



Bài 6: Matplotlib



Nội dung bài học

01 Giới thiệu

02 Các biểu đồ cơ bản

03 Tùy chỉnh cá nhân

Lịch sử ra đời

- Năm 2002, John Hunter tạo ra một bản vá cho IPython để cho phép vẽ biểu đồ kiểu MATLAB tương tác thông qua từ dòng lệnh IPython.
- Năm 2003, Matplotlib 0.1 được phát hành.
- Matplotlib trở thành gói chiến lược của Viện Khoa học Kính viễn vọng Không gian.



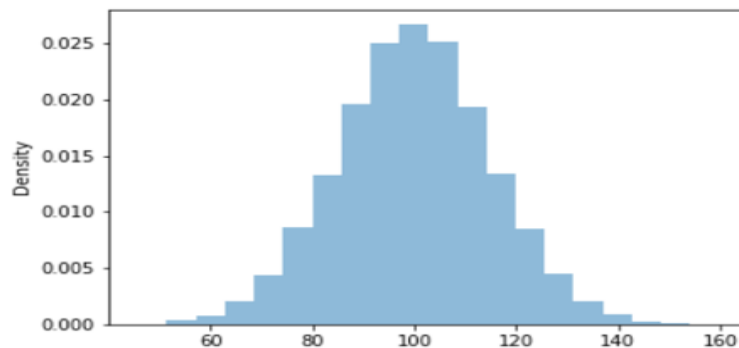
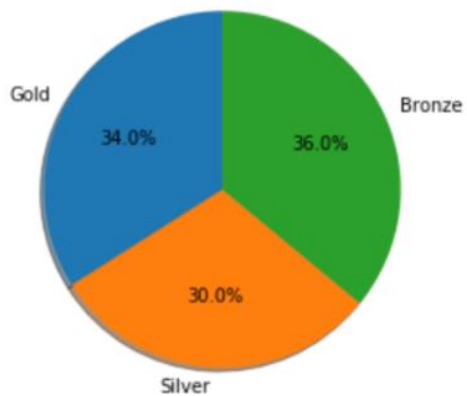
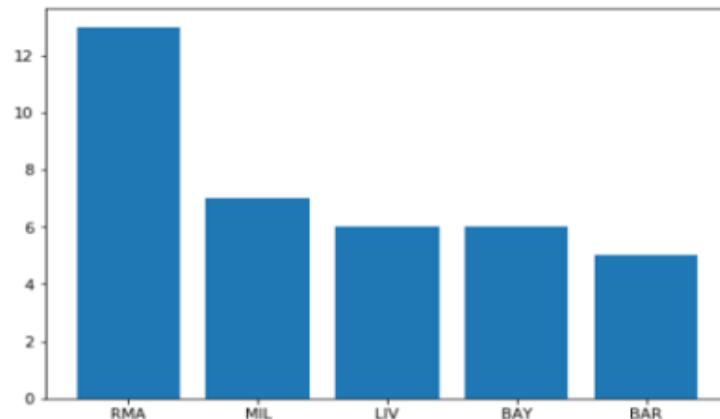
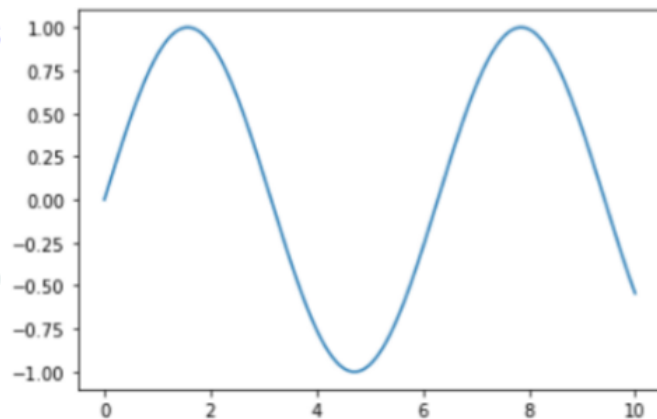
Hình: John
Hunter (1968
- 2012) - Cha
để của
Matplotlib



Ưu điểm

- ❖ Công cụ đồ họa đa nền tảng, đã được thử nghiệm tốt.
- ❖ Giúp đơn giản hóa việc vẽ biểu đồ chỉ bằng vài dòng lệnh.
- ❖ Hỗ trợ nhiều loại biểu đồ được sử dụng trong các lĩnh vực kinh tế, nghiên cứu như: biểu đồ dòng, cột, tần suất, tương quan,...
- ❖ Hiện đại và dễ dàng sử dụng hơn với sự phát triển của các API dựa trên nội dung của Matplotlib như : seaborn, ggpy, HoloViews, Altair,...

