Pyplot:

1. Cách Import?

from matplotlib import pyplot as plt

1. Cách Hoạt Động?

* Mỗi cửa sổ là 1 Figure
* Bạn có thể lưu mỗi Figure vào 1 biến riêng biệt, rồi dùng nó để chỉnh sửa Figure đó
* Mỗi Figure ban đầu đều trắng trơn, sau đó bạn có thể vẽ lên đó nhiều Subplot
* Mỗi Subplot là 1 biểu đồ riêng biệt, ngoại trừ phần miền chữ nhật chính giữa và phần chữ, thì tất cả phần không gian còn lại đều trong suốt
* “plt” cũng là 1 Figure và cũng là 1 Subplot, nó tham chiếu đến Figure và Subplot được tạo ra cuối cùng, nếu dùng câu lệnh này trong trường hợp chưa có Figure nào được tạo, thì nó sẽ tự động tạo mới 1 Figure và 1 Subplot bao toàn Figure đó, nếu đã có Figure mà chưa có Subplot thì nó tạo mới 1 Subplot bao toàn Figure đó
* Tưởng tượng như xấp ảnh, bất kì đối tượng nào được Plot ra sau cũng sẽ che đối tượng cùng kiểu được Plot ra trước, để thay đổi thứ tự đối tượng hay ảnh trong xấp ảnh, dùng câu lệnh

<Đối Tượng>.set\_zorder(<Số Thứ Tự>)

* <Đối Tượng> có <Số Thứ Tự> lớn hơn sẽ che <Đối Tượng> có <Số Thứ Tự> nhỏ hơn

1. Tạo 1 Figure Mới?

<Figure> = plt.figure(num = <Index>, figsize = (<Ngang>, <Dọc>), dpi = <DPI>)

* <Figure> sẽ là 1 cửa sổ với chiều dài = <Ngang> \* <DPI> Pixel,

chiều dọc = <Dọc> \* <DPI> Pixel

* Lưu ý bạn cũng có thể thay đổi “figsize” bằng cách kéo thả lề cửa sổ <Figure> khi nó được Show
* <DPI> có giá trị mặc định = 100, nếu <DPI> càng nhỏ thì những đối tượng như đường, chữ, khung, … trên Figure sẽ có độ mập hoặc Font càng nhỏ
* <Index> dùng để chỉ định số thứ tự của Figure mới, nếu đã có Figure với số thứ tự này thì sẽ không có Figure mới được tạo thêm và <Figure> sẽ là Figure cũ với số thứ tự <Index>, nếu không có tham số này thì Figure mới sẽ tự động được tạo với số thứ tự là số thứ tự lớn nhất + 1

1. Trả Về Số Thứ Tự Của 1 Figure Nào Đó?

<Figure>.number

* Figure được tạo đầu tiên sẽ có số thứ tự 1, thằng tiếp theo được tạo sẽ có số thứ tự thứ 2, …

1. Cách Show Figure?

plt.show()

* Cơ chế hoạt động của câu lệnh này như sau
* Khi ta tạo mới 1 Figure, nó sẽ lưu vào 1 Stack, khi chạy câu lệnh trên, tất cả các Figure trong Stack sẽ được Show, đồng thời Stack trở về rỗng, tất cả dữ liệu về những Figure trong Stack đều bị xóa, số thự tự Figure cũng được Reset
* Chương trình sẽ tạm dừng cho đến khi bạn đóng tất cả Figure lại

1. Cách Tạo Hoạt Ảnh?

* Bản chất của hoạt ảnh là ta Show Figure rồi đóng lại nhiều lần liên tiếp
* Lệnh sau y chang khi Show Figure

plt.pause(<Số Giây>)

* Chỉ có điều, tất cả Figure sẽ tự động đóng lại sau khoảng thời gian = <Số Giây>, đồng thời không ảnh hưởng gì tới Stack lưu trữ Figure

