

Chương 5: Trực quan hóa Dữ liệu với Matplotlib

Contents

- Trực quan hóa Dữ liệu với Matplotlib
- 5.1 Giới thiệu về Matplotlib
- 5.2 Biểu đồ cơ bản với Matplotlib
- 5.3 Tùy chỉnh biểu đồ
- 5.4 Sử dụng Object-Oriented API
- 5.5 Trực quan hóa nhiều biểu đồ
- 5.6 Lưu biểu đồ
- 5.7 Thực hành
- 5.8 Tổng kết chương 5

Trực quan hóa Dữ liệu với Matplotlib

Matplotlib là một thư viện mạnh mẽ và phổ biến để trực quan hóa dữ liệu trong Python. Nó cung cấp các công cụ linh hoạt để tạo ra các loại biểu đồ khác nhau như biểu đồ đường, cột, tán xạ, và nhiều loại khác. Trong chương này, bạn sẽ học cách sử dụng Matplotlib để trực quan hóa dữ liệu.

5.1 Giới thiệu về Matplotlib

5.1.1 Cài đặt Matplotlib

Để cài đặt Matplotlib, sử dụng lệnh:

```
pip install matplotlib
```

5.1.2 Nhập Matplotlib và cấu trúc cơ bản

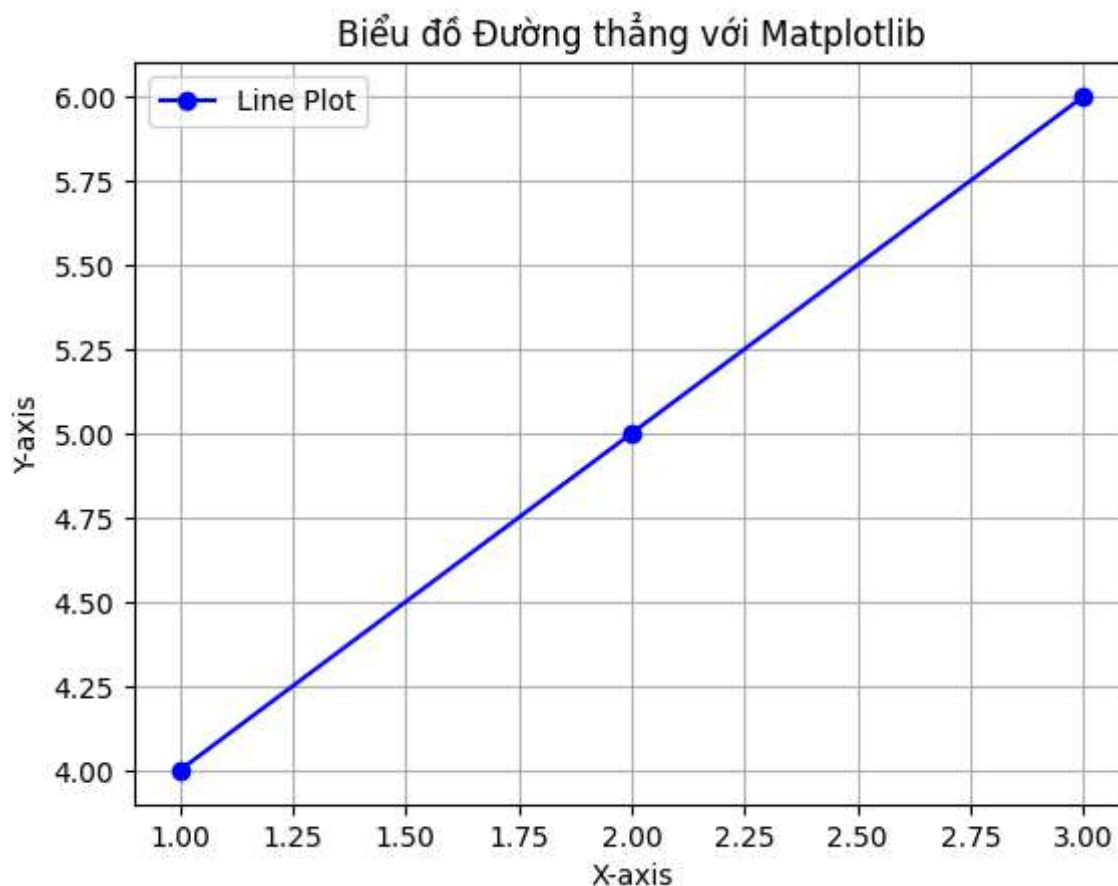
- Matplotlib được tổ chức theo hai cách sử dụng:

Pyplot API: Giao diện đơn giản và phổ biến. **Object-Oriented API:** Linh hoạt hơn, cho phép tùy chỉnh sâu. Biểu đồ cơ bản đường thẳngthẳng

```
import matplotlib.pyplot as plt

# Dữ liệu
x = [1, 2, 3]
y = [4, 5, 6]

# Vẽ biểu đồ
plt.plot(x, y, marker='o', linestyle='-', color='blue', label='Line Plot')
plt.title('Biểu đồ Đường thẳng với Matplotlib')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend()
plt.grid(True)
plt.show()
```