Ưu điểm





Cài đặt

❖ Dùng lệnh:

```
pip install matplotlib
```

❖ Import thư viện:

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



Nội dung bài học

01 Giới thiệu

02 Các biểu đồ cơ bản

03 Tùy chỉnh cá nhân



Biểu đồ đường (line chart)

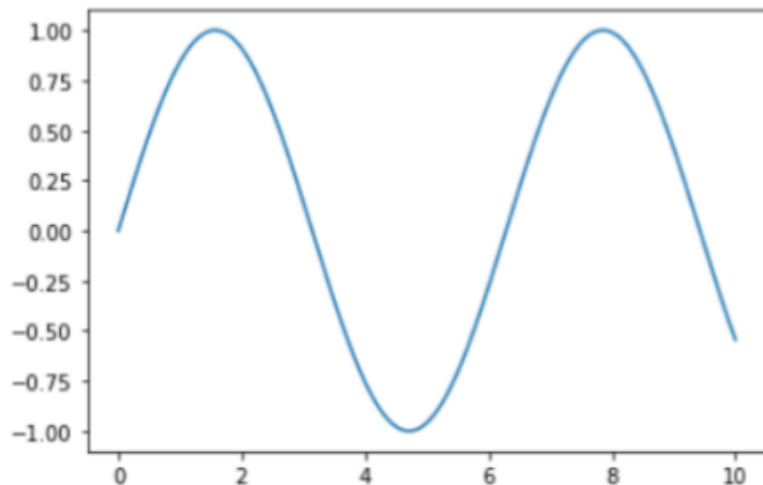
- ❖ Đặc điểm dữ liệu : một biến định lượng và một biến có thứ tự (định danh hoặc định lượng)
- ❖ Tác vụ:
 - ❖ Tiến trình phát triển
 - ❖ Tìm xu hướng phát triển
 - ❖ Biểu diễn phân bố
 - ❖ ...

Biểu đồ đường (line chart)

Yêu cầu trực quan

Trực quan hàm $y = \sin(x)$ bằng cách tạo ra 1000 điểm dữ liệu và nối lại.

```
# Tải thư viện cần thiết
import matplotlib.pyplot as plt
import numpy as np
# Tạo dữ liệu
x= np.linspace(0, 10, 1000)
y = np.sin(x)
# Vẽ biểu đồ đường (y = sin(x))
plt.plot(x,y)
plt.show()
```



Biểu đồ phân tán (Scatterplot)

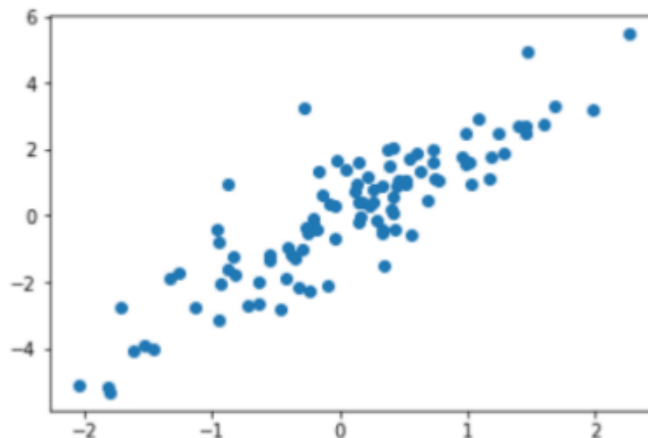
- ❖ Đặc điểm dữ liệu : Biểu diễn 2 biến định lượng.
- ❖ Tác vụ:
 - ❖ Tìm xu hướng (trend).
 - ❖ Phát hiện điểm dị biệt (outliers).
 - ❖ Thể hiện phân bố dữ liệu.
 - ❖ Thể hiện mối tương quan của dữ liệu (correlation).
 - ❖ Xác định các cụm dữ liệu (locate clusters).

Biểu đồ phân tán (Scatterplot)

Yêu cầu trực quan

Minh họa hàm $y = 2x + \epsilon$ với ϵ là một biến ngẫu nhiên theo phân phối chuẩn.

```
# Tải thư viện cần thiết
import matplotlib.pyplot as plt
import numpy as np
# Tạo dữ liệu
X= np.random.randn(100)
y= 2 * X + np.random.randn(100)
# Vẽ biểu đồ phân tán
plt.scatter(X, y)
plt.show()
```



Biểu đồ cột

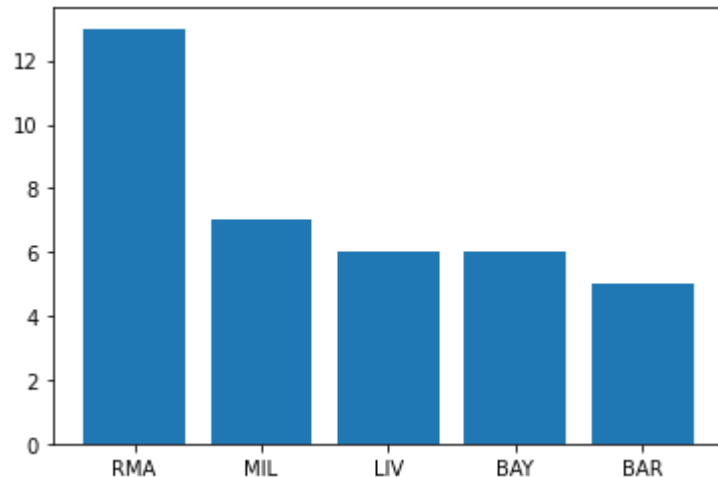
- ❖ Đặc điểm dữ liệu : một biến định lượng và một biến định danh.
- ❖ Tác vụ:
 - ❖ Tìm kiếm giá trị
 - ❖ So sánh các giá trị.
 - ❖ ...

