

THƯ VIỆN MATPLOTLIB

Quá trình phân tích dữ liệu là không đủ để tổng kết kết quả hoặc tóm tắt dữ liệu bên trong. Bởi vì chúng ta cần phải thể hiện được kết quả phân tích dữ liệu một cách hiệu quả và chúng ta cần một số các đồ thị hoặc biểu đồ để biểu diễn kết quả được phân tích, cho nên trực quan hóa dữ liệu là bước rất quan trọng trong các quá trình phân tích dữ liệu.

Trực quan hóa dữ liệu là một phần thiết yếu của phân tích dữ liệu; trong chương này, chúng ta sẽ học những điều cơ bản về trực quan hóa dữ liệu với sự trợ giúp của thư viện Matplotlib, một thư viện được sử dụng rộng rãi trong Python dành cho việc trực quan hóa dữ liệu.

Cấu trúc: Trong chương này, chúng ta sẽ thảo luận về các chủ đề sau:

- Trực quan hóa dữ liệu là gì?
- Thư viện Matplotlib là gì?
- Bắt đầu làm việc với Matplotlib
- Vẽ đồ thị đường thẳng đơn giản bằng cách sử dụng Matplotlib
- Hướng đối tượng API trong Matplotlib
- Hàm subplot() trong Matplotlib
- Chỉnh sửa đồ thị:
 - Thêm tiêu đề
 - Thêm nhãn trục
 - Thêm chữ trong đồ thị
 - Thêm đánh dấu trong đồ thị
 - Thêm kiểu nét, màu nét, và độ rộng của nét
 - Thêm đường lưới
 - Cài đặt giới hạn trục
 - Thêm các nhãn đánh dấu
- Một số kiểu đồ thị cơ bản trong Matplotlib:
 - Biểu đồ cột
 - Biểu đồ tần suất
 - Biểu đồ phân tán
 - Biểu đồ tròn
- Xuất đồ thị vào tệp:
 - Xuất đồ thị vào tệp pdf
 - Xuất đồ thị vào tệp jpeg

Mục tiêu: Sau khi hoàn thành nghiên cứu xong chương này, có thể trả lời các câu hỏi sau:

- Thư viện Matplotlib là gì, và tại sao chúng ta lại cần trực quan hóa dữ liệu?
- Làm sao để vẽ đồ thị bằng các phương thức và lựa chọn của matplotlib?
- Làm sao để cài đặt đồ thị với từng kiểu yêu cầu?
- Làm sao để xuất đồ thị vào một tệp pdf và jpeg?

1. Trực quan hóa dữ liệu là gì?

Trực quan hóa dữ liệu là sự thể hiện của dữ liệu dưới dạng đồ thị, bằng cách sử dụng biểu đồ trực quan, đồ họa trực quan,... Trực quan hóa dữ liệu nhằm mục đích thể hiện dữ liệu bằng hình ảnh để hiểu câu chuyện đằng sau dữ liệu theo cách tương tác. Nói cách khác, trực quan hóa dữ liệu là một cách kể câu chuyện về các sự kiện mà chúng ta đã thu thập hoặc kết luận từ quá trình phân tích dữ liệu. Có nhiều công cụ khác nhau dành riêng cho việc trực quan hóa dữ liệu, như Tableau và Power BI. Python cũng có các thư viện hiệu quả và hữu ích như Matplotlib, Seaborn, Plotly, Morrison,...

2. Matplotlib là gì?

Matplotlib là một thư viện được sử dụng rộng rãi và mạnh mẽ của Python, **là thư viện giúp thể hiện thông tin trực quan với sự trợ giúp của biểu đồ và đồ họa trực quan**. Nó có rất nhiều đồ thị để có thể biểu diễn dữ liệu theo yêu cầu và sự chỉ định.