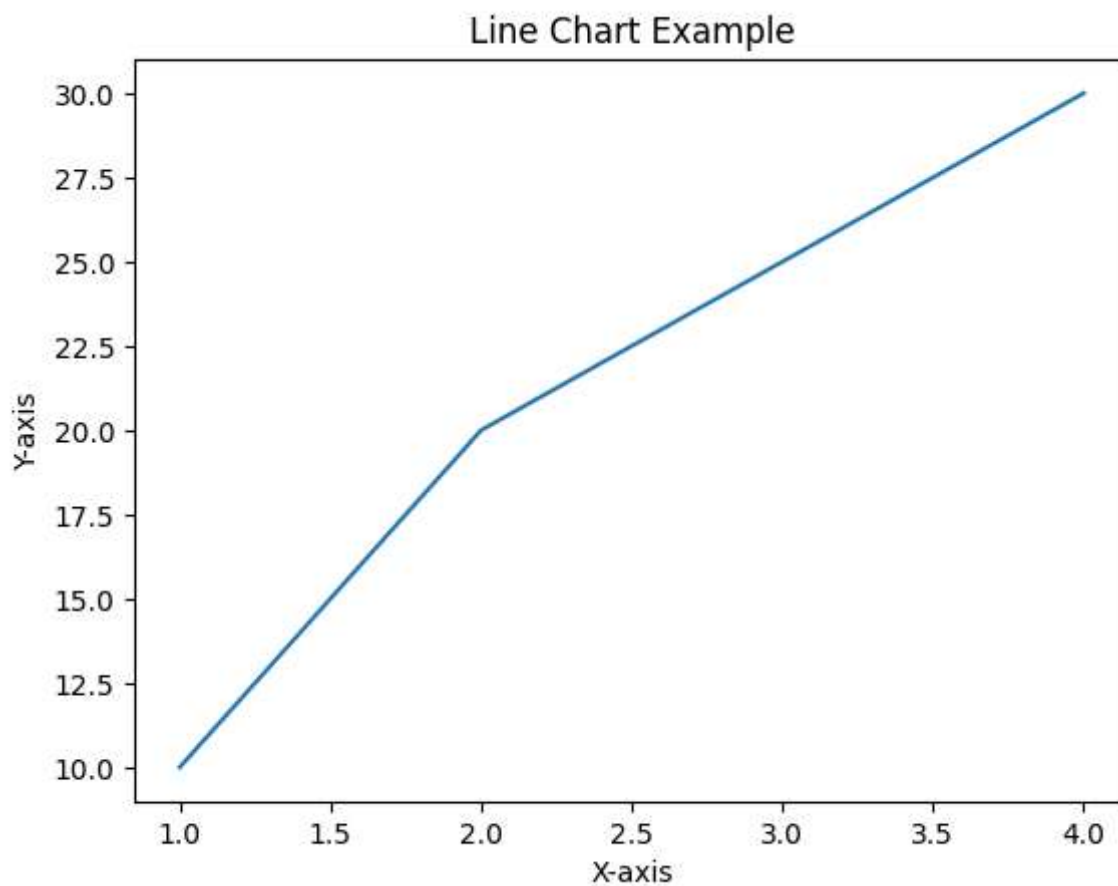


Chú thích: Thêm dấu chấm tròn (marker='o') tại các điểm dữ liệu.

Ví dụ cơ bản với Pyplot:

```
import matplotlib.pyplot as plt

# Tạo biểu đồ đường
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
plt.plot(x, y)
plt.title("Line Chart Example")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.show()
```



Chú thích:

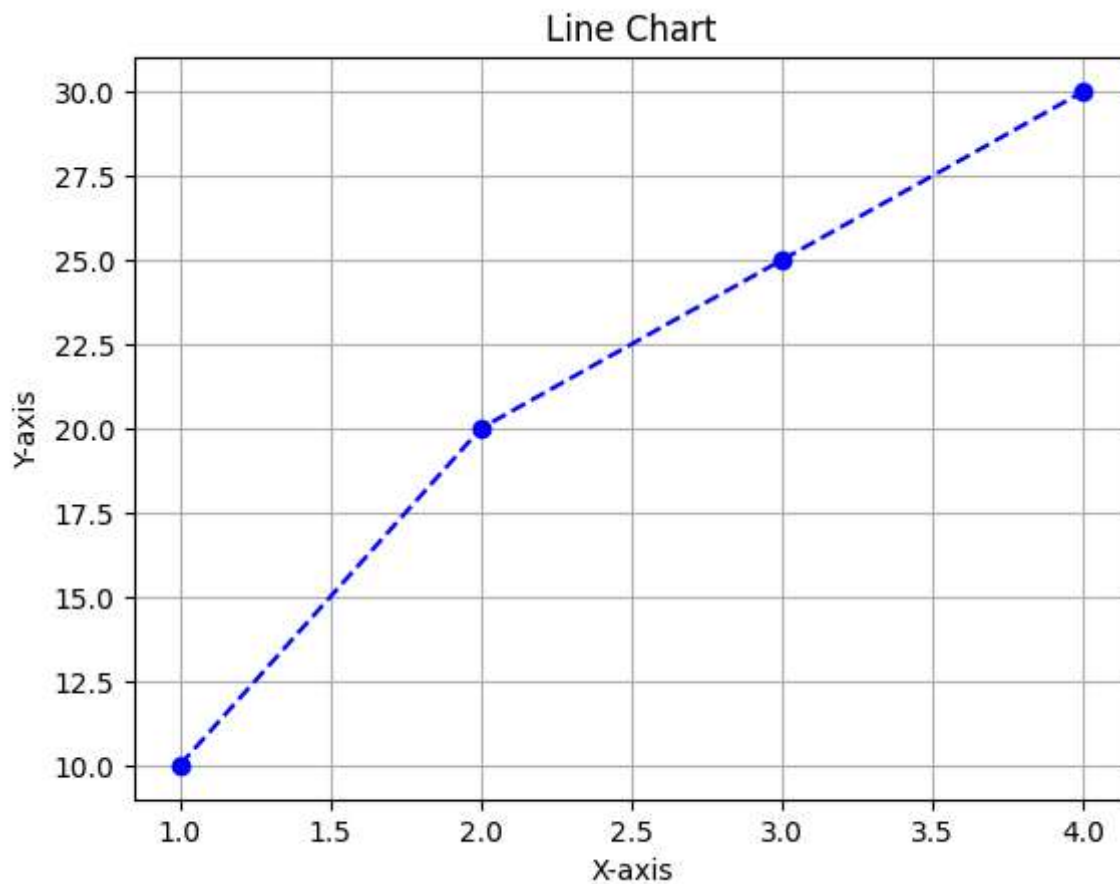
- `plt.plot(x, y)` vẽ một đường nối giữa các điểm (x, y).
- `plt.title` và `plt.xlabel`, `plt.ylabel` thêm tiêu đề và nhãn trục.

5.2 Biểu đồ cơ bản với Matplotlib

5.2.1 Biểu đồ đường (Line Chart)

Dùng để hiển thị xu hướng theo thời gian hoặc tuần tự.

```
plt.plot(x, y, color='blue', linestyle='--', marker='o')
plt.title("Line Chart")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.grid(True)
plt.show()
```