Biểu đồ cột

Yêu cầu trực quan

Trực quan 5 đội vô địch C1 nhiều nhất là : Real Madrid (13 lần), AC Milan - (7 lần), Liverpool - (6 lần), Bayern Munich - (6 lần) và Barcelona (5 lần).

```
import matplotlib.pyplot as plt
fc=["RMA","MIL","LIV","BAY","BAR"]
time=[13,7,6,6,5]
plt.bar(fc,time)
plt.show()
```

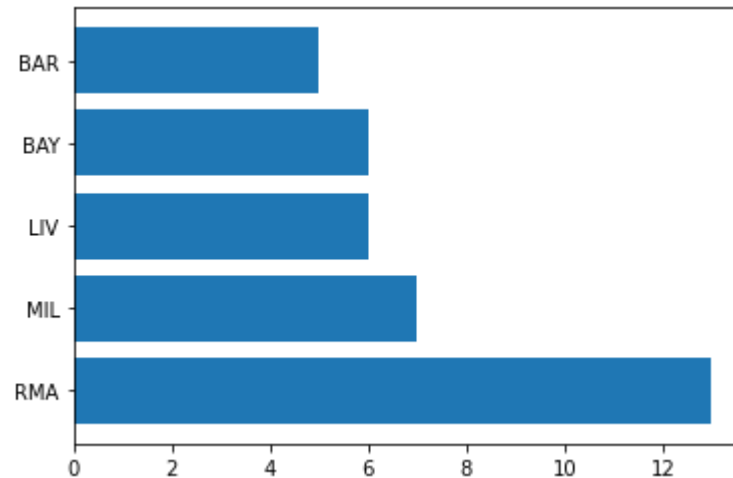


Biểu đồ cột

Yêu cầu trực quan

Trực quan 5 đội vô địch C1 nhiều nhất là : Real Madrid (13 lần), AC Milan - (7 lần), Liverpool - (6 lần), Bayern Munich - (6 lần) và Barcelona (5 lần).

```
import matplotlib.pyplot as plt
fc=["RMA","MIL","LIV","BAY","BAR"]
time=[13,7,6,6,5]
plt.barh(fc,time)
plt.show()
```





Biểu đồ tròn

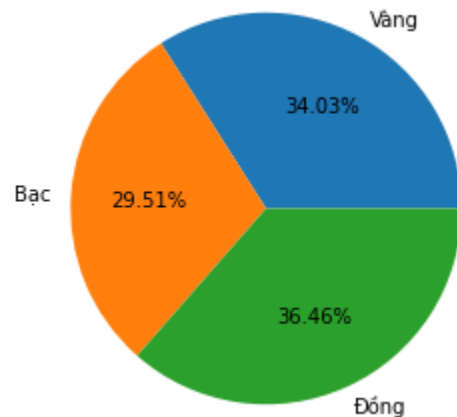
- ❖ Đặc điểm dữ liệu : một biến định lượng và một biến định danh.
- ❖ Tác vụ : thể hiện mối quan hệ với tổng thể.

Biểu đồ tròn

Yêu cầu trực quan

Trực quan phần trăm huy chương của đoàn thể thao Việt Nam tại Seagame 30 (98 vàng, 85 bạc và 105 đồng).

```
import matplotlib.pyplot as plt
huychuong=["Vàng","Bạc","Đồng"]
so=[98,85,105]
plt.pie(so,labels=huychuong,autopct="%1.2f%%")
plt.axis("equal")
plt.show()
```



Biểu đồ cột chồng

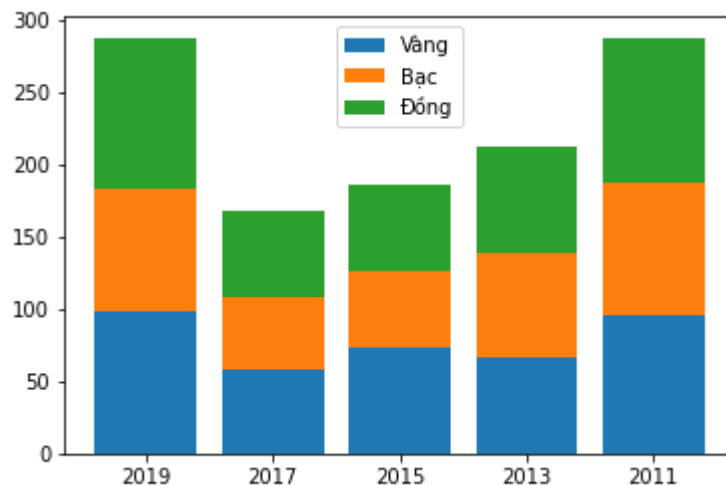
- ❖ Đặc điểm dữ liệu : Một biến định lượng và hai biến định danh làm khóa.
- ❖ Tác vụ:
 - ❖ Thể hiện quan hệ với tổng thể
 - ❖ Tìm kiếm giá trị
 - ❖ Tìm xu hướng
 - ❖ ...

