

How can I replace NaN values in a pivot table with zeros using Pandas?

Authored by
stats writer

June 28, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I replace NaN values in a pivot table with zeros using Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=156429>

Pandas is a popular data analysis tool that offers many useful functions, including the ability to create pivot tables. However, it is not uncommon for pivot tables to contain missing values, represented as NaN (Not a Number). To address this issue, Pandas provides a simple method for replacing these NaN values with zeros. This can be achieved by using the "fillna()" function and specifying the value to be replaced as zero. This process is useful when dealing with large datasets or when performing mathematical operations on the pivot table. By replacing NaN values with zeros, the data becomes more accurate and can be used for further analysis.

Pandas: Replace NaN Values in Pivot Table with Zeros

You can use the fill_value argument in pandas to replace NaN values in a pivot table with zeros instead.

You can use the following basic syntax to do so:

```
pd.pivot_table(df, values='col1', index='col2',  
columns='col3', fill_value=0)
```

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

Example: Replace NaN Values in Pivot Table with Zeros

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

```
import pandas as pd
```

```
#create DataFrame
```

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

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

```
team position points
```

```
0 A G 4
```

```
1 A G 4
```

```
2 A F 6
```

```
3 A C 8
```

```
4 B F 9
```

```
5 B F 5
```

```
6 B F 5
```

```
7 B F 12
```

We can use the following code to create a pivot table in pandas that shows the mean value of points for each team and position in the DataFrame:

```
#create pivot table
```

```
df_pivot = pd.pivot_table(df, values='points',  
index='team', columns='position')
```

```
#view pivot table
```

```
print(df_pivot)
```

```
position C F G
```

```
team
```

```
A 8.0 6.00 4.0
```

```
B NaN 7.75 NaN
```

Notice that there are two NaN values in the pivot table because there are no players who have a position of C or G on team B in the original DataFrame, so both of these positions have NaN values in the pivot table.

To fill in these NaN values with zeros in the pivot table, we can use the `fill_value` argument:

```
#create pivot table with zeros instead of NaN values
```

```
df_pivot = pd.pivot_table(df, values='points',  
index='team', columns='position',  
fill_value=0)
```

```
#view pivot table
```

```
print(df_pivot)
```

```
position C F G
```

team

A 8 6.00 4

B 0 7.75 0

Notice that each of the NaN values from the previous pivot table have been filled with zeros.

Note: You can find the complete documentation for the pandas `pivot_table()` function .

Additional Resources

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