



ChatGPT ▾

Saved memory full ⓘ

Share ...



what is the difference between plt.figure() and plt.subplot()

Let's break this down carefully. Both `plt.figure()` and `plt.subplot()` are part of Matplotlib, but they serve very different purposes.

1. `plt.figure()`

Purpose: Create a new figure (a blank canvas) on which you can draw plots.

- Think of it as creating a new page in your notebook.
- You can have multiple figures open at the same time.
- You can customize the figure size, DPI, background color, etc.

Example:

```
python  Copy code

import matplotlib.pyplot as plt

# Create a new figure
plt.figure(figsize=(6,4))

# Add a plot to this figure
plt.plot([1, 2, 3], [4, 5, 6])
plt.show()
```

Here, `plt.figure()` initializes a new figure for plotting. Without it, Matplotlib will plot on the **current active figure**.

2. `plt.subplot()`

Purpose: Divide a single figure into a grid of subplots and select which subplot to plot on.

- Think of it as dividing a page into multiple sections and choosing where to draw.
- You can have multiple plots on the **same figure** using subplots.

Syntax:

```
python  Copy code

plt.subplot(nrows, ncols, index)
```

- `nrows` : Number of rows in the grid
- `ncols` : Number of columns in the grid
- `index` : Which subplot you want to draw on (counted row-wise)

Example:

python

 Copy code

```
import matplotlib.pyplot as plt

# Divide figure into 2 rows, 1 column
plt.subplot(2, 1, 1) # Top subplot
plt.plot([1, 2, 3], [1, 4, 9])

plt.subplot(2, 1, 2) # Bottom subplot
plt.plot([1, 2, 3], [2, 5, 10])

plt.show()
```

Here, the figure is **split into 2 subplots**, and each `plt.subplot()` specifies which one to draw in.

Key Differences

Feature	<code>plt.figure()</code>	<code>plt.subplot()</code>
Purpose	Create a new figure (canvas)	Divide a figure into subplots and select one
Multiple plots	Use multiple figures for separate plots	Multiple subplots in a single figure
Grid support	No	Yes, organizes multiple plots in rows/columns
Usage	Optional if using default figure	Needed when using multiple subplots

Summary:

- `plt.figure()` → starts a new figure.
 - `plt.subplot()` → divides a figure into sections to plot multiple charts on the same figure.
-

If you want, I can make a **small diagram** showing how `figure` and `subplot` relate visually — it makes this really clear. Do you want me to do that?



then what does `fig, axis = plt.subplots()` do

1. `plt.subplot()` vs `plt.subplots()`

- `plt.subplot()` (singular)
 - Creates **one subplot** in a figure and **activates it** for plotting.
 - Returns an `Axes` object if used like `ax = plt.subplot(...)`.
 - Syntax: `plt.subplot(nrows, ncols, index)`
 - Example:

python

 Copy code

```
import matplotlib.pyplot as plt

ax = plt.subplot(1, 1, 1) # Single subplot
ax.plot([1, 2, 3], [4, 5, 6])
plt.show()
```

- `plt.subplots()` (plural)
 - Creates **a figure and a grid of subplots** at the same time.
 - Returns **two objects**: `(fig, ax)`
 - `fig` → the **Figure object** (the canvas)
 - `ax` → the **Axes object(s)** (the area(s) to plot on)
 - Syntax: `fig, ax = plt.subplots(nrows=1, ncols=1)`

Example:

python

 Copy code

```
import matplotlib.pyplot as plt

# Create a figure with a 2x2 grid of subplots
fig, axes = plt.subplots(2, 2)

axes[0, 0].plot([1,2,3], [1,4,9]) # Top-Left
axes[0, 1].plot([1,2,3], [2,5,10]) # Top-right
axes[1, 0].plot([1,2,3], [3,6,11]) # Bottom-Left
axes[1, 1].plot([1,2,3], [4,7,12]) # Bottom-right

plt.show()
```

Key Points:

- `plt.subplot()` → for **one subplot**, returns **Axes**.
- `plt.subplots()` → for **creating figure + multiple subplots**, returns **(Figure, Axes array)**.