Biểu đồ cột chồng

Yêu cầu trực quan

Trực quan hóa số lượng huy chương của đoàn thể thao Việt Nam qua các 5 kì seagame gần nhất.

```
import numpy as np
import matplotlib.pyplot as plt
vang=[98,58,73,66,96]
bac=[85,50,53,73,92]
dong=[105,60,60,74,100]
years=["2019","2017","2015","2013","2011"]
tam=np.add(vang,bac).tolist()
plt.bar(years,vang,label="Vàng")
plt.bar(years,bac,bottom=vang,label="Bạc")
plt.bar(years,dong,bottom=tam,label="Đồng")
plt.legend()
plt.show()
```





Biểu đồ histogram

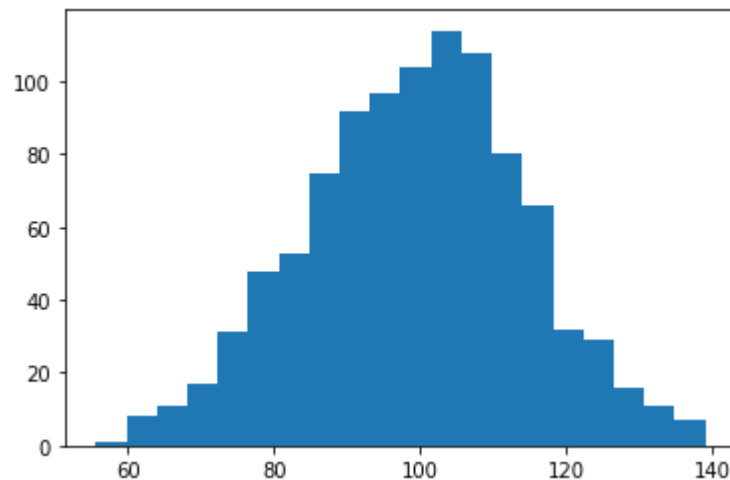
- ❖ Đặc điểm dữ liệu : một biến định lượng.
- ❖ Tác vụ:
 - ❖ Tìm phân bố
 - ❖ Phát hiện điểm dị thường
 - ❖ ...

Biểu đồ histogram

Yêu cầu trực quan

Trực quan hóa phân phối chuẩn với trung bình bằng 100 và độ lệch chuẩn là 15.

```
import numpy as np
import matplotlib.pyplot as plt
x = np.random.normal(100, 15, 1000)
plt.hist(x, 20)
plt.show()
```

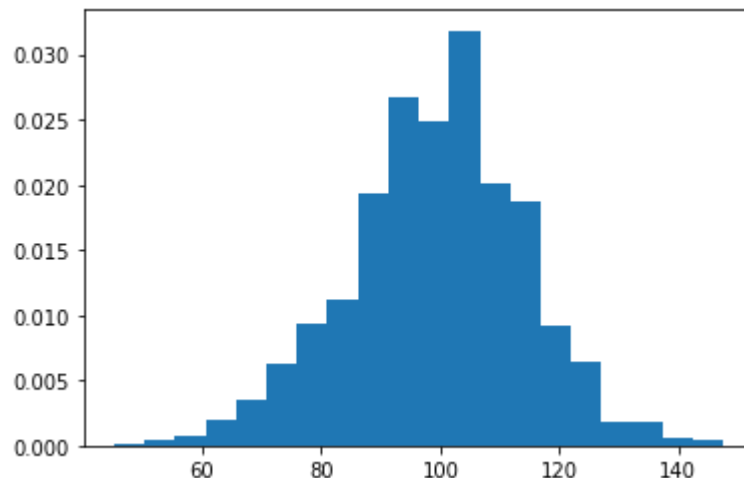


Biểu đồ mật độ - density

Cú pháp

Thêm thuộc tính `density = True` vào hàm `hist()`

```
import numpy as np
import matplotlib.pyplot as plt
x = np.random.normal(100, 15, 1000)
plt.hist(x, 20, density=True)
plt.show()
```



boxplot

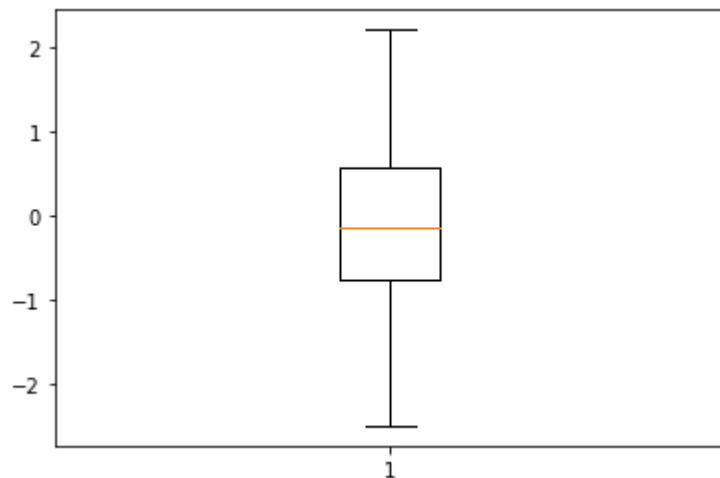
- ❖ Đặc điểm dữ liệu : một biến định lượng.
- ❖ Tác vụ:
 - ❖ Tìm xu hướng.
 - ❖ Phát hiện điểm dị thường
- ❖ Thể hiện 5 giá trị của dữ liệu:
 - ❖ Giá trị lớn nhất (Max)
 - ❖ Giá trị nhỏ nhất (Min)
 - ❖ Trung vị (Medium)
 - ❖ Phân vị thứ nhất (Q1)
 - ❖ Phân vị thứ 3 (Q3)

boxplot

Yêu cầu trực quan

Vẽ biểu đồ boxplot cho phân phối chuẩn hóa

```
import numpy as np
import matplotlib.pyplot as plt
x = np.random.normal(size=100)
plt.boxplot(x)
plt.show()
```



