

How can I plot multiple lines in Matplotlib?

Authored by
stats writer

April 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I plot multiple lines in Matplotlib?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=138638>

In order to plot multiple lines in Matplotlib, you can use the "plt.plot()" function. This function allows you to specify multiple sets of data to be plotted on the same graph, using different colors and styles for each line. Additionally, you can use the "plt.legend()" function to add a legend to the graph, labeling each line with a corresponding description. By using these functions, you can easily create a clear and visually appealing plot with multiple lines in Matplotlib.

Plot Multiple Lines in Matplotlib

You can display multiple lines in a single Matplotlib plot by using the following syntax:

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

```
plt.plot(df)
```

```
plt.plot(df)
```

```
plt.plot(df)
```

```
...
```

```
plt.show()
```

This tutorial provides several examples of how to plot multiple lines in one chart using the following pandas DataFrame:

```
import numpy as np
```

```
import pandas as pd
```

```
#make this example reproducible
```

```
np.random.seed(0)
```

```
#create dataset
```

```
period = np.arange(1, 101, 1)
```

```
leads = np.random.uniform(1, 50, 100)
```

```
prospects = np.random.uniform(40, 80, 100)
```

```
sales = 60 + 2*period + np.random.normal(loc=0,  
scale=.5*period, size=100)
```

```
df = pd.DataFrame({'period': period,  
'leads': leads,  
'prospects': prospects,  
'sales': sales})
```

```
#view first 10 rows
```

```
df.head(10)
```

```
period leads prospects sales
```

```
0 1 27.891862 67.112661 62.563318
```

```
1 2 36.044279 50.800319 62.920068
```

```
2 3 30.535405 69.407761 64.278797
```

```
3 4 27.699276 78.487542 67.124360
```

```
4 5 21.759085 49.950126 68.754919
```

```
5 6 32.648812 63.046293 77.788596
```

```
6 7 22.441773 63.681677 77.322973
```

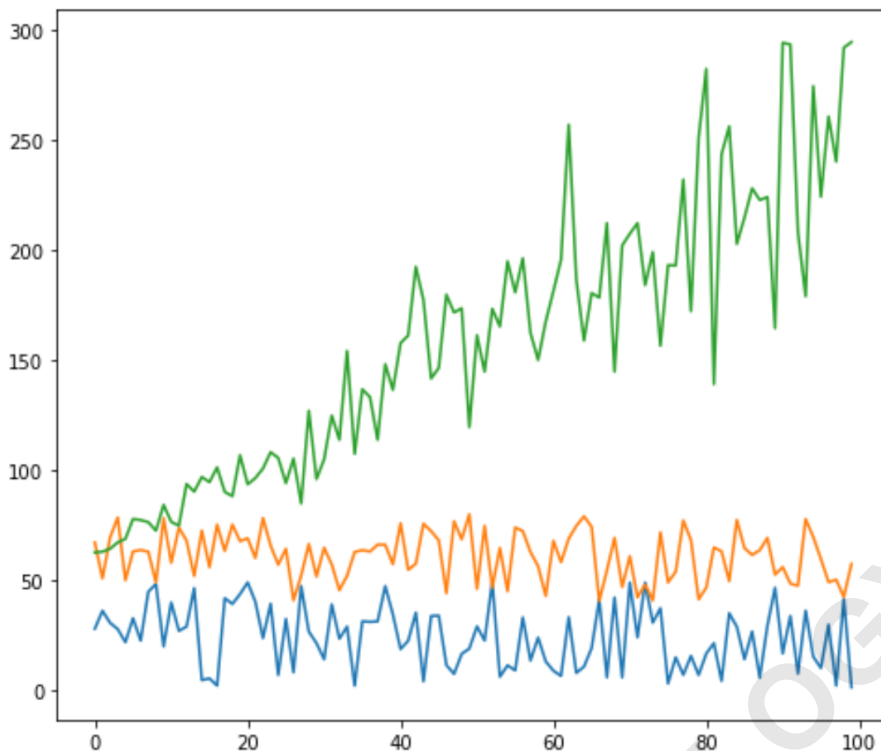
```
7 8 44.696877 62.890076 76.350205
```

```
8 9 48.219475 48.923265 72.485540  
9 10 19.788634 78.109960 84.221815
```