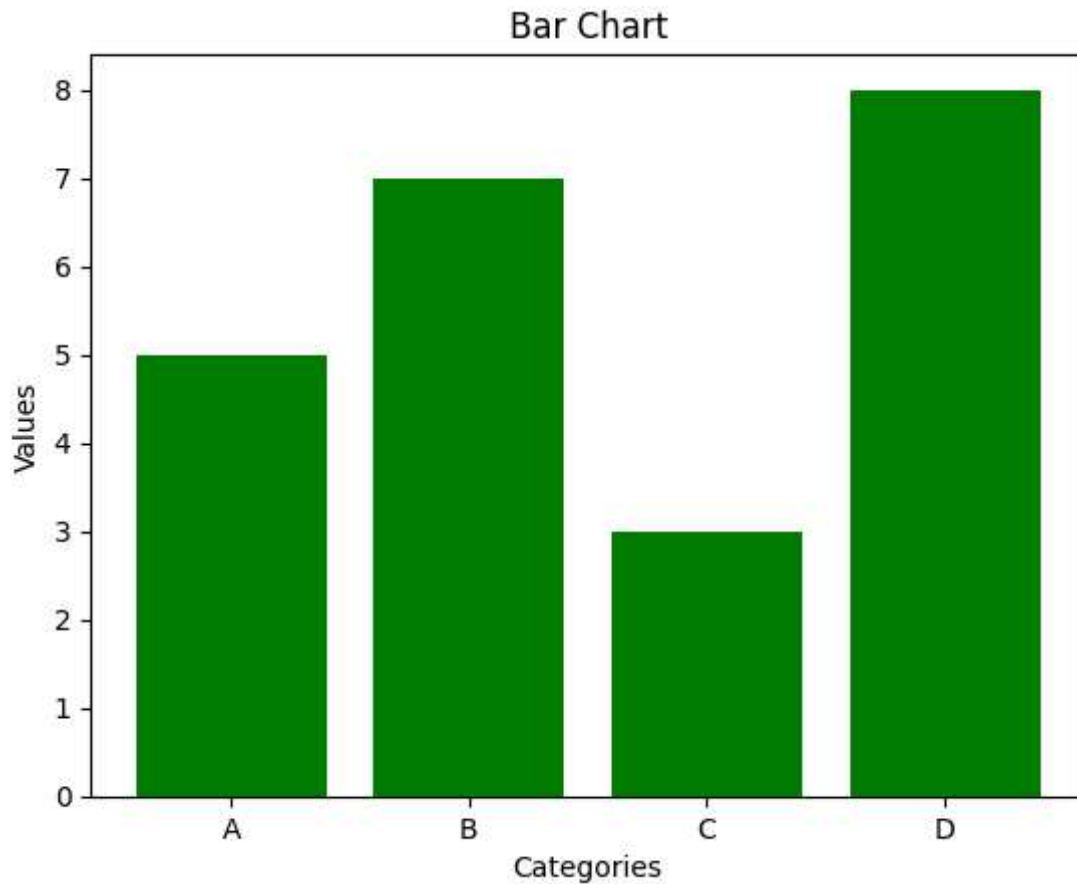


Chú thích:

- `color='blue'`: Đặt màu xanh cho đường.
- `linestyle='--'`: Kẻ đường nét đứt.
- `marker='o'`: Thêm các điểm đánh dấu (marker) trên mỗi điểm dữ liệu.
- `plt.grid(True)`: Thêm lưới để biểu đồ dễ đọc hơn.

5.2.2 Biểu đồ cột (Bar Chart) Dùng để so sánh các nhóm.

```
categories = ['A', 'B', 'C', 'D']  
values = [5, 7, 3, 8]  
plt.bar(categories, values, color='green')  
plt.title("Bar Chart")  
plt.xlabel("Categories")  
plt.ylabel("Values")  
plt.show()
```

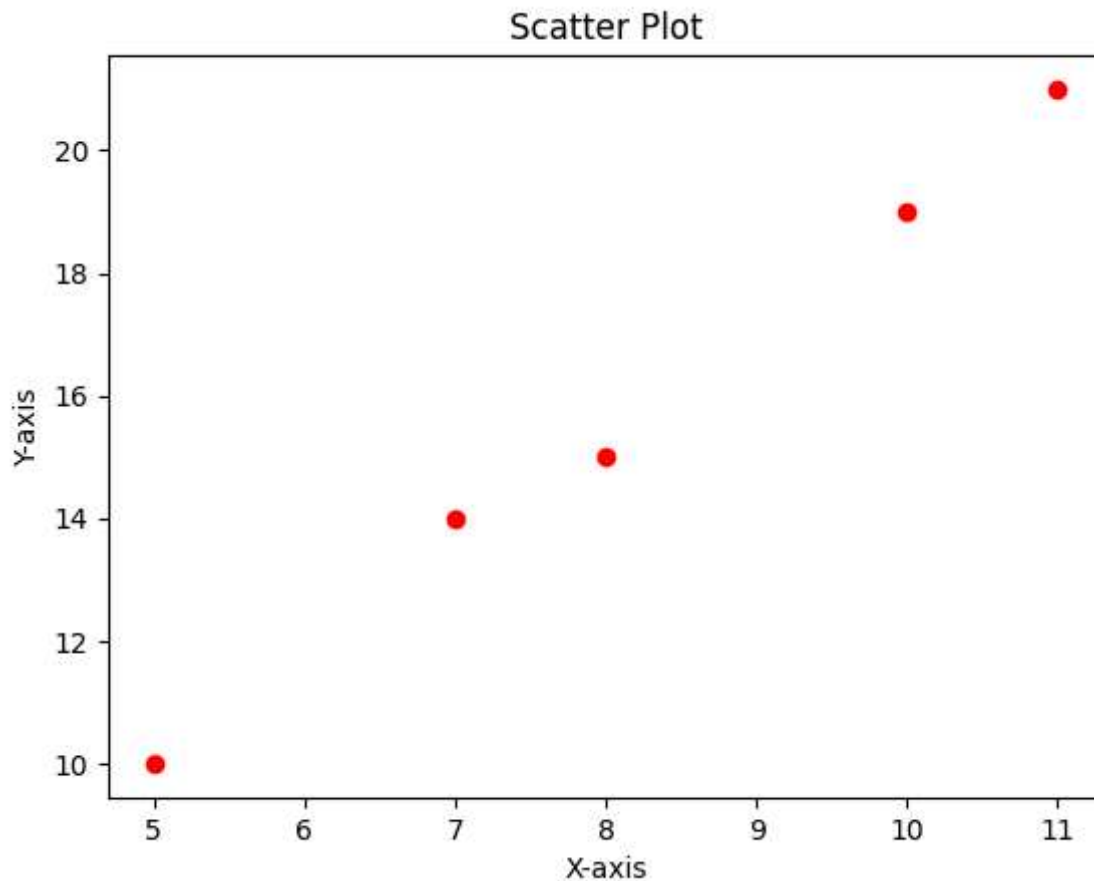


Chú thích:

- `plt.bar` vẽ các cột tương ứng với `categories` và giá trị `values`.
- `color='green'` đặt màu xanh lá cây cho các cột.

