

# How can I create stacked bar charts in Matplotlib?

Authored by  
**stats writer**

April 19, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I create stacked bar charts in Matplotlib?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=136920>

To create a stacked bar chart in Matplotlib, one must first import the necessary libraries and datasets. Then, using the "bar" function, the data can be plotted onto a graph with each bar representing a different category. By using the "bottom" parameter and setting it to the previous bar's height, the bars can be stacked on top of each other. Additional customization such as labeling and color options can also be applied. The resulting graph will show the total value of each category, as well as the relative contribution of each subcategory to the overall value.

## Create Stacked Bar Charts in Matplotlib (With Examples)

A stacked bar chart is a type of chart that uses bars to display the frequencies of different categories.

We can create this type of chart in Matplotlib by using the `matplotlib.pyplot.bar()` function.

This tutorial shows how to use this function in practice.

### Create a Basic Stacked Bar Chart

The following code shows how to create a stacked bar chart to display the total sales of two products during four different sales quarters:

```
import numpy as np
import matplotlib.pyplot as plt

#create data
quarter =
```

```
product_A =
```

```
product_B =
```

```
#define chart parameters
```

```
N = 4
```

```
barWidth = .5
```

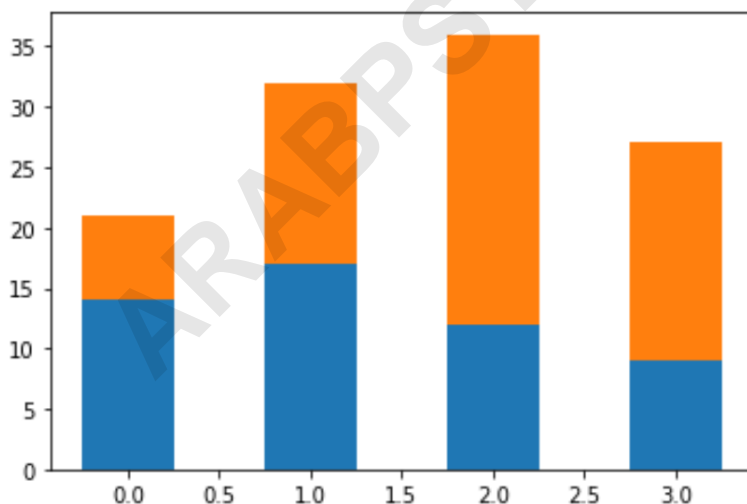
```
xloc = np.arange(N)
```

```
#display stacked bar chart
```

```
p1 = plt.bar(xloc, product_A, width=barWidth)
```

```
p2 = plt.bar(xloc, product_B, bottom=product_A,  
width=barWidth)
```

```
plt.show()
```



**Add a Title, Labels, and Legend**

**We can also add a title, labels, tick marks, and a legend**

to make the chart easier to read:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
#create data for two teams
```