Biểu đồ streamplot

Ý nghĩa

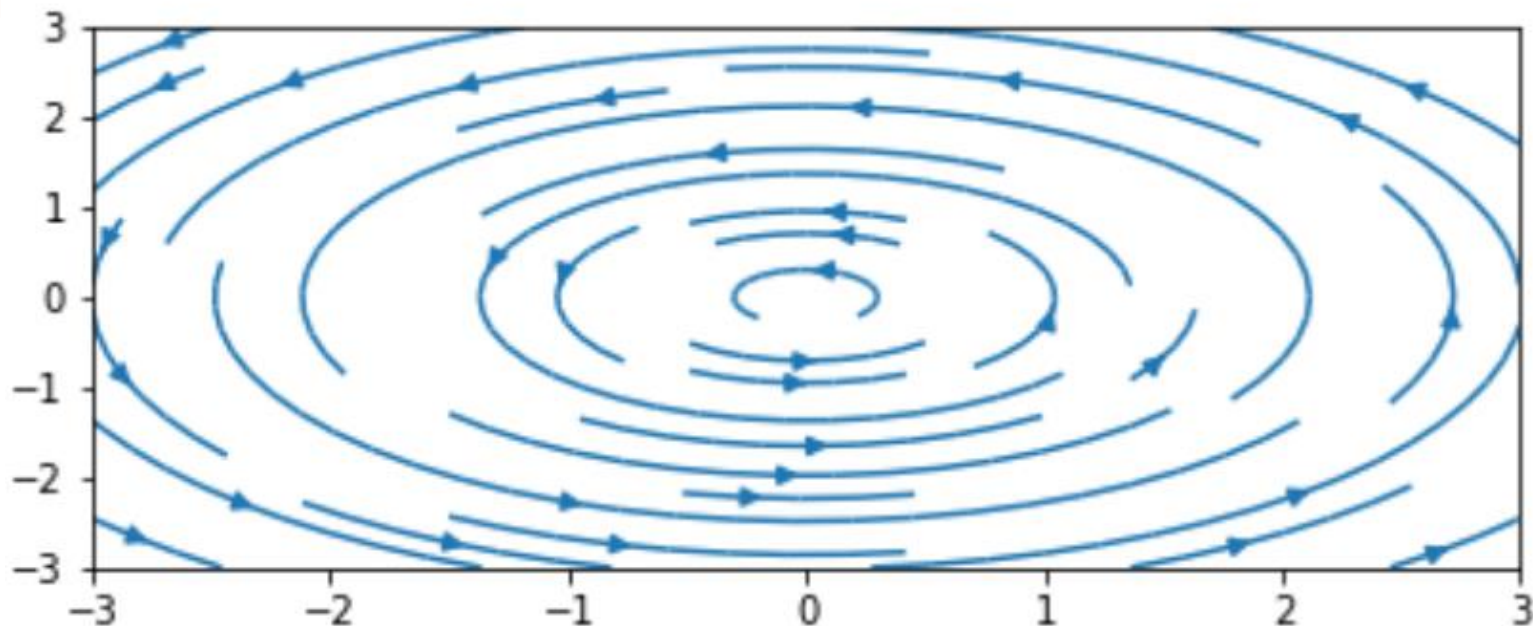
Được sử dụng để vẽ các biểu đồ trong trường vector.