5.2.3 Biểu đồ tán xạ (Scatter Plot) Dùng để hiển thị mối quan hệ giữa hai biến.

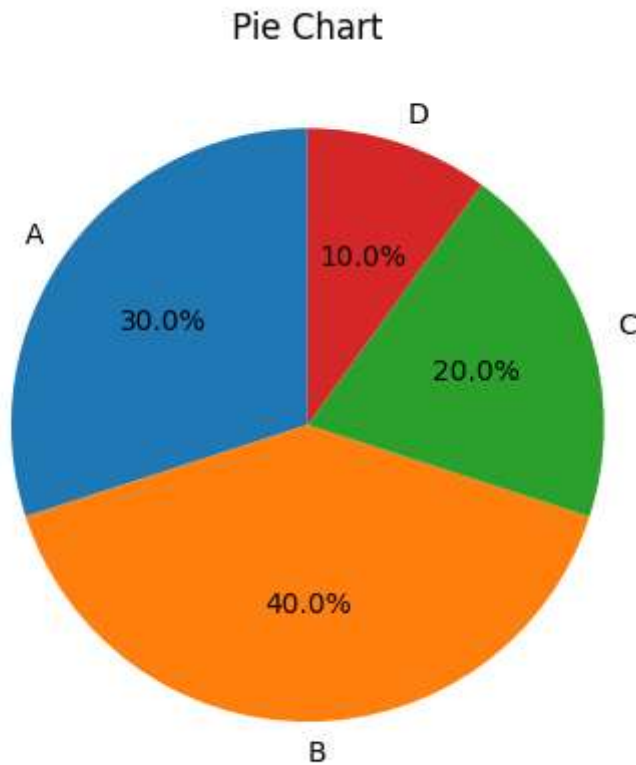
```
x = [5, 7, 8, 10, 11]  
y = [10, 14, 15, 19, 21]  
plt.scatter(x, y, color='red')  
plt.title("Scatter Plot")  
plt.xlabel("X-axis")  
plt.ylabel("Y-axis")  
plt.show()
```

**Chú thích:**

- plt.scatter vẽ các điểm rời rạc tương ứng với dữ liệu.
- Dùng khi bạn muốn phân tích mối quan hệ giữa hai biến (x, y).

5.2.4 Biểu đồ hình tròn (Pie Chart) Dùng để biểu diễn tỷ lệ phần trăm của các phần tử.

```
labels = ['A', 'B', 'C', 'D']
sizes = [30, 40, 20, 10]
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
plt.title("Pie Chart")
plt.show()
```

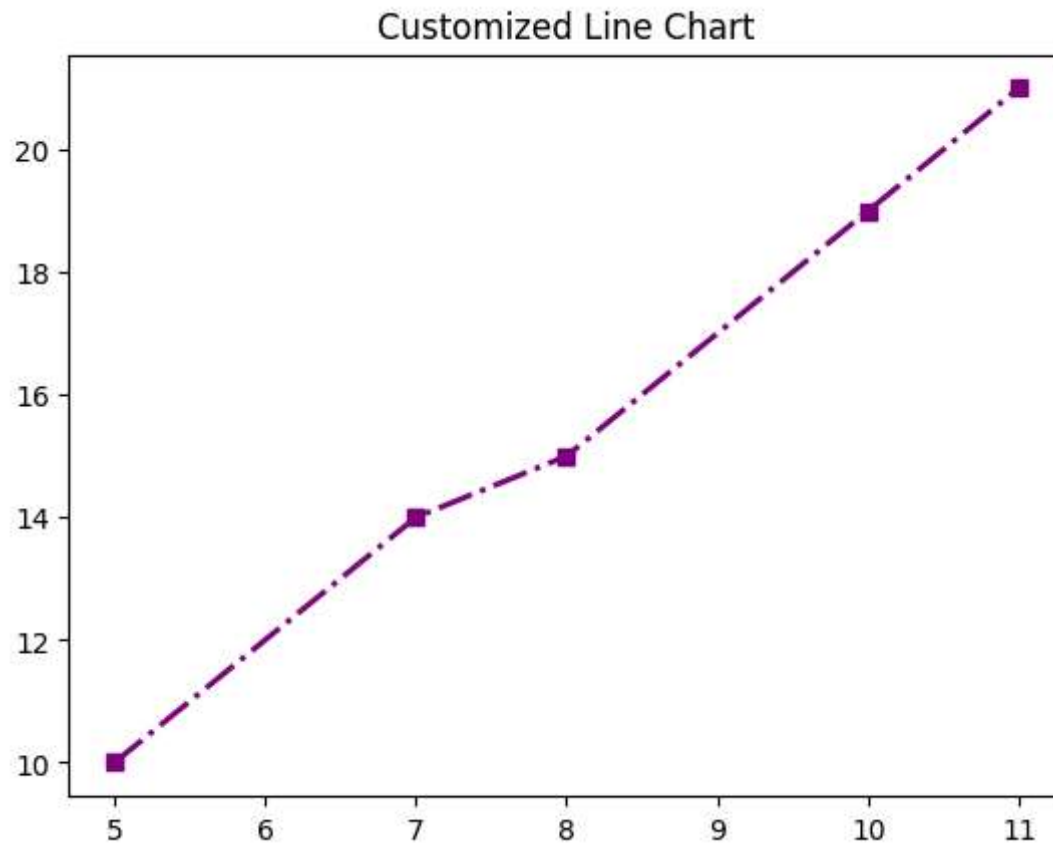
**Chú thích:**

- `autopct='%1.1f%%'`: Hiển thị tỷ lệ phần trăm trên từng phần tử.
- `startangle=90`: Xoay biểu đồ để bắt đầu từ góc 90 độ.

