

How can I create a frequency table in Pandas based on multiple columns?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I create a frequency table in Pandas based on multiple columns?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152540>

Creating a frequency table in Pandas based on multiple columns can be achieved by using the "value_counts" function. This function allows for the counting and grouping of data in a specific column or multiple columns, providing a clear breakdown of the frequency of each unique value. By specifying the desired columns, the "value_counts" function will generate a table with the counts for each unique combination of values in the specified columns. This allows for a comprehensive analysis of the data, making it easier to identify patterns and trends. The resulting frequency table can be used to gain insights and make informed decisions based on the data.

Pandas: Create Frequency Table Based on Multiple Columns

You can use the following basic syntax to create a frequency table in pandas based on multiple columns:

```
df.value_counts()
```

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

Example: Create Frequency Table in Pandas Based on Multiple Columns

Suppose we have the following pandas DataFrame that contains information on team name, position, and points scored by 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 24
```

```
1 A G 33
```

```
2 A G 20
```

```
3 A F 15
```

```
4 B G 16
```

```
5 B G 16
```

```
6 B F 29
```

```
7 B F 25
```

We can use the `value_counts()` function to create a frequency table that shows the occurrence of each combination of values in the team and position columns:

```
#count frequency of values in team and position  
columns
```

```
df.value_counts()
```

```
team position
```

```
A G 3
```

```
B F 2
```

```
G 2
```

```
A F 1
```

```
dtype: int64
```

From the results we can see:

There are 3 occurrences of team A and position G
There are 2 occurrences of team B and position F
There are 2 occurrences of team B and position G
There is 1 occurrence of team A and position F

Note that we can use `reset_index()` to return a DataFrame as a result instead:

```
#count frequency of values in team and position  
columns and return DataFrame
```

```
df.value_counts().reset_index()
```

```
team position 0
```

```
0 A G 3
```

1 B F 2

2 B G 2

3 A F 1

We can use the `rename()` function to rename the column that contains the counts:

#get frequency of values in team and position column and rename count column

```
df.value_counts().reset_index().rename(columns={0:'count'})
```

team position count

0 A G 3

1 B F 2

2 B G 2

3 A F 1

The end result is a DataFrame that contains the frequency of each unique combination of values in the team and position columns.

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