How can I group a Pandas DataFrame by week, with an example?

Authored by

stats writer

June 28, 2024

Grouping a Pandas DataFrame by week refers to the process of organizing data in a DataFrame by week, where each week is considered as a separate group. This allows for easier analysis and comparison of data within specific time periods. For example, if a DataFrame contains sales data for a business over a span of several months, grouping the data by week will create separate groups for each week, making it easier to analyze and compare sales performance week by week. This can be accomplished using the "groupby()" function in Pandas, where the data is grouped by a specific column containing dates, and then resampled by week. This will result in a new DataFrame with the data grouped by week, allowing for further analysis and visualization.

Group by Week in Pandas DataFrame (With Example)

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

#convert date column to datetime and subtract one week
df = pd.to_datetime(df) - pd.to_timedelta(7, unit='d')

#calculate sum of values, grouped by week
df.groupby().sum()

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

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

Example: How to Group by Week 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='1/5/2022', freq='D', periods=15),
'sales': })

#view DataFrame
print(df)
```

```
date sales
0 2022-01-05 6
1 2022-01-06 8
2 2022-01-07 9
3 2022-01-08 5
4 2022-01-09 4
5 2022-01-10 8
6 2022-01-11 8
7 2022-01-12 3
```

8 2022-01-13 5

9 2022-01-14 9

10 2022-01-15 8

11 2022-01-16 3

12 2022-01-17 4

13 2022-01-18 7

14 2022-01-19 7

Related:

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

```
#convert date column to datetime and subtract one week
df = pd.to_datetime(df) - pd.to_timedelta(7, unit='d')

#calculate sum of values, grouped by week
df.groupby().sum()
```

date

2022-01-02 32

2022-01-09 44

2022-01-16 18

Freq: W-SUN, Name: sales, dtype: int64

Here's how to interpret the output:

There were 32 total sales made during the week starting the day after 1/2/2022.There were 44 total sales made during the week starting the day after 1/9/2022.There were 18 total sales made during the week starting the day after 1/16/2022.

It's worth noting that by default, pandas assumes that the week starts the day after Sunday (W-SUN).

However, according to you can change this value for Freq.

For example, you can specify Freq=W-MON if you'd like each week to start the day after Monday (i.e. Tuesday) instead.

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

```
#convert date column to datetime and subtract one week
df = pd.to_datetime(df) - pd.to_timedelta(7, unit='d')

#calculate max of values, grouped by week
```

df.groupby().max()

date
2022-01-02 9
2022-01-09 9
2022-01-16 7
Freq: W-SUN, Name: sales, dtype: int64

The max sales on an individual day during the week starting the day after 1/2/2022 was 9.The max sales on an individual day during the week starting the day after 1/9/2022 was 9.The max sales on an individual day during the week starting the day after 1/16/2022 was 7.

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

Additional Resources

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