5.3 Tùy chỉnh biểu đồ

5.3.1 Tùy chỉnh màu sắc và kiểu dáng

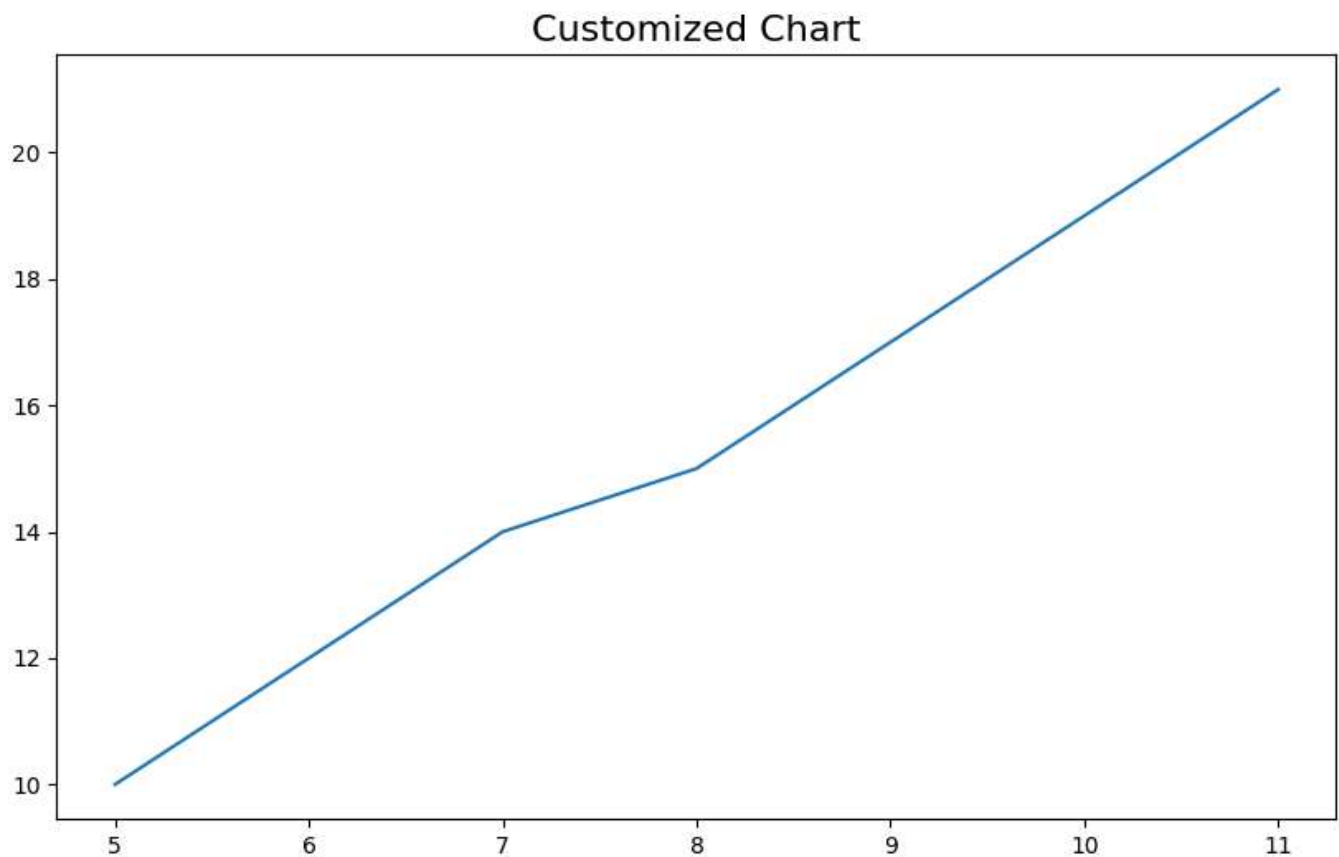
```
plt.plot(x, y, color='purple', linestyle='-.', marker='s', linewidth=2)
plt.title("Customized Line Chart")
plt.show()
```

**Chú thích:**

- `linewidth=2`: Đặt độ dày cho đường.
- `color`, `linestyle`, và `marker` cho phép tùy chỉnh màu sắc, kiểu đường, và kiểu điểm đánh dấu.

5.3.2 Tùy chỉnh kích thước biểu đồ

```
plt.figure(figsize=(10, 6))  
plt.plot(x, y)  
plt.title("Customized Chart", fontsize=16)  
plt.show()
```

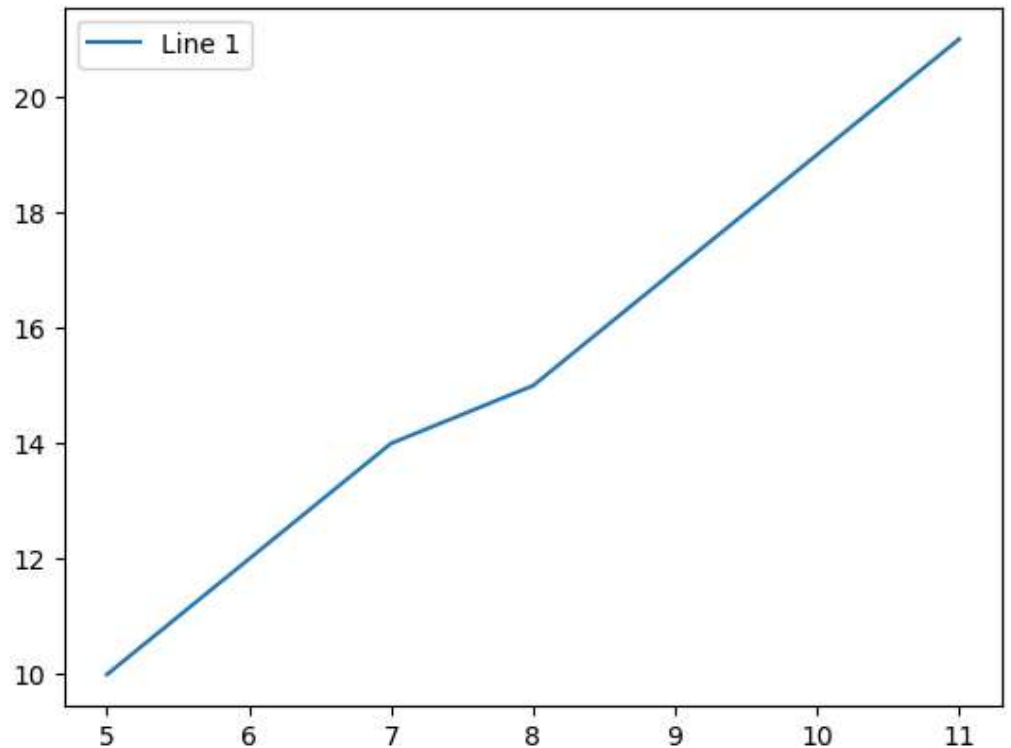

**Chú thích:**

- `figsize=(10, 6)`: Đặt kích thước biểu đồ (chiều ngang 10, chiều cao 6).
- `fontsize=16`: Đặt kích cỡ cho tiêu đề.

5.3.3 Thêm chú thích và văn bản

```
plt.plot(x, y)
plt.text(2, 20, 'Point of Interest', fontsize=12, color='blue')
plt.legend(['Line 1'], loc='upper left')
plt.show()
```

Point of Interest



Chú thích:

- `plt.text`: Thêm văn bản tại vị trí chỉ định.
- `plt.legend`: Hiển thị chú giải ở góc trên bên trái.