3. Bắt đầu làm việc với thư viện Matplotlib

Đầu tiên, tải nền tảng Anaconda cho Python, và khi đó thư viện Matplotlib đã được tải. Ngoài ra, cần tải cái này để sử dụng kho quản lý như pip với ví dụ sau đây:

```
C:\>pip install matplotlib
```

Khi đã tải Matplotlib, nhập chúng vào và sử dụng như bên dưới:

```
import matplotlib
print("Running Version of Matplotlib is:",matplotlib.__version__)
```

4. Vẽ đồ thị đường thẳng đơn giản bằng cách sử dụng Matplotlib

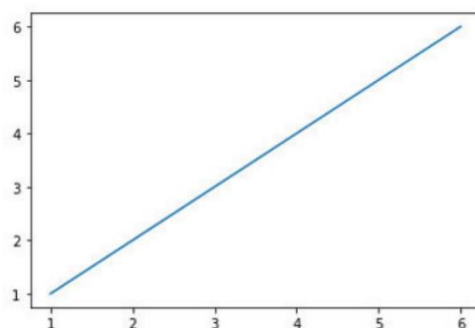
Ví dụ về cách vẽ đồ thị đường thẳng bằng cách sử dụng thư viện Matplotlib như dưới đây:

```
import matplotlib.pyplot as plt

x=[1,2,3,4,5,6]
y=[1,2,3,4,5,6]

plt.plot(x,y)
```

[<matplotlib.lines.Line2D at 0x1b49d7ee0a0>]



5. Hướng đối tượng API trong matplotlib

Mô-đun pyplot là một phương thức chức năng và sử dụng giao diện dựa trên trạng thái.

Nhưng Matplotlib còn có một giao diện khác, đó là giao diện hướng đối tượng. Trong trường hợp này, sử dụng cách: **matplotlib.pyplot** hoặc đối tượng của lớp **matplotlib.axis.Axis** hoặc đối tượng của lớp **matplotlib.figure.figure**.

Hình trong Matplotlib là vùng chứa cấp cao nhất để vẽ tất cả các thành phần của biểu đồ, hình có thể chứa một hoặc nhiều trục và trục là vùng được thêm vào trên hình được sử dụng để vẽ dữ liệu hoặc có thể nói rằng trục là biểu đồ riêng lẻ trên hình.

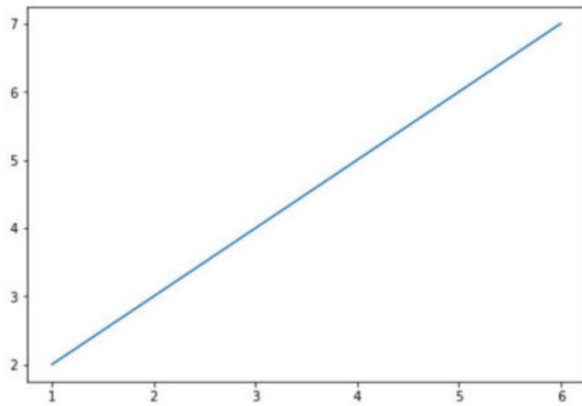
Ví dụ:

```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

ax.plot(x,y)

plt.show()
```



6. Hàm subplot() trong matplotlib

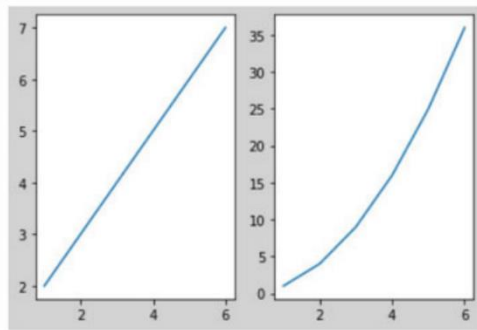
Khi cần vẽ nhiều đồ thị trong một hình thì sử dụng hàm subplot(). Ví dụ::

- Ví dụ 1 (đồ thị 1x2):

```
import matplotlib.pyplot as plt

fig, axes = plt.subplots(nrows=1, ncols=2)
fig.set_facecolor('lightgray')
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]
x_square = [x*x for x in x]

axes[0].plot(x,y)
axes[1].plot(x,x_square)
plt.show()
```



