How to Group Data by Hour in Pandas: A Step-by-Step Guide

Authored by

stats writer

November 24, 2025

In Pandas, you can group data by hour using the resample() function. You can specify the desired time period to group the data by, such as hourly, daily, monthly, etc. The resample() function also allows you to specify how to aggregate the data, such as the mean or sum. Additionally, you can also use the groupby() function to group data by hour in Pandas.

You can use the following syntax to group data by hour and perform some aggregation in pandas:

df.groupby(.dt.hour]).sales.sum()

This particular example groups the values by hour in a column called **time** and then calculates the sum of values in the **sales** column for each hour.

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

Example: Group Data by Hour in Pandas

Suppose we have the following pandas DataFrame that shows the number of sales made at various times throughout the day for some store:

import pandas as pd

#create DataFrame
df = pd.DataFrame({'time': ,
'sales': })

#convert date column to datetime
df = pd.to_datetime(df)

#view DataFrame
print(df)

time sales
0 2022-01-01 01:14:00 18
1 2022-01-01 01:24:15 20
2 2022-01-01 02:52:19 15
3 2022-01-01 02:54:00 14
4 2022-01-01 04:05:10 10
5 2022-01-01 05:35:09 9

We can use the following syntax to group the **time** column by hours and calculate the sum of **sales**

for each hour:

#group by hours in time column and calculate sum of sales
df.groupby(.dt.hour]).sales.sum()

time
1 38
2 29
4 10
5 9
Name: sales, dtype: int64

From the output we can see:

A total of **38** sales were made during the first hour.
A total of **29** sales were made during the second hour.
A total of **10**sales were made during the fourth hour.
A total of **9** sales were made during the fifth hour.

Note that we can also perform some other aggregation.

For example, we could calculate the **mean** number of sales per hour:

#group by hours in time column and calculate mean of sales
df.groupby(.dt.hour]).sales.mean()

time
1 19.0
2 14.5
4 10.0
5 9.0
Name: sales, dtype: float64

We can also group by hours and minutes if we'd like.

For example, the following code shows how to calculate the sum of sales, grouped by hours and minutes:

#group by hours and minutes in time column and calculate mean of sales
df.groupby(.dt.hour, df.dt.minute]).sales.mean()

time time

1 14 18
24 20
2 52 15
54 14
4 5 10
5 35 9
Name: sales, dtype: int64

The mean number of sales during 1:14 was **18**.
The mean number of sales during 1:23 was **20**.
The mean number of sales during 2:52 was **15**.

And so on.