

# How can Pandas be used to group data and create plots?

Authored by  
**stats writer**

July 2, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can Pandas be used to group data and create plots?*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165689>

Pandas is a powerful data analysis library in Python that offers a variety of functions for manipulating and visualizing data. One of the key features of Pandas is its ability to group data based on specific variables or categories. This grouping allows for a more organized and comprehensive analysis of the data. Additionally, Pandas provides various plotting functions that enable users to create visually appealing and informative graphs and charts from the grouped data. These plots can help identify patterns, trends, and relationships within the data, making it easier to interpret and communicate the insights. Overall, the combination of grouping and plotting capabilities in Pandas allows for efficient and effective data analysis, making it a valuable tool for data scientists and analysts.

## **Pandas: Use Groupby and Plot (With Examples)**

**You can use the following methods to perform a groupby and plot with a pandas DataFrame:**

### **Method 1: Group By & Plot Multiple Lines in One Plot**

```
#define index column
```

```
df.set_index('day', inplace=True)
```

```
#group data by product and display sales as line chart
```

```
df.groupby('product').plot(legend=True)
```

### **Method 2: Group By & Plot Lines in Individual Subplots**

```
pd.pivot_table(df.reset_index(),
```

```
index='day', columns='product', values='sales'
```

```
).plot(subplots=True)
```

The following example shows how to use each method in practice with the following pandas DataFrame:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'day': ,  
'product': ,  
'sales': })
```

```
#view DataFrame
```

```
df
```

```
day product sales
```

```
0 1 A 4
```

```
1 2 A 7
```

```
2 3 A 8
```

```
3 4 A 12
```

```
4 5 A 15
```

```
5 1 B 8
```

```
6 2 B 11
```

```
7 3 B 14
```

```
8 4 B 19
```

```
9 5 B 20
```

## Method 1: Group By & Plot Multiple Lines in One Plot

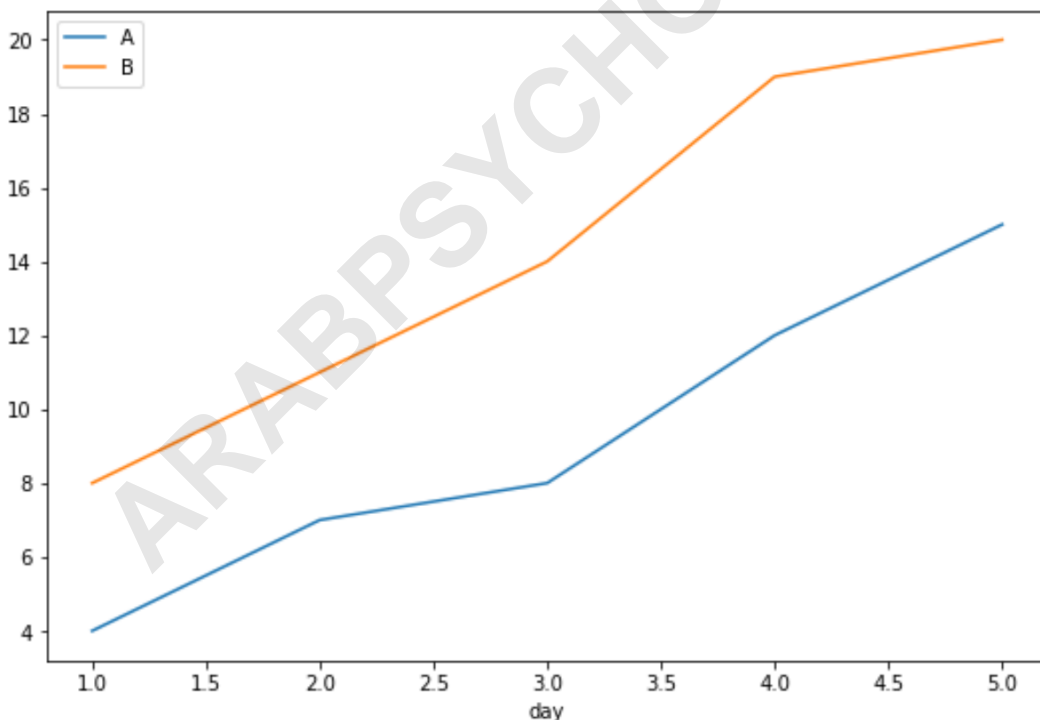
The following code shows how to group the DataFrame by the 'product' variable and plot the 'sales' of each product in one chart:

```
#define index column
```

```
df.set_index('day', inplace=True)
```

```
#group data by product and display sales as line chart
```

```
df.groupby('product').plot(legend=True)
```



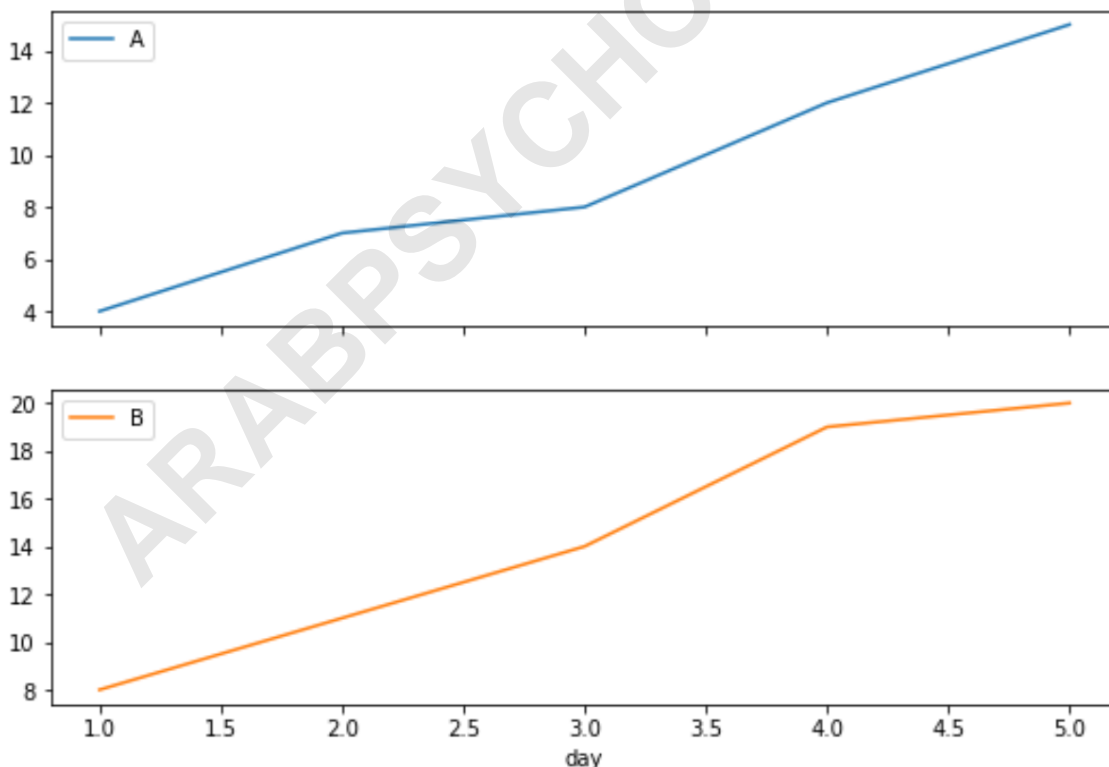
The x-axis displays the day, the y-axis displays the sales, and each individual line displays the sales of the

## individual products.

### Method 2: Group By & Plot Lines in Individual Subplots

The following code shows how to group the DataFrame by the 'product' variable and plot the 'sales' of each product in individual subplots:

```
pd.pivot_table(df.reset_index(),  
index='day', columns='product', values='sales'  
) .plot(subplots=True)
```

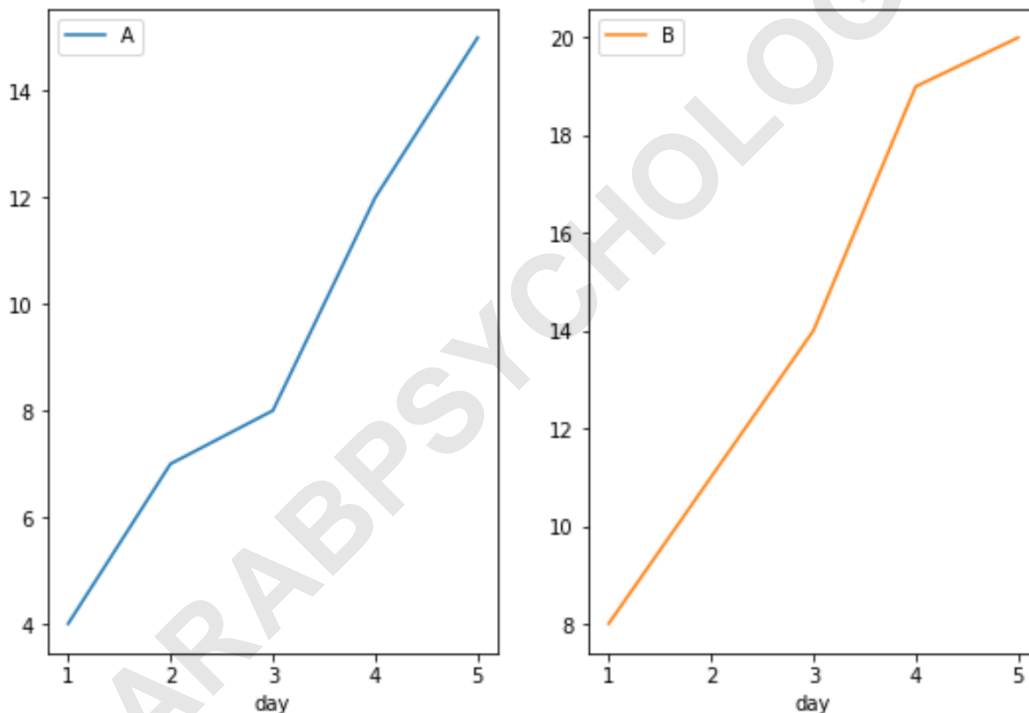


The first plot shows the sales of product A and the

**second plot shows the sales of product B.**

**For example, we could specify the subplots to be in a grid with one row and two columns:**

```
pd.pivot_table(df.reset_index(),  
index='day', columns='product', values='sales'  
).plot(subplots=True, layout=(1,2))
```



### **Additional Resources**

**The following tutorials explain how to create other common visualizations in pandas:**