- Ví dụ 2 (đồ thị 2x2):

```
import matplotlib.pyplot as plt

fig, axes = plt.subplots(nrows=2, ncols=2)
fig.set_facecolor('lightgray')
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]
x_square = [x*x for x in x]
y_square = [y*y for y in y]

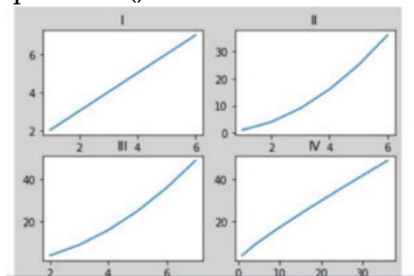
axes[0][0].plot(x,y)
axes[0][0].set_title("I")

axes[0][1].plot(x,x_square)
axes[1][0].set_title("II")

axes[1][0].plot(y,y_square)
axes[1][0].set_title("III")

axes[1][1].plot(x_square,y_square)
axes[1][1].set_title("IV")

plt.show()
```



7. Chỉnh sửa đồ thị

Sử dụng một số các hàm sau để chỉnh sửa đồ thị:

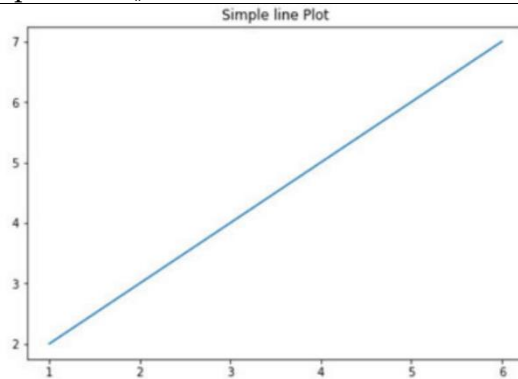
- Thêm tiêu đề:

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

```
fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

ax.set_title("Simple line Plot",loc='center')
ax.plot(x,y)

plt.show()
```



- Thêm nhãn trục:

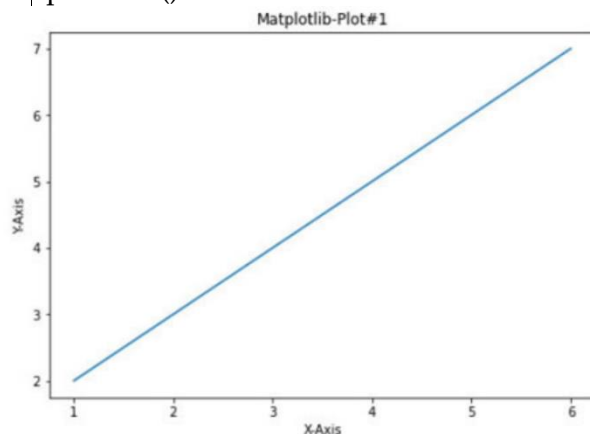
```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

ax.set_title("Simple lin Plot",loc='center')

ax.set_xlabel("X-Axis")
ax.set_ylabel("Y-Axis")

ax.plot(x,y)

plt.show()
```



- Thêm chữ trong đồ thị:

```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

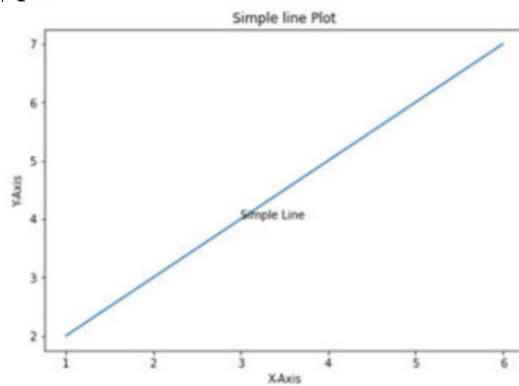
ax.set_title("Simple lin Plot", loc='center')

ax.set_xlabel("X-Axis")
ax.set_ylabel("Y-Axis")

plt.text(3,4,"Simple Line")

ax.plot(x,y)

plt.show()
```



- Thêm đánh dấu:

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

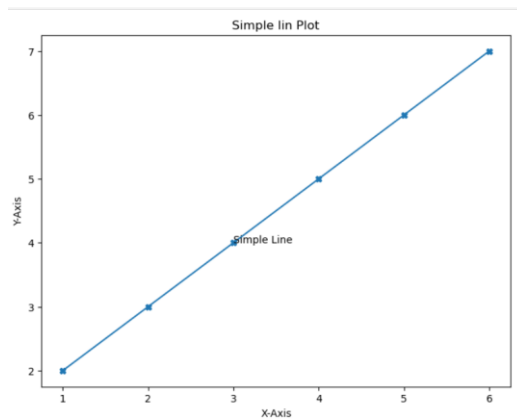
ax.set_title("Simple lin Plot", loc='center')

