

How can I group data by 5-minute intervals in Pandas?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I group data by 5-minute intervals in Pandas?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153704>

Pandas is a popular data analysis library in Python that allows for efficient manipulation and organization of data. One common task in data analysis is grouping data by time intervals to gain insights and perform further analysis. To group data by 5-minute intervals in Pandas, one can use the built-in function "resample" which allows for resampling of time series data. The "resample" function can be used to group data into various time intervals, including 5-minute intervals. This can be achieved by specifying the "rule" parameter as "5T", which stands for 5 minutes. This will automatically group the data into 5-minute intervals and allow for further analysis and visualization. Overall, Pandas provides a convenient and efficient way to group data by 5-minute intervals, making it a valuable tool for time series analysis.

Group by 5-Minute Intervals in Pandas

You can use the following basic syntax to group rows by 5-minute intervals in a pandas DataFrame:

```
df.resample('5min').sum()
```

This particular formula assumes that the index of your DataFrame contains datetime values and it calculates the sum of every column in the DataFrame, grouped by 5-minute intervals.

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

Related:

Example: How to Group by 5-Minute Intervals in Pandas

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

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'date':  
pd.date_range(start='1/1/2020', freq='min', periods=12),  
'sales': ,  
'returns': })
```

```
#set 'date' column as index
```

```
df = df.set_index('date')
```

```
#view DataFrame
```

```
print(df)
```

```
sales returns
```

```
date
```

```
2020-01-01 00:00:00 6 0
```

```
2020-01-01 00:01:00 8 3
```

```
2020-01-01 00:02:00 9 2
```

```
2020-01-01 00:03:00 11 2
```

```
2020-01-01 00:04:00 13 1
```

```
2020-01-01 00:05:00 8 3
2020-01-01 00:06:00 8 2
2020-01-01 00:07:00 15 4
2020-01-01 00:08:00 22 1
2020-01-01 00:09:00 9 5
2020-01-01 00:10:00 8 3
2020-01-01 00:11:00 4 2
```

Related:

We can use the following syntax to calculate the sum of sales grouped by 5-minute intervals:

```
#calculate sum of sales and returns grouped by 5-
minute intervals
df.resample('5min').sum()
```

```
sales returns
```

```
date
```

```
2020-01-01 00:00:00 47 8
2020-01-01 00:05:00 62 15
2020-01-01 00:10:00 12 5
```

Here's how to interpret the output:

Total sales during minutes 0-4 was 47 and total returns was 8. Total sales during minutes 5-9 was 62 and total returns was 15. Total sales during minutes 10-14 was 12 and total returns was 5.

We can use similar syntax to calculate the max of the sales values and returns values, grouped by 5-minute intervals :

#calculate max of sales and max of returns grouped by 5-minute intervals

```
df.resample('5min').max()
```

sales	returns
-------	---------

date	sales	returns
------	-------	---------

2020-01-01 00:00:00	13	3
---------------------	----	---

2020-01-01 00:05:00	22	5
---------------------	----	---

2020-01-01 00:10:00	8	3
---------------------	---	---

We can use similar syntax to calculate any value we'd like grouped by 5-minute intervals.