

Get started

Open in app



Follow

512K Followers

About

You have **1** free member-only story left this month. [Sign up for Medium and get an extra one](#)

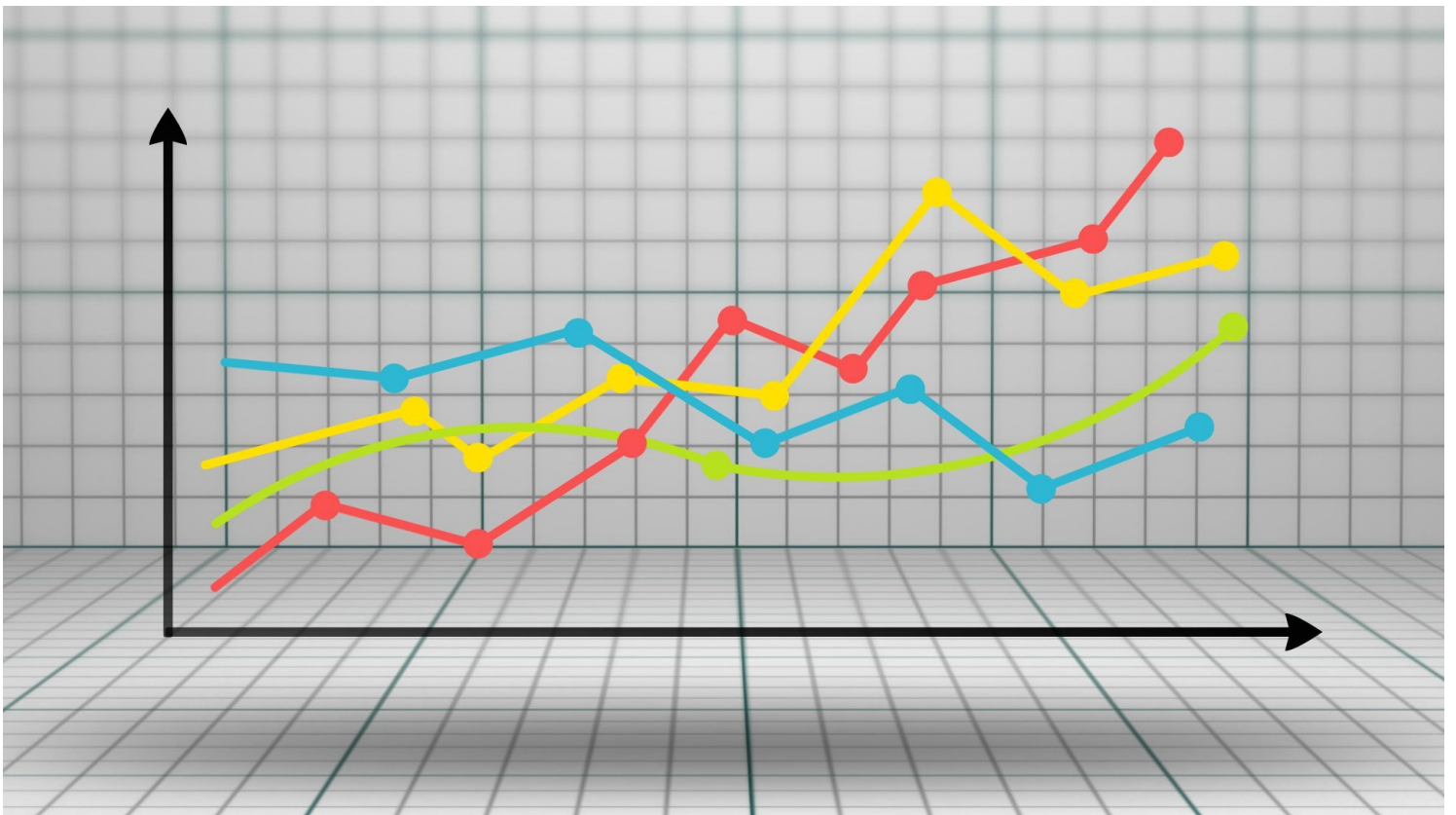


Image by [Mediamodifier](#) on [Pixabay](#)

What Are the “plt” and “ax” in Matplotlib Exactly?

Plotting on a paper, or in a cell on the paper?



Christopher Tao Aug 16 · 5 min read ★



- When should I use “axes”?
- Why some examples using “plt” while someone else using “ax”?
- What’s the difference between them?

It is good that there are so many examples online to show people how to use Matplotlib to draw this kind of chart or that kind of chart, but I rarely see any tutorials mentioning “why”. This may cause people who have less programming experience or switching from other languages like R becomes very confusing.