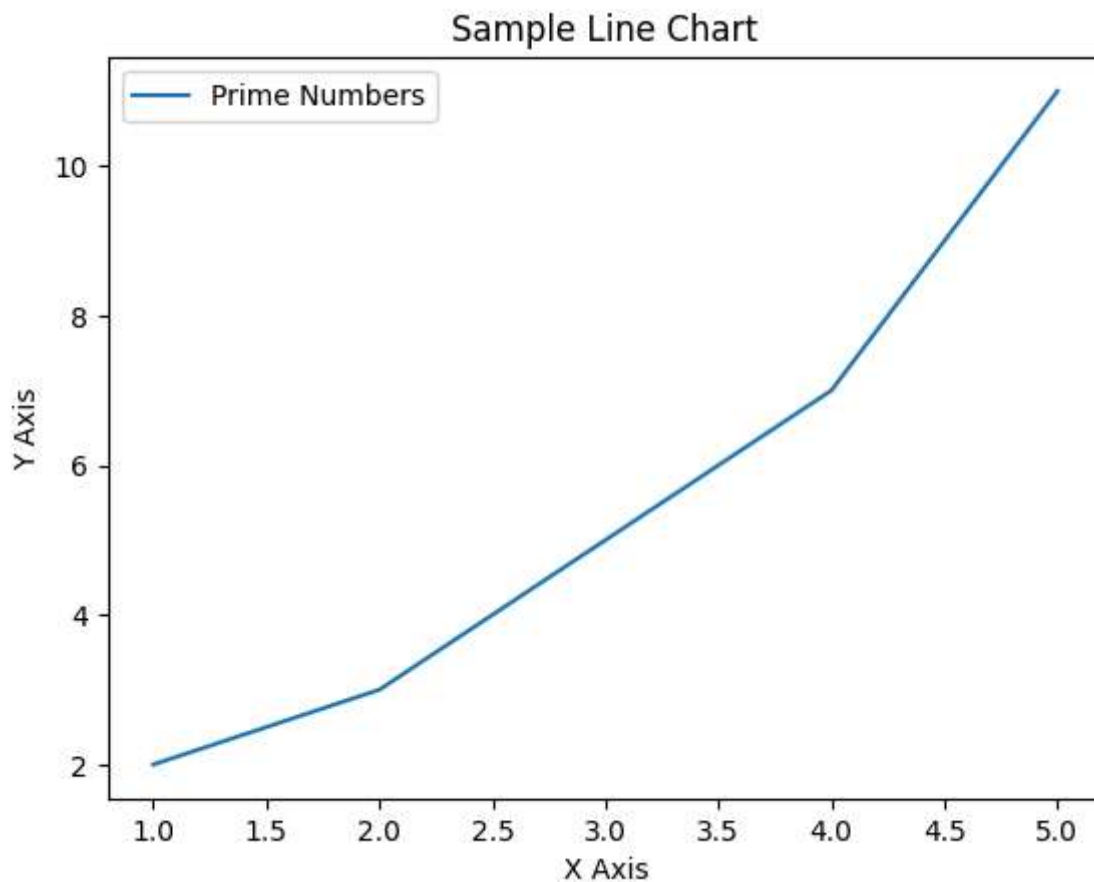
5.4 Sử dụng Object-Oriented API

5.4.1 Biểu đồ đường với Matplotlib

- Đây là biểu đồ đường API

```
import matplotlib.pyplot as plt

class LineChart:
    def __init__(self, title, x_label, y_label):
        self.fig, self.ax = plt.subplots()
        self.ax.set_title(title)
        self.ax.set_xlabel(x_label)
        self.ax.set_ylabel(y_label)
    def plot(self, x_data, y_data, label):
        self.ax.plot(x_data, y_data, label=label)
        self.ax.legend()
    def show(self):
        plt.show()
if __name__ == "__main__":
    # Sample data
    x = [1, 2, 3, 4, 5]
    y = [2, 3, 5, 7, 11]
    chart = LineChart("Sample Line Chart", "X Axis", "Y Axis")
    chart.plot(x, y, label="Prime Numbers")
    chart.show()
```



Chú thích:

Lớp LineChart : Lớp này bao gồm chức năng tạo biểu đồ đường.

- `__init__` Phương pháp : Khởi tạo hình và trục, đặt tiêu đề và nhãn cho các trục.
- `plot` Phương pháp : Lấy dữ liệu x và y cùng với nhãn để vẽ đường thẳng trên các trục.
- `show` Phương pháp : Hiển thị biểu đồ.

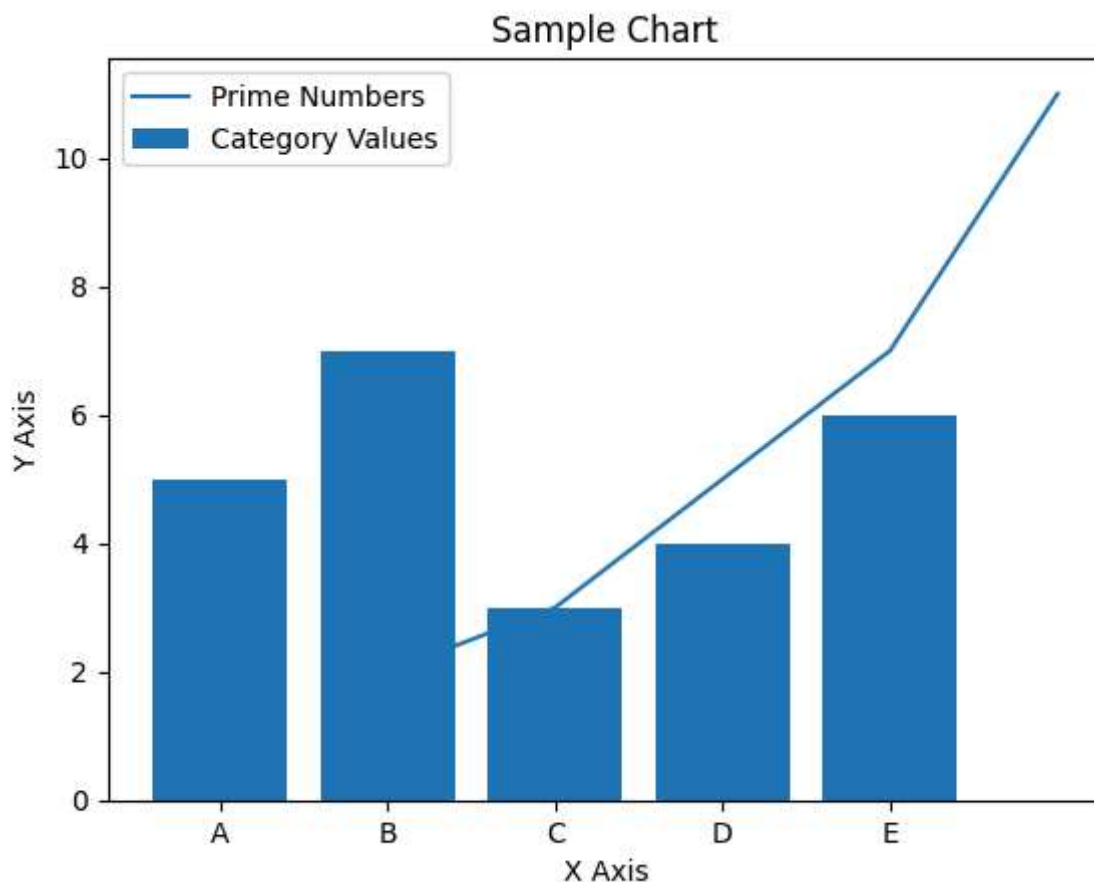
5.4.2 Thêm phương pháp biểu đồ thanh

```
import matplotlib.pyplot as plt

class Chart:
    def __init__(self, title, x_label, y_label):
        self.fig, self.ax = plt.subplots()
        self.ax.set_title(title)
        self.ax.set_xlabel(x_label)
        self.ax.set_ylabel(y_label)
    def plot_line(self, x_data, y_data, label):
        self.ax.plot(x_data, y_data, label=label)
        self.ax.legend()
    def plot_bar(self, x_data, y_data, label):
        self.ax.bar(x_data, y_data, label=label)
        self.ax.legend()

    def show(self):
        plt.show()

if __name__ == "__main__":
    # Sample data for line chart
    x_line = [1, 2, 3, 4, 5]
    y_line = [2, 3, 5, 7, 11]
    x_bar = ['A', 'B', 'C', 'D', 'E']
    y_bar = [5, 7, 3, 4, 6]
    chart = Chart("Sample Chart", "X Axis", "Y Axis")
    chart.plot_line(x_line, y_line, label="Prime Numbers")
    chart.plot_bar(x_bar, y_bar, label="Category Values")
    chart.show()
```



