that the DataFrame has been filtered to only contain rows where the value in the all\_star and starter column is True.**

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