How can we use Pandas GroupBy to count the unique values in a dataset?

Authored by

stats writer

July 2, 2024

Pandas GroupBy is a powerful function that allows for efficient and organized data analysis in Python. By using the GroupBy function, we can easily count the unique values in a dataset. This is particularly useful when working with large datasets, where manually counting unique values can be time-consuming and prone to errors. The GroupBy function works by grouping the data based on a specific variable or column and then applying a function, such as "nunique()", which counts the number of unique values within each group. This allows for a quick and accurate way to determine the number of distinct values in a dataset, providing valuable insights for data analysis and decision-making. Overall, Pandas GroupBy offers a convenient and efficient solution for counting unique values in a dataset.

Count Unique Values Using Pandas GroupBy

You can use the following basic syntax to count the number of unique values by group in a pandas DataFrame:

df.groupby('group_column').nunique()

The following examples show how to use this syntax with the following DataFrame:

import pandas as pd

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

#view DataFrame
df

team position points rebounds
0 A G 5 11
1 A G 7 8
2 A G 7 10
3 A F 9 6
4 A F 12 6
5 B G 9 5
6 B G 9 9
7 B F 4 12
8 B F 7 13
9 B F 7 15

Example 1: Group By One Column & Count Unique Values

The following code shows how to count the number of unique values in the 'points' column for each team:

#count number of unique values in 'points' column grouped by 'team' column
df.groupby('team').nunique()

team

A 4
B 3
Name: points, dtype: int64

From the output we can see:

There are 4 unique 'points' values for team A.There are 3 unique 'points' values for team B.

Note that we can also use the unique() function to display each unique 'points' value by team:

#display unique values in 'points' column grouped by 'team'
df.groupby('team').unique()

team
A
B
Name: points, dtype: object

Example 2: Group By Multiple Columns & Count Unique Values

The following code shows how to count the number of unique values in the 'points' column, grouped by team *and* position:

#count number of unique values in 'points' column grouped by 'team' and 'position'
df.groupby().nunique()

team position
A F 2
G 2
B F 2
G 1
Name: points, dtype: int64

From the output we can see:

There are 2 unique 'points' values for players in position 'F' on team A.There are 2 unique 'points' values for players in position 'G' on team A.There are 2 unique 'points' values for players in position 'F' on team B.There is1 unique 'points' value for players in position 'G' on team B.

Once again, we can use the unique() function to display each unique 'points' value by team and position:

#display unique values in 'points' column grouped by 'team' and 'position'

df.groupby().unique()

team position
A F
G
B F
G
Name: points, dtype: object

Additional Resources

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