Yêu cầu trực quan

Biểu diễn trường vector $\vec{F}(x,y) = \langle -y, x \rangle$

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.gridspec as grid
w = 3; U = -Y ; V = X
Y,X=np.mgrid[-w:w:100j,-w:w:100j]
speed=np.sqrt(U**2 + V**2)
fig=plt.figure(figsize=(12,10))
gs=grid.GridSpec(nrows = 3,ncols=2, height_ratios=[1,1,2])
ax=fig.add_subplot(gs[0,0])
ax.streamplot(X, Y, U, V, density = [0.4, 0.8])
plt.show()
```

Biểu đồ streamplot



Hình: Biểu đồ streamplot

Contour

Công dụng

Dùng để thể hiện các *đường đồng mức*. Thường thấy trong các bản đồ địa hình, bản đồ đường đẳng áp, đường đẳng nhiệt, ...

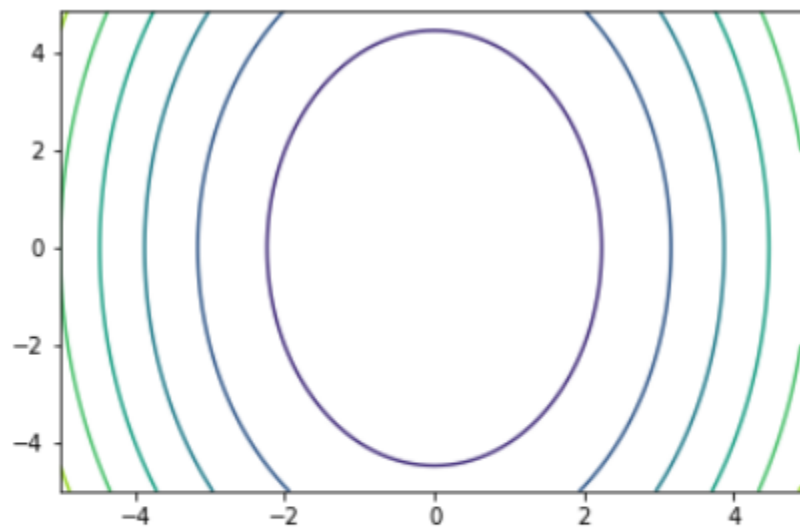
Yêu cầu trực quan

Phác họa vài đường đồng mức của hàm số $f(x, y) = 4x^2 + y$

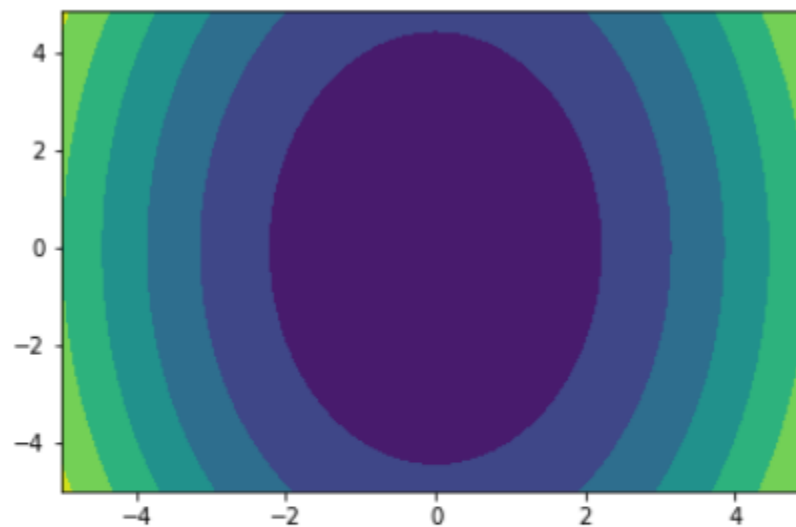
Code

```
import matplotlib.pyplot as plt
x = np.arange(-5, 5, 0.1)
y = np.arange(-5, 5, 0.1)
xx, yy = np.meshgrid(x, y)
z = 4*xx**2 + yy**2
plt.contour(x, y, z)
plt.show()
```

Contour



Hình: Kết
quả hàm
`contour()`



Hình: Kết
quả hàm
`contourf()`



Nội dung bài học

01 Giới thiệu

02 Các biểu đồ cơ bản

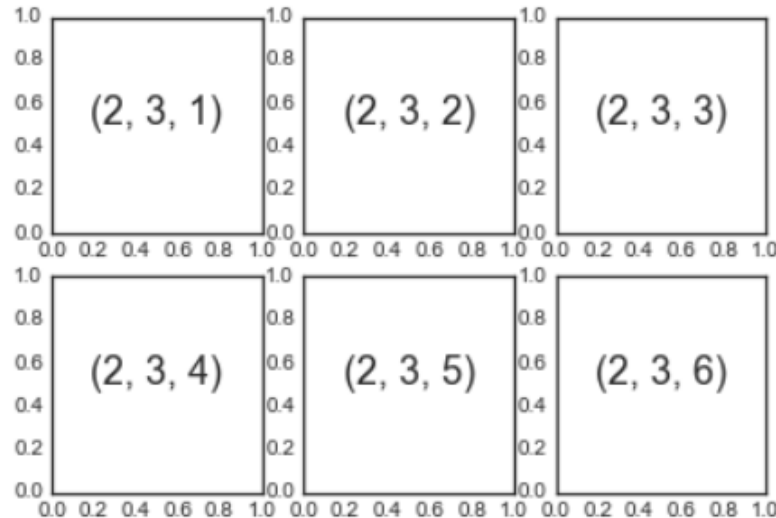
03 Tùy chỉnh cá nhân

Nhiều biểu đồ trong một hình - kiểu MATLAB

Cú pháp

Sử dụng hàm `subplot(m , n , p)` để tạo ra một lưới $m \times n$ phần tử và p là vị trí trong lưới.

Cách bố trí trong lưới 2×3 như hình bên dưới:

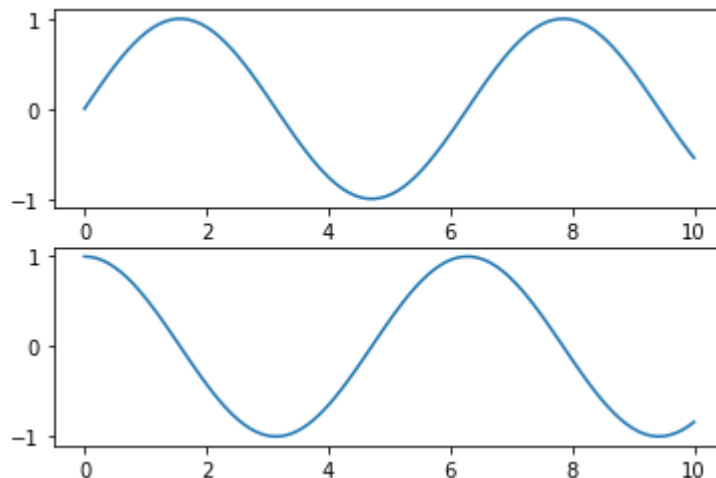


Nhiều biểu đồ trong một hình - kiểu MATLAB

Yêu cầu trực quan

Vẽ hai biểu đồ $y = \sin(x)$ và $y = \cos(x)$ trong cùng một hình minh họa.

