

How can I group data by hour in Pandas, using an example?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I group data by hour in Pandas, using an example?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=154581>

Pandas is a popular Python library used for data manipulation and analysis. It offers various functions for grouping and aggregating data, including grouping by hour. This allows us to organize data based on specific time intervals, such as hours, and perform operations on them. For example, if we have a dataset containing sales data from a store, we can group the data by hour to see the total sales made in each hour of the day. This can be achieved by using the 'groupby' function in Pandas, along with the 'hour' attribute. The resulting grouped data can then be further analyzed or visualized for insights. In summary, grouping data by hour in Pandas is a useful technique for organizing and analyzing time-based data.

Group Data by Hour in Pandas (With Example)

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.

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

ARABPSYCHOLOGY.COM