Chú thích: Lớp biểu đồ : Lớp này hiện hỗ trợ cả biểu đồ đường và biểu đồ thanh.

`plot_bar` Phương pháp : Phương pháp mới này sử dụng dữ liệu x và y cùng với nhãn để tạo biểu đồ thanh trên các trục. Ví dụ sử dụng :

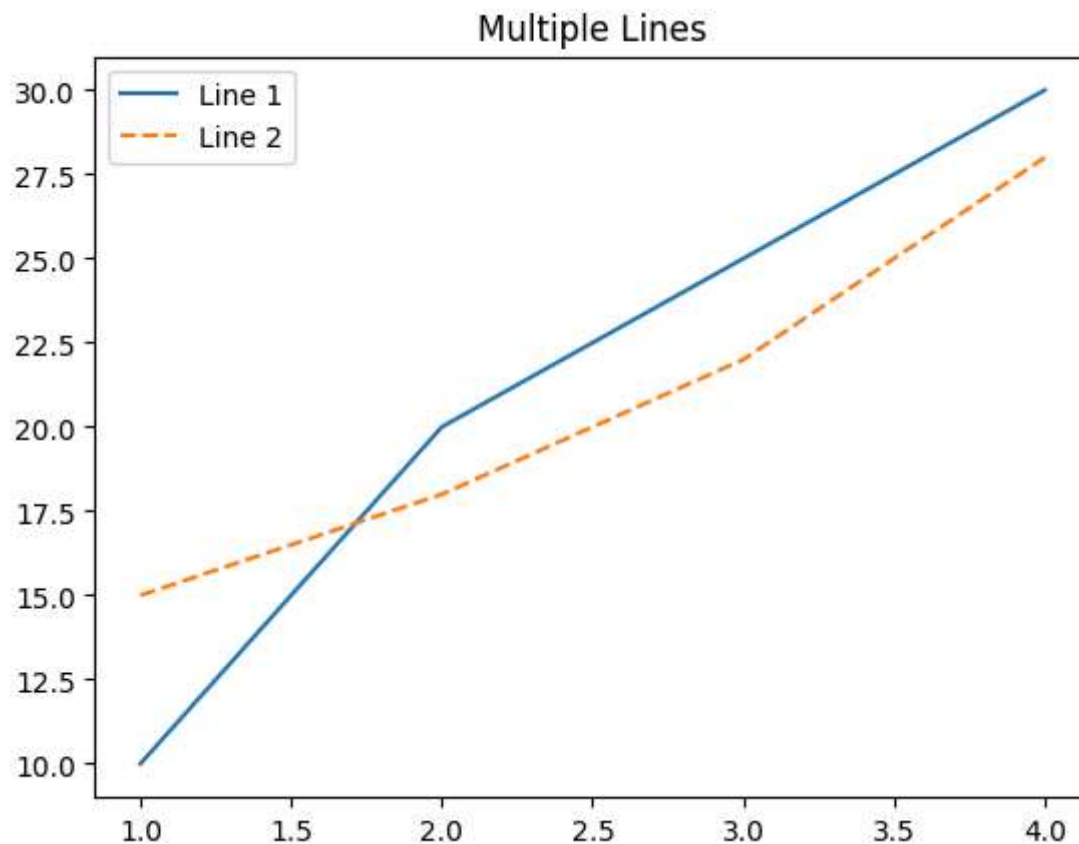
- Dữ liệu mẫu được xác định cho cả biểu đồ đường và biểu đồ thanh.
- Một thể hiện của `Chart` được tạo ra với tiêu đề và nhãn trục.
- Phương pháp này `plot_line` được gọi để thêm dữ liệu đường vào biểu đồ.
- Phương pháp này `plot_bar` được gọi để thêm dữ liệu thanh vào cùng một biểu đồ.
- Cuối cùng, `show` phương thức này được gọi để hiển thị biểu đồ kết hợp.

5.5 Trực quan hóa nhiều biểu đồ

5.5.1 Nhiều biểu đồ trong cùng một hình

```
x = [1, 2, 3, 4] # Chỉ giữ lại 4 giá trị
y1 = [10, 20, 25, 30]
y2 = [15, 18, 22, 28]

plt.plot(x, y1, label='Line 1')
plt.plot(x, y2, label='Line 2', linestyle='--')
plt.title("Multiple Lines")
plt.legend()
plt.show()
```



Chú thích:

- Vẽ nhiều đường trên cùng một biểu đồ, mỗi đường có chú thích riêng.