Plot Multiple Lines in Matplotlib

The following code shows how to plot three individual lines in a single plot in matplotlib:

```
import matplotlib.pyplot as plt  
  
#plot individual lines  
plt.plot(df)plt.plot(df)plt.plot(df)  
  
#display plotplt.show()
```



Customize Lines in Matplotlib

You can also customize the color, style, and width of each line:

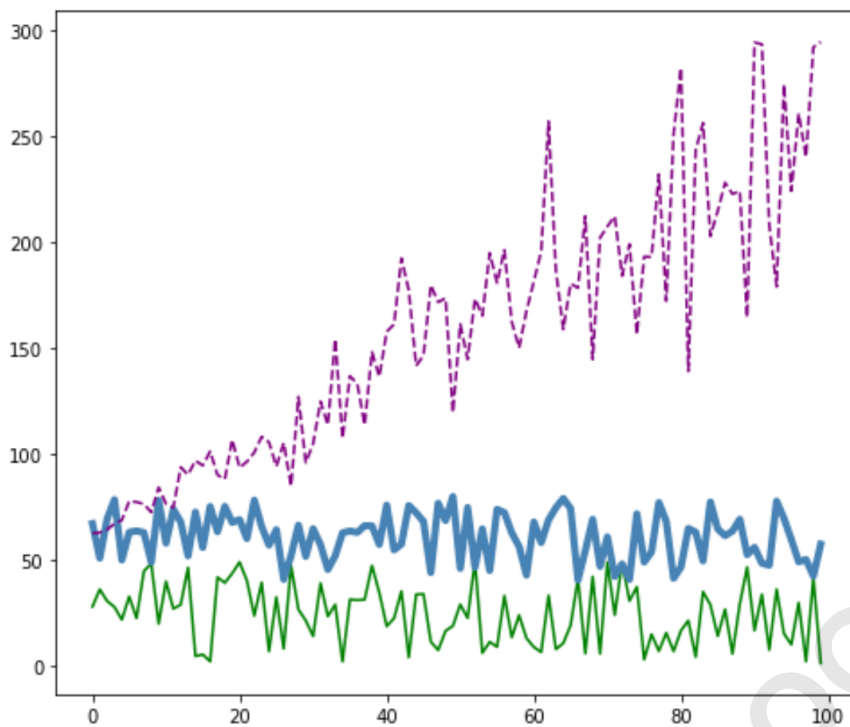
#plot individual lines with custom colors, styles, and widths

```
plt.plot(df, color='green')
```

```
plt.plot(df, color='steelblue', linewidth=4)
```

```
plt.plot(df, color='purple', linestyle='dashed')
```

```
#display plotplt.show()
```



Add a Legend in Matplotlib

You can also add a legend so you can tell the lines apart:

#plot individual lines with custom colors, styles, and widths

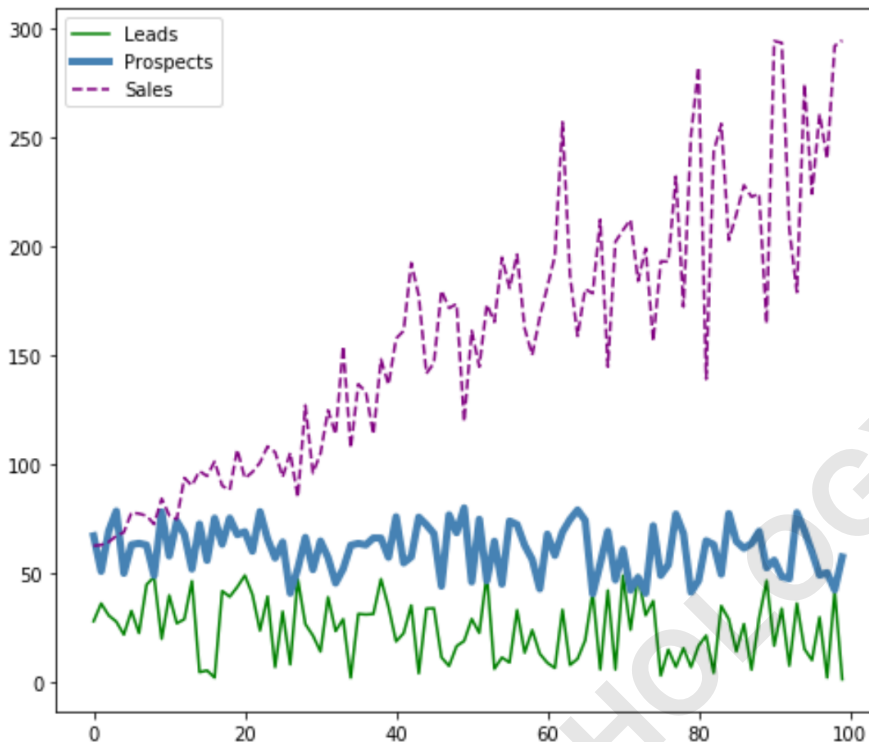
```
plt.plot(df, label='Leads', color='green')
```

```
plt.plot(df, label='Prospects', color='steelblue',  
linewidth=4)
```

```
plt.plot(df, label='Sales', color='purple',  
linestyle='dashed')
```

