

How can I group a pandas DataFrame by day, and what is an example of how to do this?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I group a pandas DataFrame by day, and what is an example of how to do this?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153702>

Grouping a pandas DataFrame by day refers to the process of grouping the data in the DataFrame based on the values in the date column, with each group representing data from a specific day. This allows for easier analysis and comparison of data for each day. An example of how to do this is by using the "groupby" function in pandas, specifying the date column as the key for the grouping. This will create separate groups for each unique value in the date column, allowing for further operations and calculations to be performed on the data within each group.

Group by Day in Pandas DataFrame (With Example)

You can use the following basic syntax to group rows by day in a pandas DataFrame:

```
df.groupby(df.your_date_column.dt.day).sum()
```

This particular formula groups the rows by date in your_date_column and calculates the sum of values for the values_column in the DataFrame.

Note that the dt.day() function extracts the day from a date column in pandas.

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

Example: How to Group by Day in Pandas

Suppose we have the following pandas DataFrame that shows the sales made by some company on various

dates:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'date':  
pd.date_range(start='1/1/2020', freq='8h', periods=10),  
'sales': ,  
'returns': })
```

```
#view DataFrame
```

```
print(df)
```

```
date sales returns
```

```
0 2020-01-01 00:00:00 6 0  
1 2020-01-01 08:00:00 8 3  
2 2020-01-01 16:00:00 9 2  
3 2020-01-02 00:00:00 11 2  
4 2020-01-02 08:00:00 13 1  
5 2020-01-02 16:00:00 8 3  
6 2020-01-03 00:00:00 8 2  
7 2020-01-03 08:00:00 15 4  
8 2020-01-03 16:00:00 22 1  
9 2020-01-04 00:00:00 9 5
```

Related:

We can use the following syntax to calculate the sum of sales grouped by day:

```
#calculate sum of sales grouped by day  
df.groupby(df.date.dt.day).sum()
```

date

1 23

2 32

3 45

4 9

Name: sales, dtype: int64

Here's how to interpret the output:

The total sales made on January 1st was 23. The total sales made on January 2nd was 32. The total sales made on January 3rd was 45. The total sales made on January 4th was 9.

We can use similar syntax to calculate the max of the sales values grouped by month:

```
#calculate max of sales grouped by day
```

```
df.groupby(df.date.dt.day).max()
```

```
date
```

```
1 9
```

```
2 13
```

```
3 22
```

```
4 9
```

```
Name: sales, dtype: int64
```

We can use similar syntax to calculate any value we'd like grouped by the day value of a date column.

Note: You can find the complete documentation for the GroupBy operation in pandas .

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