1. Cách Tạo 1 Subplot Mới Trong 1 Figure Nào Đó?

<Subplot> = <Figure>.add\_subplot(

<Số Hàng>, <Số Cột>, <Index>,

projection = <Hệ Tọa Độ>

)

* Mặc định

|  |  |
| --- | --- |
| <Hệ Tọa Độ> | "rectilinear" |

* <Subplot> sẽ có kích thước và đặt tại ví trị với quy luật như sau
* Ban đầu, chia cửa sổ Figure thành các ô chữ nhật = nhau, sao cho có số hàng = <Số Hàng> và số cột = <Số Cột>, sau đó, đếm từ trái qua phải, từ trên xuống dưới, bắt đầu từ 1, đếm đến khi nào tới <Index> thì dừng, khi đó kích thước và vị trí của <Subplot> sẽ khít với hình chữ nhật tương ứng
* Nếu <Hệ Tọa Độ> = "rectilinear", thì <Subplot> có hệ tọa độ 2D, nếu = "3d", thì <Subplot> có hệ tọa độ 3D
* Mặc định khi chưa có dữ liệu, trục hoành, trục tung, trục cao của <Subplot> sẽ có cận (0, 1)

1. Kẻ Lưới Tọa Độ Cho 1 Subplot?

<Subplot>.grid()

* Các đường của lưới tọa độ sẽ đi qua các mốc của trục

1. Cách Plot Thêm Nhiều Line Lên 1 Subplot?

<Các Line> = <Subplot>.plot(<X1>, <Y1>, <X2>, <Y2>, <X3>, <Y3>, …)

* Xét cặp <Xn> và <Yn>, cả 2 đều là đối tượng Iterable và có số phần tử = nhau, khi đó <Xn>[m] và <Yn>[m] lần lượt là hoành độ và tung độ của điểm thứ m trong Line thứ n, các điểm trong 1 Line sẽ được nối với nhau theo đúng thứ tự Index
* <Các Line> là List có các phần tử là Line, ví dụ phần tử thứ 3 tương ứng với Line của cặp <X3>, <Y3>
* Nếu <Subplot> có hệ tọa độ 3D, thì cũng tương tự

<Các Line> = <Subplot>.plot(<X1>, <Y1>, <Z1>, <X2>, <Y2>, <Z2>, …)

1. Cách Plot Thêm Nhiều Điểm Lên 1 Subplot?

<Scatter> = <Subplot>.scatter(

<X>, <Y>, cmap = <Hệ Màu>, c = <Màu>,

vmin = <Cận Dưới>, vmax = <Cận Trên>, alpha = <Trong Suốt>

)

* Mặc định

|  |  |
| --- | --- |
| <Hệ Màu> | "viridis" |
| <Màu> | "1f77b4" |
| <Cận Dưới> | None |
| <Cận Trên> | None |
| <Trong Suốt> | None |

* <X>[m] và <Y>[m] lần lượt là hoành độ và tung độ của điểm thứ m
* Nếu <Hệ Màu> = "viridis", thì thang màu bắt đầu từ tím tới vàng, khi đó nếu chỉ định <Màu> = 1 List có phần tử là số biểu thị màu mỗi điểm, thì màu mỗi điểm tương ứng sẽ được nội suy theo <Cận Dưới> và <Cận Trên>, nếu <Cận Dưới> không được chỉ định, thì nó tự động là giá trị nhỏ nhất trong List, tượng tự

<Cận Trên> là giá trị lớn nhất trong List

* Nếu <Hệ Màu> = "rgb", thì nếu chỉ định <Màu> là 1 List thì List đó phải 2D, mỗi hàng mang giá trị R, G, B cho mỗi điểm tương ứng, giá trị từ 0 tới 1
* Nếu <Trong Suốt> = None thì mỗi điểm sẽ có độ trong suốt tùy theo góc nhìn, nếu = 1 số từ 0 đến 1 thì tất cả các điểm sẽ có cùng độ trong suốt =

<Trong Suốt>

* Nếu <Subplot> có hệ tọa độ 3D, thì cũng tương tự

<Scatter> = <Subplot>.scatter(<X>, <Y>, <Z>, …)

