

# 6 Lesser-Known Pandas Aggregate Functions

...That make the groupby function even more useful.

[Soner Yildirim](#)



Photo by [Baher Khairy](#) on [Unsplash](#)

The groupby is one of the most frequently used Pandas functions for data analysis. It first divides the data points (i.e. rows in a data frame) into groups based on the distinct values in a column. Then, it calculates aggregated values for each group.

Consider we have a dataset that contains brands and

prices of cars. In order to calculate the average price for each branch, we group the rows based on the brand column and then apply the mean function on the price column.

Pandas provides several aggregate functions that can be used along with the groupby function such as mean, min, max, sum, and so on. In this article, we will see some of the lesser-known aggregate functions that make the groupby function even more useful.

The functions we will cover are:

- first
- last
- nth
- nunique
- describe
- quantile

Let's start with creating a sample data frame.

```
import numpy as np
import pandas as pd
df = pd.DataFrame({
    "Brand": ["Ford", "Honda", "Toyota", "Seat"] * 2,
    "Price": np.random.randint(10000, 30000, size=4)
})
df.head()
```

	Brand	Price
0	Ford	24412
1	Honda	18097
2	Toyota	24247
3	Seat	21503
4	Ford	15715

(image by author)

We have a data frame that contains the price and brand information of 100 cars.

## 1. First

The first function, as its name suggests, returns the first value for each group.

```
df.groupby("Brand", as_index=False).first()
```

	Brand	Price
0	Ford	24412
1	Honda	18097
2	Seat	21503
3	Toyota	24247

(image by author)

## 2. Last

The last function returns the last value for each group.

```
df.groupby("Brand", as_index=False).last()
```

	Brand	Price
0	Ford	28348
1	Honda	22114
2	Seat	23510
3	Toyota	20862

(image by author)

The first and last functions may not seem very useful for this dataset. However, there will be cases where you need a simple solution to find the first or last entry for each group. When you work with date or time-based data, the order matters even more.

### 3. Nth

The nth function extends the capabilities of the first and last functions. It allows for getting the nth row for each group.

```
df.groupby("Brand", as_index=False).nth(2)
```

	Brand	Price
8	Ford	24560
9	Honda	24676
10	Toyota	18012
11	Seat	19234

(image by author)

- `nth(0)` is the same as `first()`
- `nth(-1)` is the same as `last()`

## 4. Nunique

The `nunique` function returns the number of distinct values for each group. It will probably be 25 for each brand in our dataset because we generated 25 random integers in a large range.

When working with real life datasets, the unique values per category or group might be a valuable insight.

```
df.groupby("Brand", as_index=False).nunique()
```

	Brand	Price
0	Ford	25
1	Honda	25
2	Seat	25
3	Toyota	25

(image by author)

## 5. Describe

The describe function returns several statistics for each group. It is usually used to get an overview about the entire data frame. We can also use it with the groupby function to compare the groups from a few different perspectives.

```
df.groupby("Brand", as_index=False).describe()
```

	Price							
	count	mean	std	min	25%	50%	75%	max
0	25.0	19492.36	5969.829540	10140.0	14192.0	20542.0	24412.0	28688.0
1	25.0	19767.08	5126.522131	10353.0	15763.0	18818.0	23953.0	28912.0
2	25.0	19959.92	6115.233008	10004.0	17443.0	20801.0	24024.0	29677.0
3	25.0	20329.40	5889.464718	11118.0	15871.0	19751.0	25112.0	29951.0

(image by author)

The 25%, 50%, and 75% values are the first, second, and third quartiles, respectively. Together with the other

statistics, they provide a structured overview of the distribution of values.

The first quantile (25%) means that 25% of values are below this value. Similarly, 50% of values are below the second quantile so the second quantile is the median value.

## 6. Quantile

We get the 25%, 50%, and 75% quantiles with the describe function. The quantile function offers more flexibility because it accepts a parameter.

In order to find the 40% quantile, we pass 0.4 as a parameter to the quantile function.

```
df.groupby("Brand", as_index=False).quantile(0.4)
```

	Brand	Price
0	Ford	16142.8
1	Honda	17359.4
2	Seat	19142.0
3	Toyota	18790.2

(image by author)

## Conclusion

The groupby function is a life saver in exploratory data analysis. The mean, sum, min, and max are the commonly used aggregate functions with the groupby.

The functions we have covered in this article are not so commonly used but there will be cases where they come in quite handy.

Thank you for reading. Please let me know if you have any feedback.