In this article, I won’t teach you to draw any specific charts using Matplotlib but will try to explain the basic but important regarding Matplotlib — what are the “plt” and “ax” people usually use.

Concepts





“plt”, which I believe you should know, but just in case.

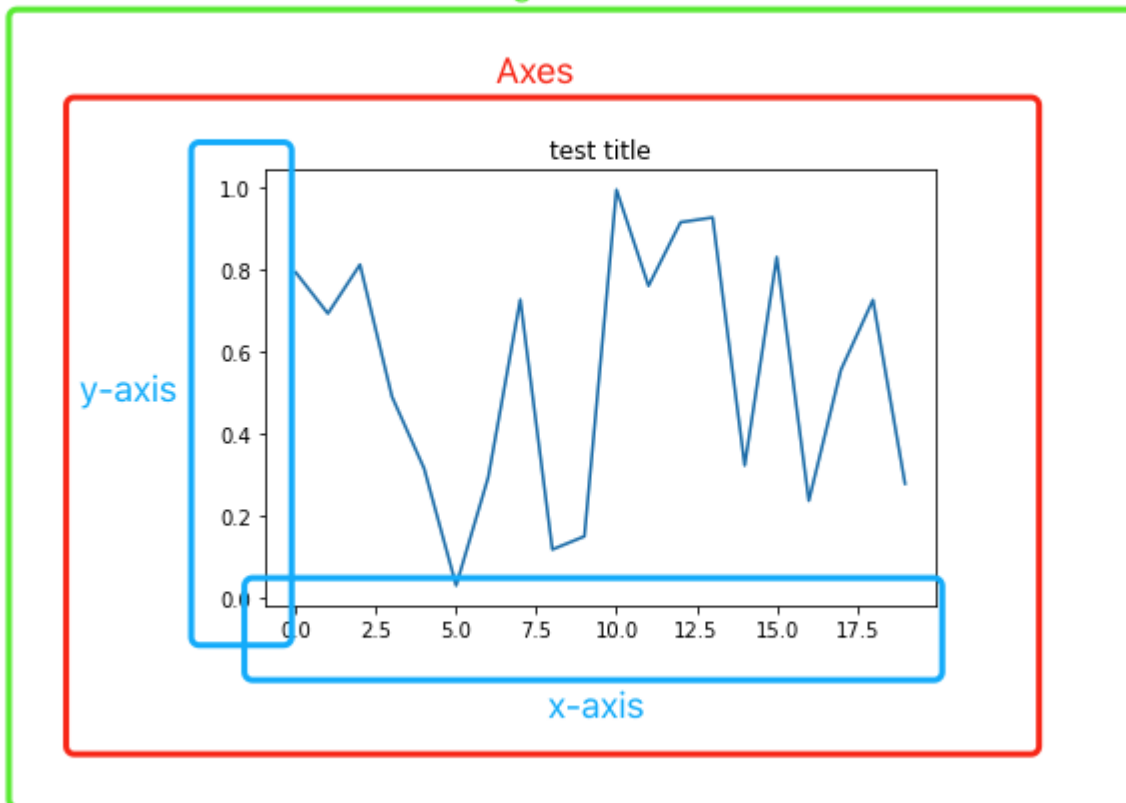
```
import matplotlib.pyplot as plt
```

Then, come back to our main topic. Let’s draw a simple chart for demonstration purposes.

```
import numpy as np
```

```
plt.plot(np.random.rand(20))  
plt.title('test title')  
plt.show()
```

Figure





1. A `Figure` object is generated (shown in green)
2. An `Axes` object is generated **implicitly** with the plotted line chart (shown in red)
3. All the elements of the plot such as x and y-axis are rendered inside the `Axes` object (shown in blue)

Well, if we use some kind of metaphor here:

- `Figure` is like a paper that you can draw anything you want
- We have to draw a chart in a “cell”, which is `Axes` in this context
- If we’re drawing only one graph, we don’t have to draw a “cell” first, just simply draw on the paper anyway. So, we can use `plt.plot(...)`.

