# How can I count observations by group in Pandas?

Authored by

**stats writer**

April 23, 2024

Pandas is a data manipulation tool in Python that allows for efficient handling and analysis of large datasets. One useful function in Pandas is the ability to count observations by group. This means that you can easily calculate the number of occurrences for a specific category within a dataset. This can be done by using the "groupby" function, which groups the data by a specified column and then using the "count" function to count the number of rows in each group. This feature is particularly helpful for summarizing and analyzing data based on different categories and can provide valuable insights for data analysis tasks.

## Count Observations by Group in Pandas

## Often you may be interested in counting the number of observations by group in a pandas DataFrame.

## Fortunately this is easy to do using the groupby() and size() functions with the following syntax:

## df.groupby('column_name').size()

## This tutorial explains several examples of how to use this function in practice using the following data frame:

## import numpy as npimport pandas as pd

## #create pandas DataFrame
## df = pd.DataFrame({'team': ,
## 'division':,
## 'rebounds': })

#display DataFrame
print(df)

team division rebounds
0 A E 11
1 A W 8
2 B E 7
3 B E 6
4 B W 6
5 C W 5
6 C E 12

### Example 1: Count by One Variable

The following code shows how to count the total number of observations by team:

#count total observations by variable 'team'
df.groupby('team').size()

team
A 2
B 3
C 2
dtype: int64

From the output we can see that:

Team A has 2 observationsTeam B has 3 observationsTeam C has 2 observations

Note that the previous code produces a Series. In most cases we want to work with a DataFrame, so we can use the reset_index() function to produce a DataFrame instead:

df.groupby('team').size().reset_index(name='obs')

team obs
0 A 2
1 B 3
2 C 2

### Example 2: Count and Sort by One Variable

We can also use the sort_values() function to sort the group counts.

We can specify ascending=False to sort group counts from largest to smallest or ascending=True to sort from smallest to largest:

```
df.groupby('team').size().reset_index(name='obs').sort_values(, ascending=True)
```

```
team obs
0 A 2
2 C 2
1 B 3
```

### Example 3: Count by Multiple Variables

```
#count observations grouped by team and division
df.groupby().size().reset_index(name='obs')
```

```
team division obs
0 A E 1
1 A W 1
2 B E 2
3 B W 1
4 C E 1
5 C W 1
```



**From the output we can see that:**

**1 observation belongs to Team A and division E1 observation belongs to Team A and division W2**

**observations belongs to Team B and division E1**
**observation belongs to Team B and division W1**
**observation belongs to Team C and division E1**
**observation belongs to Team C and division W**

## How to Calculate the Sum of Columns in Pandas

## How to Calculate the Mean of Columns in Pandas

## How to Find the Max Value of Columns in Pandas