

How to Write a Case Statement in Pandas (With Example)

Authored by
stats writer

November 30, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Write a Case Statement in Pandas (With Example)*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=102094>

A Case Statement in Pandas is a way to use the Pandas library to create a new column in a dataframe based on a condition. It can be used to apply an expression to multiple columns of data and return a result based on that expression. For example, a case statement can be used to compare the values of two columns and return a third column with a value based on the comparison. It can also be used to create a categorical column from a numerical column by mapping the numerical values to different categories. This is done using the "when" and "otherwise" clauses of the case statement.

A **case statement** is a type of statement that goes through conditions and returns a value when the first condition is met.

The easiest way to implement a case statement in a Pandas DataFrame is by using the NumPy **where()** function, which uses the following basic syntax:

```
df = np.where(df<9, 'value1',  
np.where(df<12, 'value2',  
np.where(df<15, 'value3', 'value4')))
```

This particular function looks at the value in the column called **col2** and returns:

"value1" if the value in col2 is less than 9
"value2" if the value in col2 is less than 12
"value3" if the value in col2 is less than 15
"value4" if none of the previous conditions are true

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

Example: Case Statement in Pandas

Suppose we have the following pandas DataFrame:

```
import pandas as pd  
import numpy as np  
  
#create DataFrame  
df = pd.DataFrame({'player': ,  
'points': })  
  
#view DataFrame  
df
```

```
player points
```

```
0 1 6
1 2 8
2 3 9
3 4 9
4 5 12
5 6 14
6 7 15
7 8 17
8 9 19
9 10 22
```

We can use the following syntax to write a case statement that creates a new column called **class** whose values are determined by the values in the **points** column:

```
#add 'class' column using case-statement logic
```

```
df = np.where(df<9, 'Bad',  
np.where(df<12, 'OK',  
np.where(df<15, 'Good', 'Great')))
```

```
#view updated DataFrame
```

```
df
```

```
player points class
```

```
0 1 6 Bad
1 2 8 Bad
2 3 9 OK
3 4 9 OK
4 5 12 Good
5 6 14 Good
6 7 15 Great
7 8 17 Great
8 9 19 Great
9 10 22 Great
```

The case statement looked at the value in the **points** column and returned:

"Bad" if the value in the points column was less than 9

"OK" if the value in the points column was less than 12

"Good" if the value in the points column was less than 15

"**Great**" if none of the previous conditions are true

Note: You can find the complete documentation for the NumPy **where()** function .

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

ARABPSYCHOLOGY.COM