Explicitly Draw the “Cell”

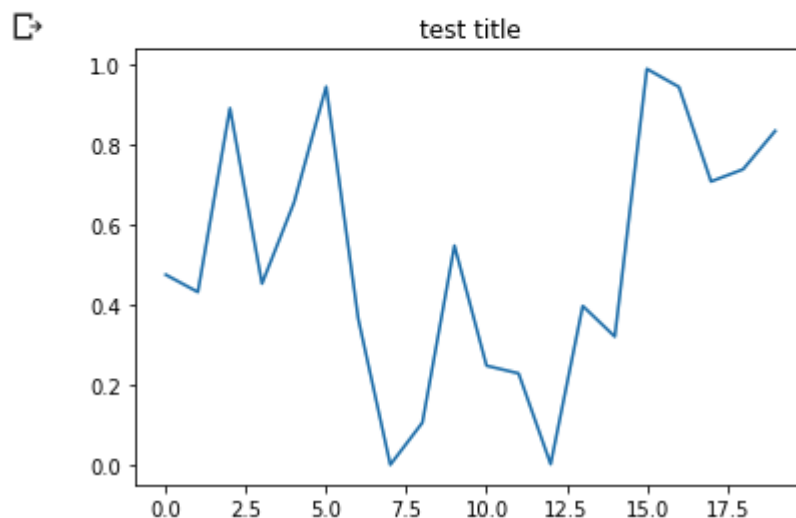




Of course, we can explicitly draw a “cell” on the “paper”, to tell matplotlib that we’re gonna draw a chart inside this cell. Then, we have the following code.

```
fig, ax = plt.subplots()
ax.plot(np.random.rand(20))
ax.set_title('test title')
plt.show()
```

```
[5] fig, ax = plt.subplots()
    ax.plot(np.random.rand(20))
    ax.set_title('test title')
    plt.show()
```



Exactly the same results. The only difference is that we explicitly draw the “cell” so that we are able to get the `Figure` and `Axes` object.

```
[7] type(fig)
```

```
matplotlib.figure.Figure
```

```
[8] type(ax)
```

```
matplotlib.axes._subplots.AxesSubplot
```




graphs in one plot. In other words, the subplots.

```
n_rows = 2
n_cols = 2

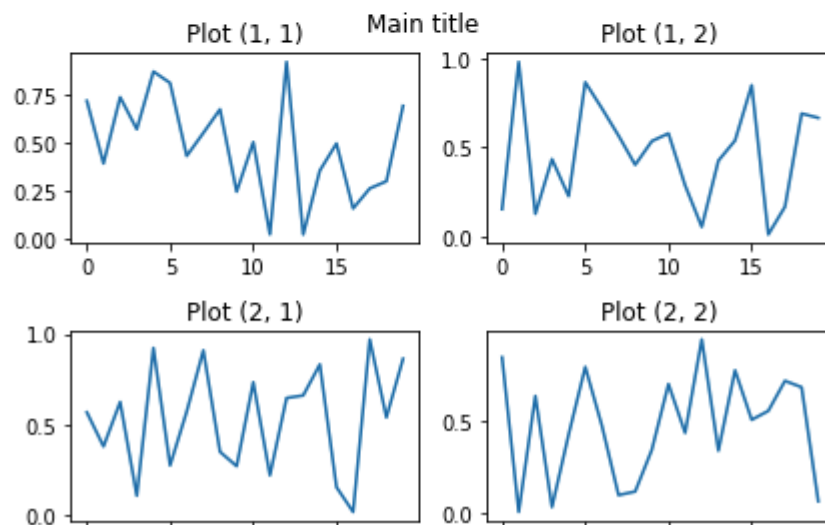
fig, axes = plt.subplots(n_rows, n_cols)
for row_num in range(n_rows):
    for col_num in range(n_cols):
        ax = axes[row_num][col_num]
        ax.plot(np.random.rand(20))
        ax.set_title(f'Plot ({row_num+1}, {col_num+1})')

fig.suptitle('Main title')
fig.tight_layout()
plt.show()
```

```
[16] n_rows = 2
      n_cols = 2

      fig, axes = plt.subplots(n_rows, n_cols)
      for row_num in range(n_rows):
          for col_num in range(n_cols):
              ax = axes[row_num][col_num]
              ax.plot(np.random.rand(20))
              ax.set_title(f'Plot ({row_num+1}, {col_num+1})')

      fig.suptitle('Main title')
      fig.tight_layout()
      plt.show()
```



In this code snippet, we firstly declared how many rows and columns we want to “draw”. 2 by 2 means that we want to draw 4 “cells”.

```
[20] axes
```

