How can I calculate the lag by group in Pandas?

Authored by

stats writer

June 27, 2024

The process of calculating the lag by group in Pandas involves using the built-in functions and methods provided by the Pandas library to group and manipulate data. This allows for the calculation of lag, which is the difference between a current value and a previous value in a specific group. By utilizing Pandas, one can easily calculate the lag by group, allowing for efficient data analysis and interpretation.

Calculate Lag by Group in Pandas

You can use the following methods to calculate lagged values by group in a pandas DataFrame:

Method 1: Calculate Lag by One Group

df = df.groupby().shift(1)

Method 2: Calculate Lag by Multiple Groups

df = df.groupby().shift(1)

Note that the value in the shift() function indicates the number of values to calculate the lag for.

The following examples show how to use each method in practice.

Example 1: Calculate Lag by One Group

Suppose we have the following pandas DataFrame that

shows the sales made by two stores on consecutive days:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'store': ,
'sales': })
#view DataFrame
print(df)
```

```
store sales
0 A 18
1 A 10
2 A 14
3 A 13
4 B 19
5 B 24
6 B 25
7 B 29
```

We can use the following syntax to create a lag column that displays the sales for the previous day for each store:

```
#add column that displays lag of sales column by store
df = df.groupby().shift(1)

#view updated DataFrame
print(df)

store sales lagged_sales
0 A 18 NaN
1 A 10 18.0
2 A 14 10.0
3 A 13 14.0
4 B 19 NaN
5 B 24 19.0
6 B 25 24.0
7 B 29 25.0
```

Here's how to interpret the output:

The first value in the lag column is NaN since there is no prior value in the sales column for store A.The second value in the lag column is 18 since this is the prior value in the sales column for store A.

And so on.

Example 2: Calculate Lag by Multiple Groups

Suppose we have the following pandas DataFrame that shows the sales made by employees at two stores on consecutive days:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'store': ,
'employee':,
'sales': })
#view DataFrame
print(df)
```

```
store employee sales
0 A O 18
1 A O 10
2 A R 14
3 A R 13
4 B O 19
5 B O 24
6 B R 25
7 B R 29
```

We can use the following syntax to create a lag column that displays the sales for the previous day for each employee at each store:

```
#add column that displays lag of sales column by store and employee
df = df.groupby().shift(1)

#view updated DataFrame
print(df)
```

```
store employee sales lagged_sales
0 A O 18 NaN
1 A O 10 18.0
2 A R 14 NaN
3 A R 13 14.0
4 B O 19 NaN
5 B O 24 19.0
6 B R 25 NaN
7 B R 29 25.0
```

The new lagged_sales column displays the sales for the previous day for each employee at each store.

Note: In this example we grouped by two columns, but

you can group by as many columns as you'd like by including as many variable names as you'd like in the groupby() function.

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