

# Python cho Khoa học dữ liệu

## Bài 20: Tổng quan về Thư viện Matplotlib

Hà Minh Tuấn  
hmtuan@hcmus.edu.vn

Khoa Toán - Tin học  
Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

Ngày 4 tháng 9 năm 2025

# Nội dung bài học

## 1. Thư viện Matplotlib

# Giới thiệu Matplotlib

Matplotlib là thư viện trực quan hóa dữ liệu phổ biến nhất trong Python. Nó cho phép tạo ra nhiều loại biểu đồ từ cơ bản đến phức tạp, hỗ trợ trình bày khoa học.

- Được phát triển từ năm 2003
- Thường dùng kết hợp với NumPy, Pandas
- Hỗ trợ nhiều kiểu biểu đồ: line, bar, scatter, histogram,...

# Cài đặt Matplotlib

Để cài đặt Matplotlib:

```
pip install matplotlib
```

dùng trong Python:

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

## Vẽ đồ thị đường cơ bản

```
import matplotlib.pyplot as plt  
x = [1, 2, 3, 4]  
y = [1, 4, 9, 16]  
plt.plot(x, y)  
plt.show()
```

Hiển thị đồ thị đường với 4 điểm dữ liệu.

## Thêm tiêu đề và nhãn trục

```
plt.plot(x, y)
plt.title("Đồ thị hàm số y = x^2")
plt.xlabel("Trục X")
plt.ylabel("Trục Y")
plt.show()
```

Hiển thị đồ thị có tiêu đề và nhãn trục.

## Biểu đồ Scatter

```
x = [5,7,8,7,2,17,2,9,4,11]  
y = [99,86,87,88,100,86,103,87,94,78]  
plt.scatter(x, y)  
plt.show()
```

Hiển thị biểu đồ phân tán (scatter plot).

# Biểu đồ Cột (Bar chart)

```
x = ["A", "B", "C", "D"]  
y = [3, 7, 1, 8]  
plt.bar(x, y)  
plt.show()
```

Hiển thị biểu đồ cột.

# Biểu đồ Cột ngang

```
plt.barh(x, y)  
plt.show()
```

Hiển thị biểu đồ cột theo chiều ngang.

# Biểu đồ Histogram

```
import numpy as np  
data = np.random.randn(1000)  
plt.hist(data, bins=30)  
plt.show()
```

Hiển thị histogram của dữ liệu ngẫu nhiên.

# Biểu đồ Tròn (Pie chart)

```
sizes = [15, 30, 45, 10]
labels = ["A", "B", "C", "D"]
plt.pie(sizes, labels=labels)
plt.show()
```

Hiển thị biểu đồ tròn với nhãn.

## Tùy chỉnh màu sắc và kiểu đường

```
plt.plot(x, y, color="red", linestyle="--", marker="o")  
plt.show()
```

Hiển thị đường màu đỏ, gạch đứt, có dấu tròn.

## Nhiều đường trên cùng đồ thị

```
t = [0,1,2,3,4]
plt.plot(t, [i**2 for i in t], label="y=x^2")
plt.plot(t, [i**3 for i in t], label="y=x^3")
plt.legend()
plt.show()
```

Hiển thị nhiều đường với chú thích.

## Subplots (nhiều biểu đồ)

```
fig, axs = plt.subplots(2)
axs[0].plot([1,2,3],[1,2,3])
axs[1].plot([1,2,3],[1,4,9])
plt.show()
```

Hiển thị 2 biểu đồ trong cùng figure.

## Kích thước Figure

```
plt.figure(figsize=(8,5))  
plt.plot(x,y)  
plt.show()
```

Đặt kích thước figure (8x5 inch).

## Chú thích (Annotation)

```
plt.plot(x,y)
plt.annotate("Điểm đặc biệt", xy=(2,4), xytext=(3,10),
             arrowprops=dict(facecolor='black', shrink=0.05))
plt.show()
```

Thêm chú thích vào đồ thị.

## Lưới (Grid)

```
plt.plot(x,y)
plt.grid(True)
plt.show()
```

Hiển thị lưới trên đồ thị.

## Lưu biểu đồ ra file

```
plt.plot(x,y)  
plt.savefig("do_thi.png")
```

Lưu biểu đồ thành file PNG.

# Biểu đồ 3D (Line)

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
z = [1,2,3,4,5]
x = [1,2,3,4,5]
y = [i**2 for i in x]
ax.plot(x, y, z)
plt.show()
```

Hiển thị đồ thị đường 3D.

## Biểu đồ 3D (Scatter)

```
ax.scatter(x, y, z)  
plt.show()
```

Hiển thị scatter plot trong không gian 3D.

## Các chủ đề nâng cao

- Tùy chỉnh style với plt.style.use()
- Kết hợp Matplotlib với Pandas DataFrame
- Biểu đồ heatmap (dùng imshow)
- Hoạt hình với FuncAnimation
- Kết hợp với seaborn để nâng cao trực quan hóa

# Kết luận về Matplotlib

- Matplotlib là công cụ cơ bản và mạnh mẽ để trực quan hóa dữ liệu
- Thích hợp cho cả phân tích dữ liệu và trình bày khoa học
- Nắm vững Matplotlib giúp dễ dàng học tiếp Seaborn, Plotly