1. Cách Xóa Line, Scatter, Subplot, … Ra Khỏi 1 Figure?

<Line, Scatter, Subplot, …>.remove()

1. Tạo 1 Phát 1 Figure Mới Và Tạo 1 Đống Subplot Rỗng Trong Đó?

<Figure>, <Các Subplot> = plt.subplots(<Số Hàng>, <Số Cột>)

* <Figure> sẽ được chia thành các ô chữ nhật = nhau, sao cho có số hàng =

<Số Hàng> và số cột = <Số Cột>, mỗi ô sẽ chứa 1 Subplot rỗng

* <Các Subplot> là 1 List có Shape là (<Số Hàng>, <Số Cột>), mỗi phần tử là 1 <Subplot> tương ứng trong <Figure>

1. Chỉ Định Cận Của Trục Của 1 Subplot?

* Chỉ định cận trục hoành

<Subplot>.set\_xlim(<Cận Trái>, <Cận Phải>)

* Chỉ định cận trục tung

<Subplot>.set\_ylim(<Cận Dưới>, <Cận Trên>)

* Chỉ định cận trục cao

<Subplot>.set\_zlim(<Cận Dưới>, <Cận Trên>)

1. Chỉ Định Tên Của Trục Của 1 Subplot?

* Chỉ định tên trục hoành

<Subplot>.set\_xlabel(<Tên Trục Hoành>)

* Chỉ định tên trục tung

<Subplot>.set\_ylabel(<Tên Trục Tung>)

* Chỉ định tên trục cao

<Subplot>.set\_zlabel(<Tên Trục Cao>)

1. Chỉ Định Mốc Hiện Trên 1 Trục Của 1 Subplot?

* Chỉ định mốc trên trục hoành

<Subplot>.set\_xticks(<Dãy Các Mốc Hoành>)

* Chỉ định mốc trên trục tung

<Subplot>.set\_yticks(<Dãy Các Mốc Tung>)

* Chỉ định mốc trên trục cao

<Subplot>.set\_zticks(<Dãy Các Mốc Cao>)

* <Dãy Các Mốc> là 1 đối tượng Iterable chứa các phần tử là mốc
* Ví dụ

foo.set\_xticks([1, 2, 2.5])

1. Hiển Thị Rõ Trục Hoành Và Trục Tung Của 1 Subplot?

* Để hiện rõ trục hoành

<Line> = <Subplot>.avhline(

<Tung Độ>, <Cận Trái>, <Cận Phải>,

color = <Màu>, linestyle = <Kiểu Đường>, linewidth = <Độ Rộng>

)

* Mặc định

|  |  |  |  |
| --- | --- | --- | --- |
| <Tung Độ> | 0 | <Cận Trái> | 0 |
| <Cận Phải> | 1 | <Màu> | 1F77B4 |
| <Kiểu Đường> | "solid" | <Độ Rộng> | 2 |

* <Subplot> sẽ được vẽ đồ thêm 1 đoạn thẳng <Line> song song trục hoành ở tung độ <Tung Độ>, đoạn thẳng vẽ từ hoành độ = phần trăm chiều rộng Subplot tính từ bên trái được chỉ định bởi <Cận Trái>, tới hoành độ = phần trăm chiều rộng Subplot từ bên phải được chỉ định bởi <Cận Phải>
* Để hiện rõ trục tung

<Line> = <Subplot>.axvline(

<Hoành Độ>, <Cận Dưới>, <Cận Trên>,

color = <Màu>, linestyle = <Kiểu Đường>, linewidth = <Độ Rộng>

)

1. Vẽ 1 Đường Thẳng Tuyến Tính Thường?

<Line> = <Subplot>.axline(

<Điểm 1>, <Điểm 2>, slope = <Độ Dốc>,

color = <Màu>, linestyle = <Kiểu Đường>, linewidth = <Độ Rộng>

)