#add legend

```
plt.legend()#display plotplt.show()
```



Add Axis Labels and Titles in Matplotlib

#plot individual lines with custom colors, styles, and widths

```
plt.plot(df, label='Leads', color='green')
```

```
plt.plot(df, label='Prospects', color='steelblue',  
linewidth=4)
```

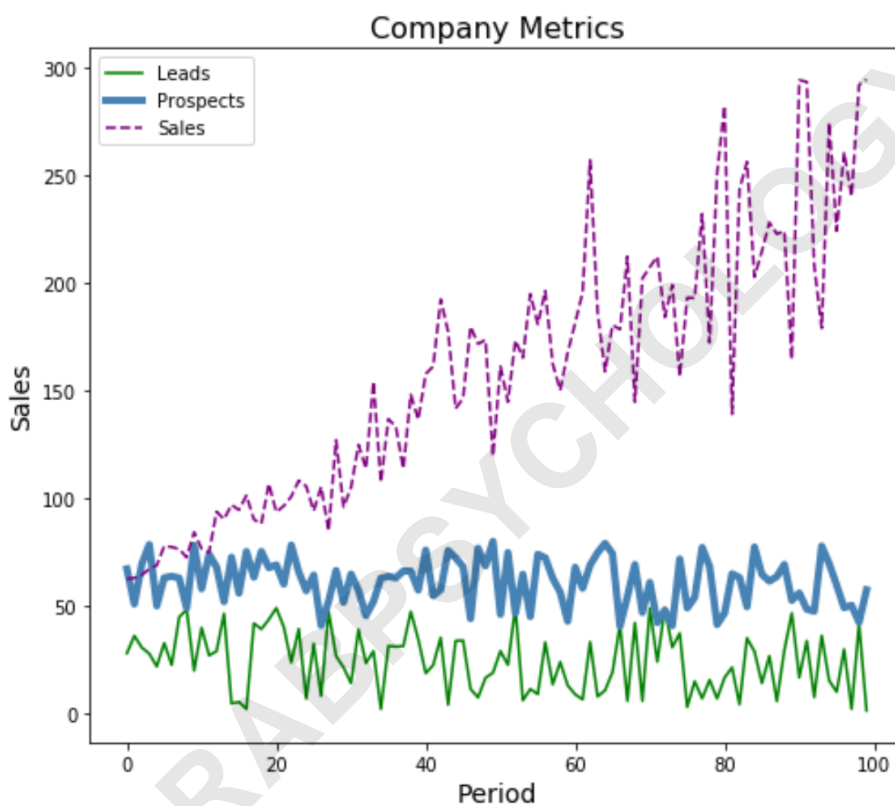
```
plt.plot(df, label='Sales', color='purple',  
linestyle='dashed')
```

#add legend

```
plt.legend()
```

```
#add axis labels and a title
```

```
plt.ylabel('Sales',      fontsize=14)plt.xlabel('Period',  
fontsize=14)plt.title('Company      Metrics',  
fontsize=16)#display plotplt.show()
```



You can find more Matplotlib tutorials .