ax.set_xlabel("X-Axis")
ax.set_ylabel("Y-Axis")

plt.text(3,4,"Simple Line")

ax.plot(x,y,marker='X')

plt.show()
```



Bảng sau đây là các loại đánh dấu điểm trong Matplotlib:

Giá trị đánh dấu	Thể hiện
marker = 'o'	Hình tròn
marker = '*'	Hình sao
marker = '.'	Điểm
marker = ','	Pixel
marker = 'x'	X
marker = 'X'	X đậm
marker = '+'	Dấu cộng
marker = 'P'	Dấu cộng đậm
marker = 's'	Hình vuông
marker = 'D'	Hình kim cương
marker = 'd'	Hình kim cương mảnh
marker = 'p'	Hình năm cạnh
marker = 'H'	Hình sáu cạnh
marker = 'h'	Hình sáu cạnh
marker = 'v'	Hình mũi tên hướng lên
marker = '^'	Hình mũi tên hướng xuống
marker = '<'	Hình mũi tên hướng sang trái
marker = '>'	Hình tam giác phải
marker = '1'	Mũi tên xuống
marker = '2'	Mũi tên lên
marker = '3'	Mũi tên trái
marker = '4'	Mũi tên phải
marker = ' '	Vline
marker = '-'	Hline

- Thêm kiểu nét, màu nét, và độ rộng của nét

```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]
```

```

ax.set_title("Simple lin Plot", loc='center')

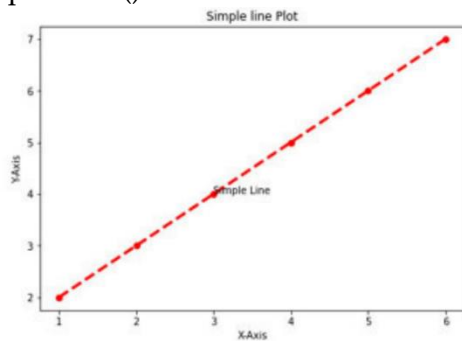
ax.set_xlabel("X-Axis")
ax.set_ylabel("Y-Axis")

plt.text(3,4,"Simple Line")

ax.plot(x,y,marker='p', linestyle="--", color='red', linewidth='3')

plt.show()

```



- Thêm lưới

Ví dụ 1: grid()

```

import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

ax.set_title("Simple lin Plot", loc='center')

ax.set_xlabel("X-Axis")
ax.set_ylabel("Y-Axis")

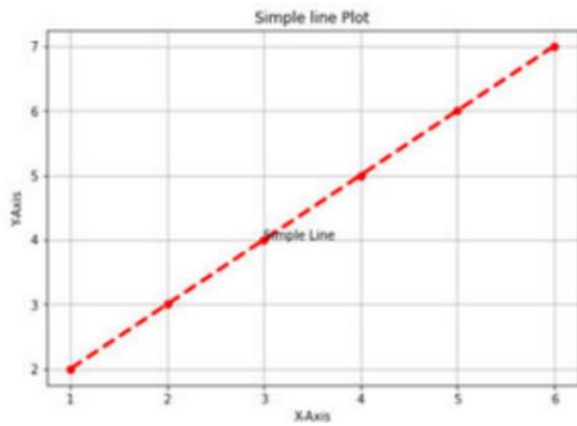
plt.text(3,4,"Simple Line")

ax.plot(x,y,marker='o', linestyle="--", color='red', linewidth='3')

ax.grid()

plt.show()

```

Ví dụ 2: `grid(axis='x')`

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

```
fig = plt.figure()
```

```
ax = fig.add_axes([0.1,0.1,1,1])
```

```
x=[1,2,3,4,5,6]
```

```
y=[2,3,4,5,6,7]
```

```
ax.set_title("Simple lin Plot", loc='center')
```

```
ax.set_xlabel("X-Axis")
```

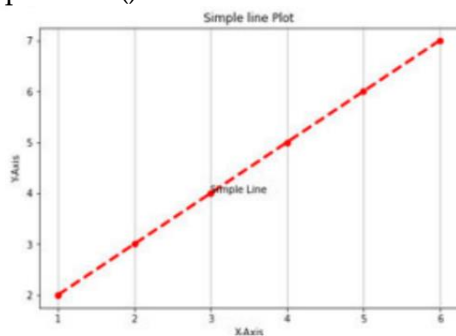
```
ax.set_ylabel("Y-Axis")
```