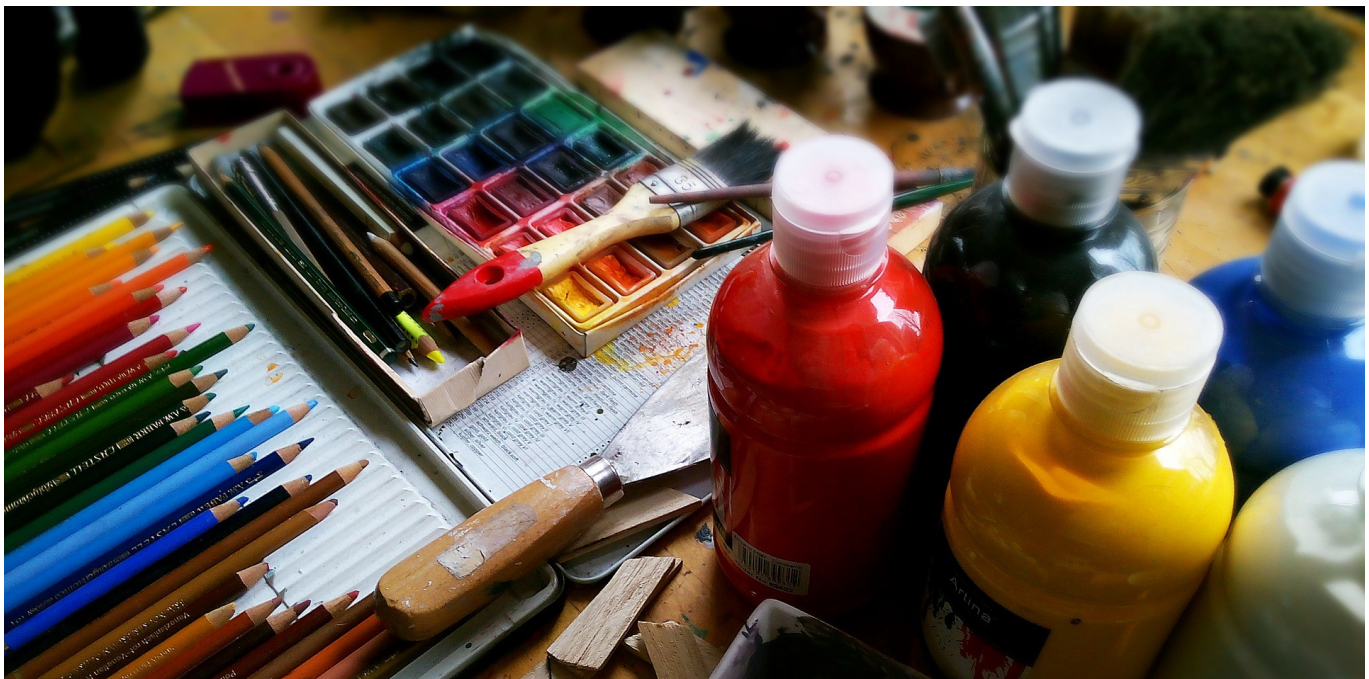
```
[>] array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f9ac63ba320>,  
          <matplotlib.axes._subplots.AxesSubplot object at 0x7f9ac63d6eb8>],  
          [<matplotlib.axes._subplots.AxesSubplot object at 0x7f9ac63920b8>,  
          <matplotlib.axes._subplots.AxesSubplot object at 0x7f9ac63432e8>]],  
          dtype=object)
```

Then, in each cell, we plot a random line chart and assign a title based on its row number and column number. Please note that we’re using `Axes` instances.

After that, we define a “Main title” on the “paper”, which is the `Figure` instance. So, we have this supertitle that does not belong to any “cell”, but on the paper.

Finally, before calling the `show()` method, we need to ask the “paper” — `Figure` instance — to automatically give enough padding between the cells by calling its `tight_layout()` method. Otherwise,

Summary





Hopefully, now you understand better what are `plt` and `ax` people are using exactly.

Basically, the `plt` is a common alias of `matplotlib.pyplot` used by most people. When we plot something using `plt` such as `plt.line(...)`, we implicitly created a `Figure` instance and an `Axes` inside the `Figure` object. This is totally fine and very convenient when we just want to draw a single graph.

However, we can explicitly call `plt.subplots()` to get the `Figure` object and `Axes` object, in order to do more things on them. When we want to draw multiple subplots on a `Figure`, it is usually required to use this approach.

Also, here are the Matplotlib official API reference for the `Figure` and `Axes` classes. It is highly recommended to check them out and try some methods yourselves to make sure you understand even deeper.

matplotlib.axes - Matplotlib 3.3.1 documentation

The instance supports callbacks through a `callbacks` attribute which is a instance. The events you can connect to are...

[matplotlib.org](https://matplotlib.org/3.3.1/api/axes_api.html)

matplotlib.figure.Figure - Matplotlib 3.3.1 documentation

The top level container for all the plot elements. The `Figure` instance supports callbacks through a `callbacks` attribute...

[matplotlib.org](https://matplotlib.org/3.3.1/api/figure_api.html)

Sign up for The Daily Pick

By Towards Data Science

[Get started](#)[Open in app](#)

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

[Data Science](#)[Data Visualization](#)[Python](#)[Matplotlib](#)[Technology](#)[About](#) [Help](#) [Legal](#)

Get the Medium app