* <Điểm 1> phải có dạng (<Hoành Độ>, <Tung Độ>), tương tự <Điểm 2>
* Chỉ 1 trong 2 <Điểm 2> hoặc <Độ Dốc> được chỉ định

1. Cách Plot Thêm Nhiều Vector Lên 1 Subplot?

<Quiver> = <Subplot>.quiver(

<X>, <Y>, <U>, <V>,

angles = "xy", scale\_units = "xy", scale = 1,

width = <Độ Mập Vector>

)

* Gốc của Vector được Plot sẽ có tọa độ (<X>, <Y>) và hướng (<U>, <V>)
* <Độ Mập Vector> nên có giá trị khoảng 0.005
* Nếu muốn Plot nhiều Vector cùng lúc, thì chèn thêm giá trị vào <X>, <Y>, <U>, <V>, biến những tham số này thành Iterable
* <Quiver> là đối tượng lưu trữ Plot của đống Vector này
* Nếu <Subplot> có hệ tọa độ 3D, thì chỉ cần

<Quiver> = <Subplot>.quiver(<X>, <Y>, <Z>, <U>, <V>, <W>)

* Gốc của Vector được Plot sẽ có tọa độ (<X>, <Y>, <Z>) và hướng

(<U>, <V>, <W>)

1. Cách Plot 1 Polygon Lên 1 Subplot?

<Polygon> = <Subplot>.add\_patch(p = <Polygon>)

1. Lưu Ảnh 1 Figure?

<Figure>.save(<Tên File>, bbox\_inches = "tight")

* <Tên File> nếu không có phần mở rộng thì tự động phần mở rộng là “.png”
* Đặt “bbox\_inches = "tight"” để loại bỏ mấy cái lề trắng

1. Cách Show Ảnh Màu Trong 1 Subplot?

<Subplot>.imshow(<Tensor Ảnh Hoặc Numpy Array Ảnh>)

* Đảm bảo chiều cao xấp ảnh = 3, tấm trên cùng là đỏ, tấm giữa là xanh lá, tấm dưới là xanh dương, cầm chặt tấm ảnh lên rồi đặt lại góc tường sao cho mặt ảnh áp vào tường
* Lưu ý các giá trị trong xấp ảnh phải có giá trị từ 0 đến 1 nếu kiểu dữ liệu là Float và từ 0 đến 255 nếu kiểu dữ liệu là Integer, nằm ngoài khoảng này sẽ bị Clip

1. Cách Show Ảnh trắng đen trong 1 subplot?

<Subplot>.imshow(

<Tensor Ảnh Hoặc Numpy Array Ảnh>, cmap = "gray",

vmin = <Cận Dưới>, vmax = <Cận Trên>

)

* Đảm bảo ảnh có Shape 2D, kiểu dữ liệu là Float

1. Cách Show Ảnh Màu Nhanh Nhất?

plt.imshow(<Tensor Ảnh Hoặc Numpy Array Ảnh>)

plt.axis("off")

plt.show()

1. Cách Show Ảnh Trắng Đen Nhanh Nhất?

plt.imshow(

<Tensor Ảnh Hoặc Numpy Array Ảnh>, cmap = "gray",

vmin = <Cận Dưới>, vmax = <Cận Trên>

)

plt.axis("off")

plt.show()

Patches:

1. Cách Import?

from matplotlib import patches

1. Tạo 1 Polygon?

<Polygon> = patches.Polygon(<Danh Sách Tọa Độ Điểm>, color = <Màu>)

* <Danh Sách Tọa Độ Điểm> là 1 đối tượng Iterable chứa các phần tử cũng là đối tượng Iterable, mỗi Iterable con bao gồm 2 phần tử là hoành độ và tung độ
* Các điểm trong <Danh Sách Tọa Độ Điểm> sẽ được nối với nhau lần lượt theo đúng thứ tự, điểm cuối cùng sẽ được nối với điểm đầu tiên, sau đó tô màu hình vẽ được
* Ví dụ

foo = patches.Polygon([(1, 4), (5, 9), (-5, 2)], color = "red")