```
plt.text(3,4,"Simple Line")
```

```
ax.plot(x,y,marker='o', linestyle="--", color='red', linewidth='3')
```

```
ax.grid(axis='x')
```

```
plt.show()
```



Ví dụ 3: grid(axis='y')

```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

ax.set_title("Simple lin Plot", loc='center')

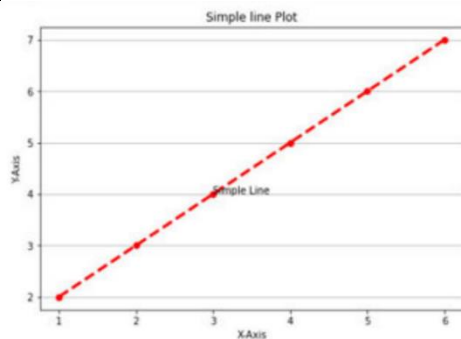
ax.set_xlabel("X-Axis")
ax.set_ylabel("Y-Axis")

plt.text(3,4,"Simple Line")

ax.plot(x,y,marker='o', linestyle="--", color='red', linewidth='3')

ax.grid(axis='y')

plt.show()
```



- Cài đặt giới hạn trục:

```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

ax.set_title("Simple lin Plot", loc='center')

ax.set_xlabel("X-Axis")
ax.set_ylabel("Y-Axis")

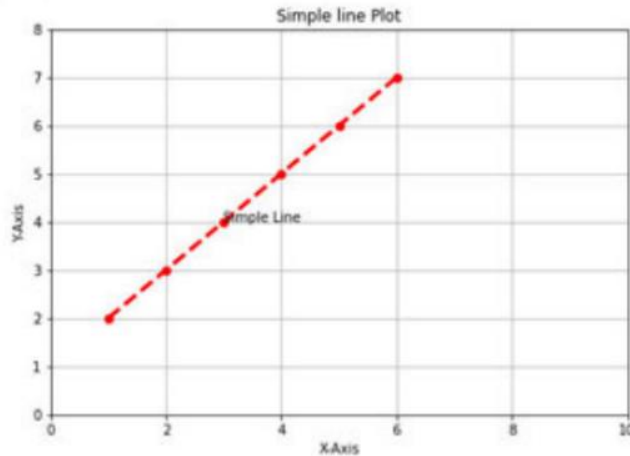
ax.set_xlim(0,10)
ax.set_ylim(0,8)
```

```
plt.text(3,4,"Simple Line")

ax.plot(x,y,marker='o', linestyle="--", color='red', linewidth='3')

ax.grid(axis='y')

plt.show()
```



- Thêm nhãn đánh dấu

Ví dụ 1: set_xticks() và set_xticklabels()

```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

ax.set_title("Simple lin Plot", loc='center')

ax.set_xlabel("X-Axis")
ax.set_ylabel("Y-Axis")

ax.set_xlim(0,10)
ax.set_ylim(0,8)

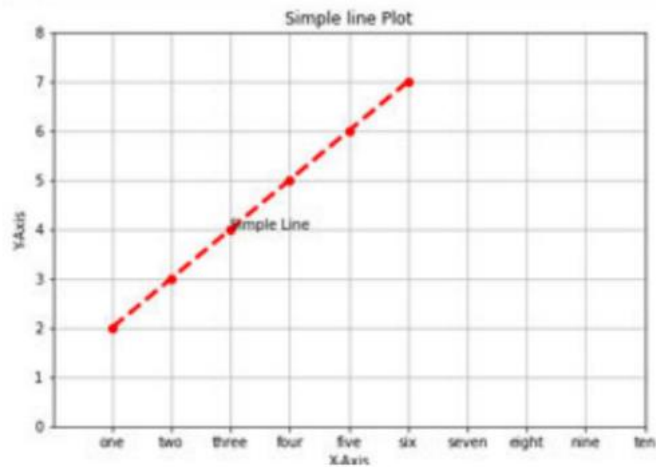
ax.set_xticks([1,2,3,4,5,6,7,8,9,10])
ax.set_xticklabels(['one', 'two', 'three', 'four', 'five', 'six', 'seven',
'eight', 'nine', 'ten'])