```
import numpy as np
import matplotlib.pyplot as plt
x=np.linspace(0,10,100)
#Hình trên:
plt.subplot(2,1,1)
plt.plot(x, np.sin(x))
#Hình dưới:
plt.subplot(2,1,2)
plt.plot(x,np.cos(x))
plt.show()
```



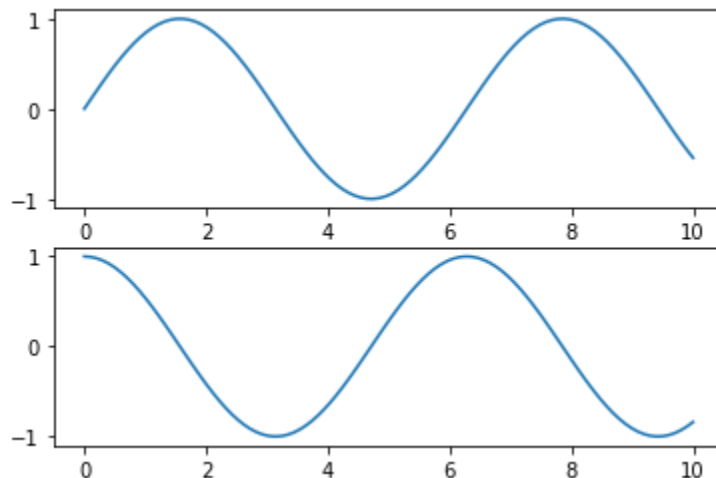
Nhiều biểu đồ trong một hình - kiểu hướng đối tượng

Yêu cầu trực quan

Vẽ hai biểu đồ $y = \sin(x)$ và $y = \cos(x)$ trong cùng một hình minh họa.
(Vẽ kiểu hướng đối tượng)

```
import numpy as np
import matplotlib.pyplot as plt
x=np.linspace(0,10,100)

fig,ax=plt.subplots(2)
#Hình trên:
ax[0].plot(x,np.sin(x))
#Hình dưới:
ax[1].plot(x,np.cos(x))
plt.show()
```



Thêm nhãn và tiêu đề

Thêm nhãn 2 trục:

- Trục x: `xlabel("Chuỗi")`
- Trục y: `ylabel("Chuỗi")`

Thêm tiêu đề: `title("Chuỗi")`

Tùy chỉnh font : thêm tham số `fontdict` trong `xlabel()`, `ylabel()`, `title()`

Tùy chỉnh vị trí tiêu đề : thêm tham số `loc` vào `title()`

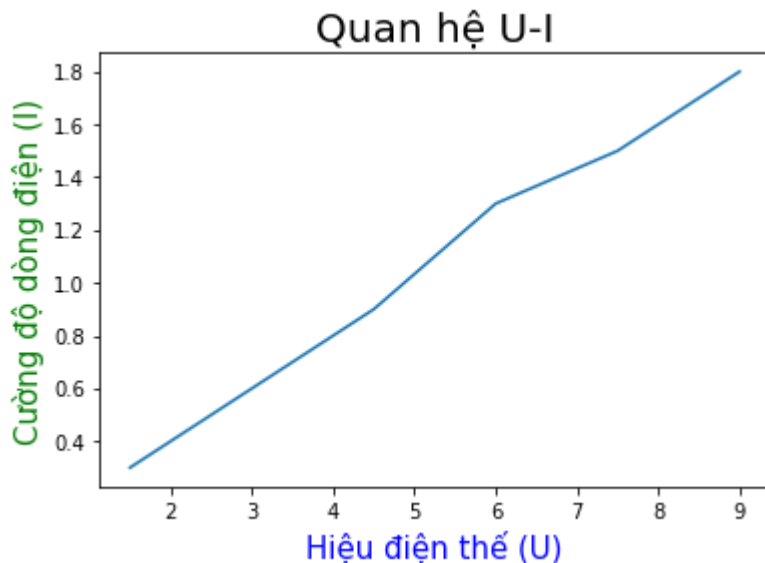
Thêm nhãn và tiêu đề

Yêu cầu trực quan

Vẽ biểu thể hiện mối quan hệ giữa cường độ dòng điện và hiệu điện thế.

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

U=[1.5,3.0,4.5,6.0,7.5,9.0]
I=[0.3,0.6,0.9,1.3,1.5,1.8]
plt.plot(U,I)
plt.xlabel("Hiệu điện thế  
(U)",fontdict={"color":"blue","size":15})
plt.ylabel("Cường độ dòng điện (I)",
fontdict={"color":"green","size":15})
plt.title("Quan hệ U-  
I",fontdict={"size":20})
plt.show()
```