```
quarter =
```

```
product_A =
```

```
product_B =
```

```
#define chart parameters
```

```
N = 4
```

```
barWidth = .5
```

```
xloc = np.arange(N)
```

```
#create stacked bar chart
```

```
p1 = plt.bar(xloc, product_A, width=barWidth)
```

```
p2 = plt.bar(xloc, product_B, bottom=product_A,  
width=barWidth)
```

```
#add labels, title, tick marks, and legend
```

```
plt.ylabel('Sales')
```

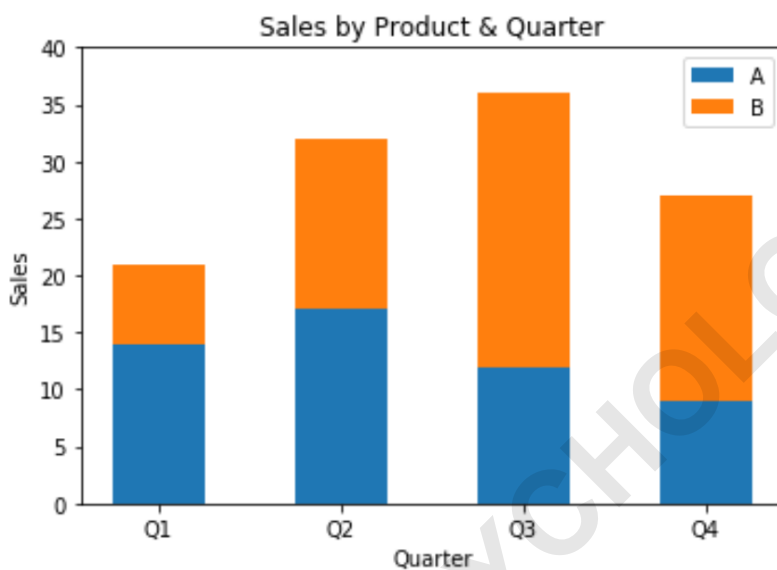
```
plt.xlabel('Quarter')
```

```
plt.title('Sales by Product & Quarter')
```

```
plt.xticks(xloc, ('Q1', 'Q2', 'Q3', 'Q4'))
```

```
plt.yticks(np.arange(0, 41, 5))  
plt.legend((p1, p2), ('A', 'B'))
```

```
#display chart  
plt.show()
```



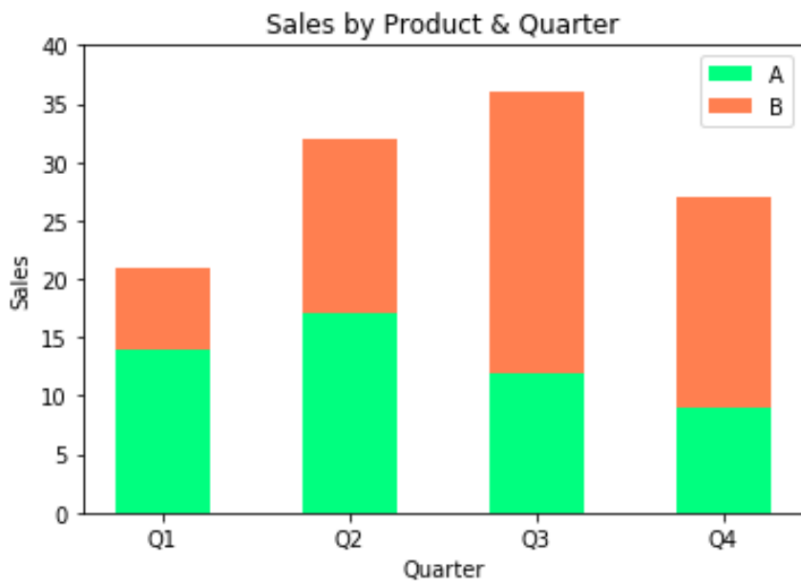
### Customize Chart Colors

Lastly, we can customize the colors used in the chart with the `colors()` argument within `plt.bar()`:

```
import numpy as np  
import matplotlib.pyplot as plt
```

```
#create data for two teams  
quarter =
```

```
product_A =  
product_B =  
  
#define chart parameters  
N = 4  
barWidth = .5  
xloc = np.arange(N)  
  
#create stacked bar chart  
p1 = plt.bar(xloc, product_A, width=barWidth,  
color='springgreen')  
p2 = plt.bar(xloc, product_B, bottom=product_A,  
width=barWidth, color='coral')  
  
#add labels, title, tick marks, and legend  
plt.ylabel('Sales')  
plt.xlabel('Quarter')  
plt.title('Sales by Product & Quarter')  
plt.xticks(xloc, ('Q1', 'Q2', 'Q3', 'Q4'))  
plt.yticks(np.arange(0, 41, 5))  
plt.legend((p1, p2), ('A', 'B'))  
  
#display chart  
plt.show()
```



**You can find a complete list of available colors within the [Matplotlib documentation](#).**

**[How to Change Font Sizes on a Matplotlib Plot](#)**

**[How to Remove Ticks from Matplotlib Plots](#)**

**[How to Show Gridlines on Matplotlib Plots](#)**