

# How can I write a formula in Pandas that checks if a value is present in a specific column and returns a specific result if the condition is met?

Authored by  
**stats writer**

June 26, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I write a formula in Pandas that checks if a value is present in a specific column and returns a specific result if the condition is met?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153201>

Pandas is a popular Python library used for data manipulation and analysis. It provides a variety of tools for handling data, including the ability to write formulas for conditional operations. One useful application of this feature is to check if a certain value is present in a specific column of a Pandas dataframe and return a specific result if the condition is met. This can be achieved using the "loc" function, which allows for conditional selection of data based on specific criteria. By specifying the column and the desired value, along with the desired result, the formula can accurately check for the presence of the value and return the specified outcome. This functionality can greatly enhance the efficiency and accuracy of data analysis and manipulation tasks in Pandas.

## **Pandas: Formula for "If Value in Column Then"**

**You can use the following syntax in pandas to assign values to one column based on the values in another column:**

```
df = df.map(lambda x: 'new1' if 'A' in x else 'new2' if 'B' in x else "")
```

**This particular syntax will create a new column called "new" that takes on the following values:**

**new1 if the value in col is equal to A. new2 if the value in col is equal to B. An empty string if the value in col is equal to any other value.**

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

## Example: Using a Formula for "If Value in Column Then" 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 14
```

```
1 A 22
```

```
2 A 25
```

```
3 A 34
```

```
4 B 30
```

```
5 B 12
```

```
6 C 10
```

```
7 C 18
```

Now suppose that we would like to create a new column

called city whose values depend on the corresponding value in the team column.

We can use the following syntax to do so:

```
#create new column called city whose values depend  
on values in team column
```

```
df = df.map(lambda x: 'Atlanta' if 'A' in x else 'Boston' if  
'B' in x else "")
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points city
```

```
0 A 14 Atlanta
```

```
1 A 22 Atlanta
```

```
2 A 25 Atlanta
```

```
3 A 34 Atlanta
```

```
4 B 30 Boston
```

```
5 B 12 Boston
```

```
6 C 10
```

```
7 C 18
```

This particular syntax created a new column called city that takes on the following values:

**Atlanta if the value in team is equal to A. Boston if the value in team is equal to B. An empty string if the value in team is equal to any other value.**

**Note that in this example we used an empty string after the last else statement to simply leave values blank that didn't meet any condition.**

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

ARABPSYCHOLOGY.COM