So, when you see:

python

 Copy code

```
fig, ax = plt.subplots()
```

It means:

1. `fig` is your **Figure object** (the overall canvas).
2. `ax` is your **Axes object** (the area where you can plot).

If you want, I can draw a **visual diagram showing `figure`, `subplot`, and `axes` relationship`**, which makes it much easier to understand.

Do you want me to do that?



list the argument and there work of figure, subplot and subplots

Sure! Let's carefully list the **arguments** and their **functions** for `plt.figure()`, `plt.subplot()`, and `plt.subplots()` in Matplotlib. I'll also explain what each does.

1. `plt.figure()`

Purpose: Create a new figure (canvas) for plotting.

Common Arguments:

Argument	Purpose
<code>figsize=(width, height)</code>	Sets the size of the figure in inches (width, height)
<code>dpi</code>	Dots per inch, controls resolution of the figure
<code>facecolor</code>	Background color of the figure
<code>edgecolor</code>	Color of the figure edge
<code>tight_layout</code>	Automatically adjusts subplots to fit into the figure area
<code>num</code>	Figure number or name. If a figure with this number exists, activates it

Example:

```
python
```

Copy code

```
plt.figure(figsize=(8,6), dpi=100, facecolor='lightgrey')
```

2. plt.subplot()

Purpose: Create a single subplot in a figure or select a subplot from a grid.

Syntax:

python

 Copy code

```
plt.subplot(nrows, ncols, index, **kwargs)
```

Arguments:

Argument	Purpose
nrows	Number of rows in the subplot grid
ncols	Number of columns in the subplot grid
index	Index of the subplot (counted row-wise, starting at 1)
projection	'3d' for 3D plots
polar	True for polar plots
**kwargs	Additional keyword arguments (like facecolor, etc.)

Example:

python

 Copy code

```
plt.subplot(2, 2, 1) # Top-Left subplot in a 2x2 grid
```

3. plt.subplots()

Purpose: Create a figure and a grid of subplots at the same time.

Syntax:

python

 Copy code

```
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(width, height), dpi=100, sharex
```

Arguments:

Argument	Purpose
nrows	Number of rows of subplots
ncols	Number of columns of subplots

<code>figsize</code>	Size of the figure (width, height in inches)
<code>dpi</code>	Resolution of the figure
<code>sharex</code>	<code>True / False</code> or <code>'row' / 'col'</code> , share x-axis among subplots
<code>sharey</code>	<code>True / False</code> or <code>'row' / 'col'</code> , share y-axis among subplots
<code>squeeze</code>	If <code>True</code> , reduces the Axes array to lower dimensions if possible
<code>subplot_kw</code>	Dictionary of keyword arguments passed to each subplot (like <code>projection='3d'</code>)
<code>gridspec_kw</code>	Dictionary of keyword arguments for <code>GridSpec</code> layout (like <code>width_ratios</code> , <code>height_ratios</code>)

Example:

python Copy code

```
fig, axes = plt.subplots(2, 2, figsize=(10,8), sharex=True, sharey=True)
```

Summary Table

Function	Purpose	Key Arguments
<code>plt.figure()</code>	Create a new figure	<code>figsize</code> , <code>dpi</code> , <code>facecolor</code> , <code>edgecolor</code> , <code>num</code> , <code>tight_layout</code>
<code>plt.subplot()</code>	Select or create a single subplot	<code>nrows</code> , <code>ncols</code> , <code>index</code> , <code>projection</code> , <code>polar</code>
<code>plt.subplots()</code>	Create figure + multiple subplots	<code>nrows</code> , <code>ncols</code> , <code>figsize</code> , <code>dpi</code> , <code>sharex</code> , <code>sharey</code> , <code>squeeze</code> , <code>subplot_kw</code> , <code>gridspec_kw</code>

If you want, I can make a **visual diagram showing `figure`, `subplot`, and `subplots` with grids and axes** — it makes it much easier to remember which argument does what.

Do you want me to make that diagram?