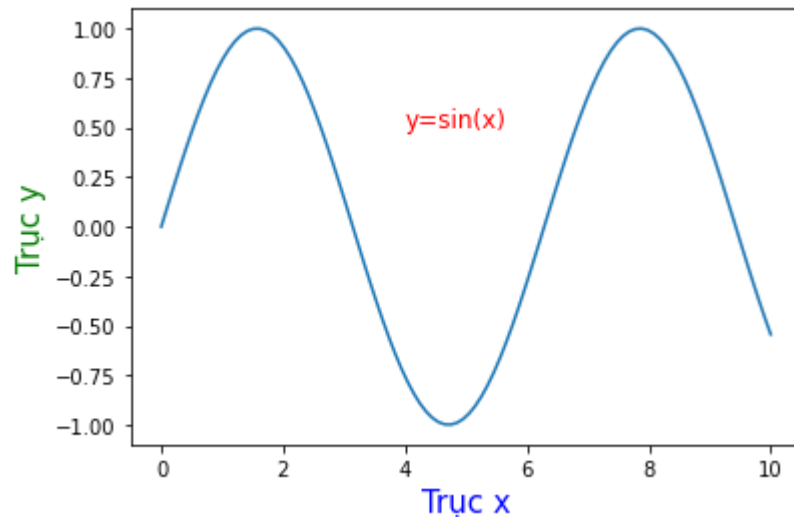
Thêm văn bản

Cú pháp

Sử dụng hàm `text(x, y, s, ...)` với các x, y là tọa độ cần đặt văn bản và s là văn bản.

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

x=np.linspace(0,10,100)
y=np.sin(x)
font1={"c":"b","size":15}
font2={"c":"g","size":15}
font3={"c":"r","size":12}
plt.plot(x,y)
plt.xlabel("Trục x",fontdict=font1)
plt.ylabel("Trục y",fontdict=font2)
plt.text(4,0.5,"y=sin(x)",fontdict=font3)
plt.show()
```



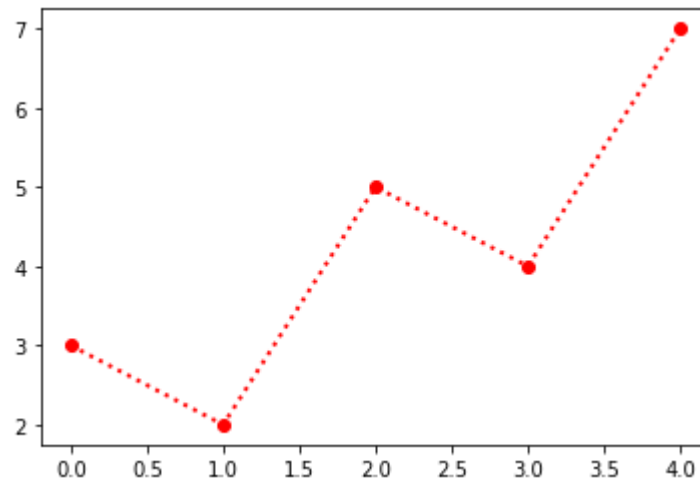
Tùy chỉnh đường bằng fmt

➤ “fmt” gồm 3 phần: **[marker]** | **[line]** | **[color]**

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

x=np.array([3,2,5,4,7])

plt.plot(x,"o:r")
plt.show()
```



Tùy chỉnh đường bằng fmt

➤ [marker]:cách đánh dấu dữ liệu

Marker	Description
'o'	Circle
'*'	Star
'.'	Point
','	Pixel
'x'	X
'X'	X (filled)
'+'	Plus
'P'	Plus (filled)
's'	Square
'D'	Diamond

'd'	Diamond (thin)
'p'	Pentagon
'H'	Hexagon
'h'	Hexagon
'v'	Triangle Down
'^'	Triangle Up
'<'	Triangle Left
'>'	Triangle Right

'1'	Tri Down
'2'	Tri Up
'3'	Tri Left
'4'	Tri Right
' '	Vline
'_'	Hline

Tùy chỉnh đường bằng fmt

➤ [line]:cách vẽ đường

Line Syntax	Description
'_'	Solid line
'.'	Dotted line
'-'	Dashed line
'-.'	Dashed/dotted line

Tùy chỉnh đường bằng fmt

➤ [color]:màu

Color Syntax	Description
'r'	Red
'g'	Green
'b'	Blue
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'w'	White

#rrggbb: mã màu theo hệ RGB

Tùy chỉnh bằng rcParams

- Thay đổi các giá trị mặc định bằng hàm `rcParams[]`, Ví dụ:

```
plt.rcParams['lines.linewidth'] = 2  
plt.rcParams['lines.linestyle'] = '--'
```
- Thay đổi nhiều giá trị cùng một lúc, sử dụng hàm `rc()`, ví dụ:

```
plt.rc('lines', linewidth=2, linestyle='--')
```
- Để khôi phục cài đặt, sử dụng hàm `rcdefaults()`
- Các thuộc tính có thể tùy chỉnh được lưu trong file `matplotlibrc`

Thank you and
happy learning
!!!