plt.text(3,4,"Simple Line")

ax.plot(x,y,marker='o', linestyle="--", color='red', linewidth='3')

ax.grid(axis='y')
```

```
plt.show()
```



Ví dụ 2: `set_yticks()` và `set_yticklabels()`

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])
x=[1,2,3,4,5,6]
y=[2,3,4,5,6,7]

ax.set_title("Simple lin Plot", loc='center')

ax.set_xlabel("X-Axis")
ax.set_ylabel("Y-Axis")

ax.set_xlim(0,10)
ax.set_ylim(0,8)

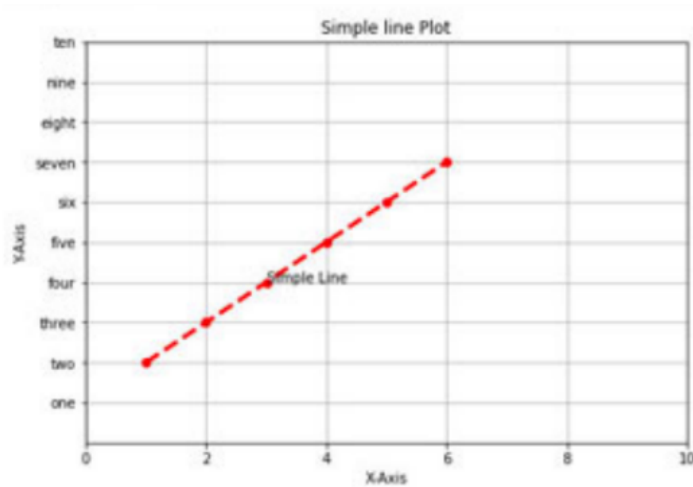
ax.set_yticks([1,2,3,4,5,6,7,8,9,10])
ax.set_yticklabels(['one', 'two', 'three', 'four', 'five', 'six', 'seven',
'eight', 'nine', 'ten'])

plt.text(3,4,"Simple Line")

ax.plot(x,y,marker='o', linestyle="--", color='red', linewidth='3')

ax.grid(axis='y')

plt.show()
```



8. Một số kiểu đồ thị cơ bản trong Matplotlib

Một số kiểu đồ thị như sau:

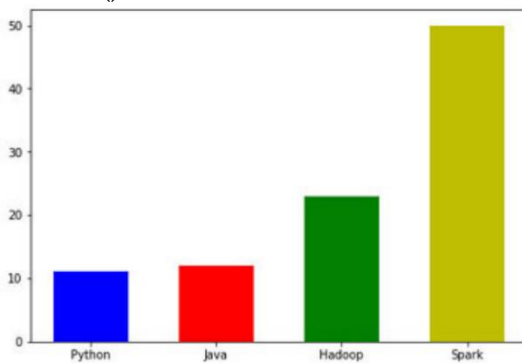
- Biểu đồ cột:

```
import matplotlib.pyplot as plt

course = ['Python', 'Java', 'Hadoop', 'Spark']
students = [11, 12, 23, 50]
fig = plt.figure()
ax = fig.add_axes([0, 0, 1, 1])

ax.bar(course, students, width=0.6, color=['b', 'r', 'g', 'y'])

plt.show()
```



- Biểu đồ tần suất:

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

fig, ax = plt.subplots(1, 1)
x = np.array([90, 91, 98, 99, 100, 105, 107, 178, 160, 140, 122, 89,])
ax.hist(x, bins = [80, 90, 100, 125, 150, 200])

ax.set_title("Emp Salary – Histogram")
```

```
ax.set_xticks([80,90,100,110,120,130,140,150,160,170])
ax.set_xlabel('Annual salary in $')
ax.set_ylabel('no. of Employees')

plt.show()
```



- Biểu đồ phân tán:

```
import matplotlib.pyplot as plt

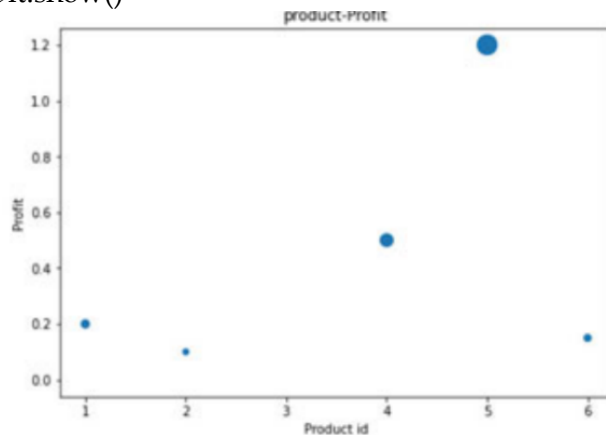
fig = plt.figure()
ax = fig.add_subplot(1,1,1)

profit = [-2,-1,0,0,5,1.2,.15]
product = [1,2,3,4,5,6]

Scale = [p*200 for p in profit]

ax.scatter(x=product, y=profit, s=Scale)
ax.set_title("product-Profit")

ax.set_xlabel("Product id")
ax.set_ylabel("Profit")
plt.show()
```



- Biểu đồ tròn:

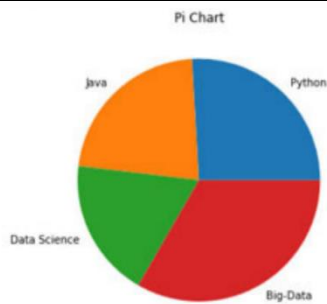
```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])

no_of_students = np.array([35,30,25,45])
courses = ['Python', 'Java', 'Data Science', 'Big Data']
ax.pie(no_of_students, labels=courses)

ax.set_title("Pie chart")

plt.show()
```



9. Xuất đồ thị vào tệp

Matplotlib cho phép xuất đồ thị vào một tệp pdf và một tệp hình ảnh jpeg. Ví dụ:

- Xuất đồ thị vào tệp jpeg:

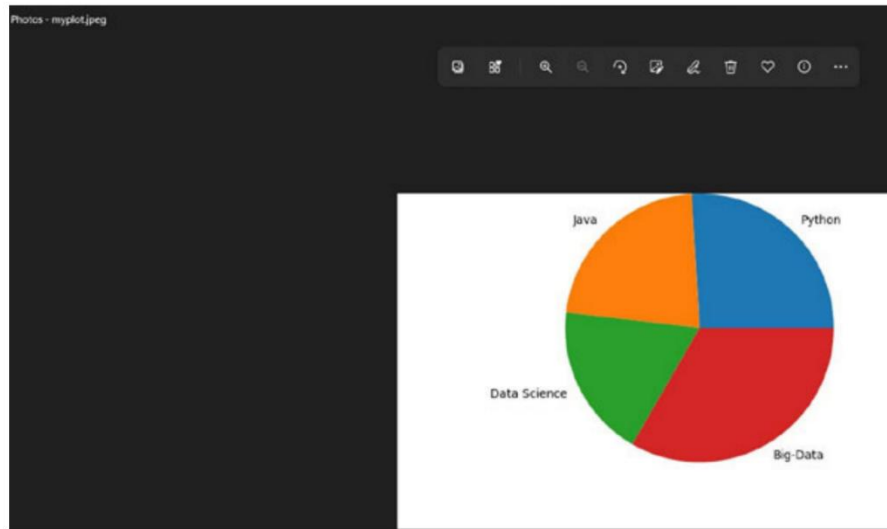
Lưu đồ thị dưới dạng tệp pdf và tệp hình ảnh jpeg. Matplotlib có hàm **savefig()**:

```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])

no_of_students = np.array([35,30,25,45])
courses = ['Python', 'Java', 'Data Science', 'Big Data']
ax.pie(no_of_students, labels=courses)
ax.set_title("Pie chart")

fig.savefig("data\out\myplot.jpeg",dpi=200)
```



- Xuất đồ thị vào một tệp pdf;

```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_axes([0.1,0.1,1,1])

no_of_students = np.array([35,30,25,45])
courses = ['Python', 'Java', 'Data Science', 'Big Data']
ax.pie(no_of_students, labels=courses)
ax.set_title("Pie chart")

mypdf = PdfPages('data\out\myplotpdf.pdf')
mypdf.savefig(fig)
mypdf.close()
```

