

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

Authored by
stats writer

June 28, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I group a Pandas DataFrame by quarter, with an example?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155727>

Grouping a Pandas DataFrame by quarter refers to the process of dividing the data into four equal parts based on the time period of each row. This can be achieved by using the `resample()` function in Pandas, specifying the frequency as 'Q' for quarterly. This will create a new DataFrame with the data grouped by quarter. For instance, if we have a DataFrame with monthly data for a year, grouping it by quarter will result in four new rows, with each row representing the data for that particular quarter. This allows for easier analysis and comparison of data over a longer time period. An example of this can be seen in a sales dataset, where grouping the data by quarter can help in identifying seasonal trends and patterns.

Group by Quarter in Pandas DataFrame (With Example)

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

```
#convert date column to datetime
```

```
df = pd.to_datetime(df)
```

```
#calculate sum of values, grouped by quarter
```

```
df.groupby(df.dt.to_period('Q')).sum()
```

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

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

Example: How to Group by Quarter 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='1/1/2022', freq='M', periods=12),  
'sales': })
```

```
#view DataFrame
```

```
print(df)
```

```
date sales
```

```
0 2022-01-31 6
```

```
1 2022-02-28 8
```

```
2 2022-03-31 10
```

```
3 2022-04-30 5
```

```
4 2022-05-31 4
```

```
5 2022-06-30 8
```

```
6 2022-07-31 8
```

```
7 2022-08-31 3
```

```
8 2022-09-30 5
```

```
9 2022-10-31 14
```

10 2022-11-30 8

11 2022-12-31 3

Related:

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

#convert date column to datetime and subtract one week

```
df = pd.to_datetime(df)
```

#calculate sum of sales, grouped by quarter

```
df.groupby(df.dt.to_period('Q')).sum()
```

date

2022Q1 24

2022Q2 17

2022Q3 16

2022Q4 25

Freq: Q-DEC, Name: sales, dtype: int64

Here's how to interpret the output:

There were 24 total sales made during the first

quarter. There were 17 total sales made during the second quarter. There were 16 total sales made during the third quarter. There were 25 total sales made during the fourth quarter.

We can use similar syntax to calculate some other metric, grouped by quarter.

For example, we could instead calculate the max value of sales, grouped by quarter:

```
#convert date column to datetime
```

```
df = pd.to_datetime(df)
```

```
#calculate max of sales, grouped by quarter
```

```
df.groupby(df.dt.to_period('Q')).max()
```

```
date
```

```
2022Q1 10
```

```
2022Q2 8
```

```
2022Q3 8
```

```
2022Q4 14
```

```
Freq: Q-DEC, Name: sales, dtype: int64
```

Here's how to interpret the output:

The max sales on an individual month during the first quarter was 10. The max sales on an individual month during the second quarter was 8. The max sales on an individual month during the third quarter was 8. The max sales on an individual month during the fourth quarter was 14.

Additional Resources

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