5.5.2 Tạo bố cục nhiều biểu đồ (Subplots)

```

x = [1, 2, 3, 4]
y = [10, 20, 25, 30]

fig, axs = plt.subplots(2, 2, figsize=(10, 8))
axs[0, 0].plot(x, y, color='blue') # Biểu đồ đường
axs[0, 0].set_title("Plot 1")

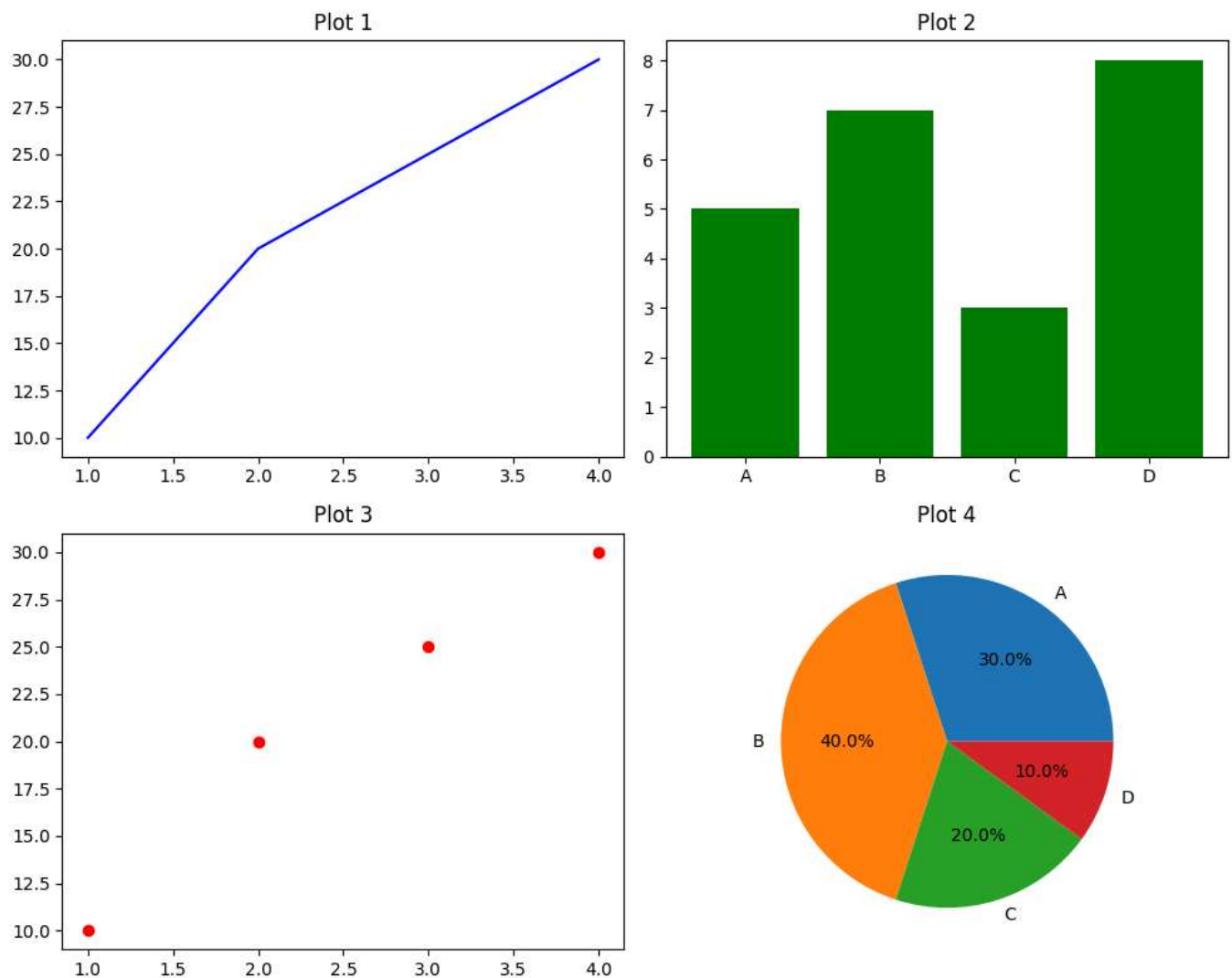
categories = ['A', 'B', 'C', 'D']
values = [5, 7, 3, 8]
axs[0, 1].bar(categories, values, color='green') # Biểu đồ cột
axs[0, 1].set_title("Plot 2")

axs[1, 0].scatter(x, y, color='red') # Biểu đồ tán xạ
axs[1, 0].set_title("Plot 3")

sizes = [30, 40, 20, 10]
labels = ['A', 'B', 'C', 'D']
axs[1, 1].pie(sizes, labels=labels, autopct='%1.1f%%') # Biểu đồ hình tròn
axs[1, 1].set_title("Plot 4")

plt.tight_layout()
plt.show()

```



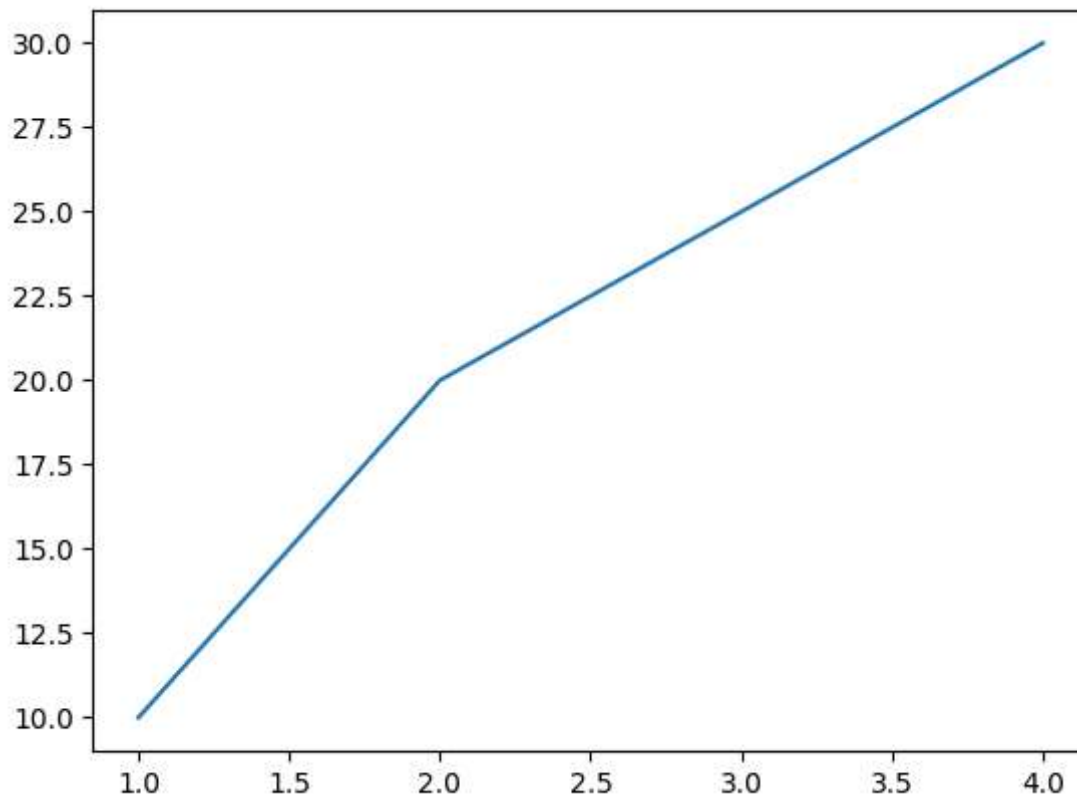
Chú thích:

- `plt.subplots(2, 2)`: Tạo một bố cục 2x2 chứa 4 biểu đồ.

5.6 Lưu biểu đồ

Matplotlib cho phép lưu biểu đồ dưới nhiều định dạng.

```
plt.plot(x, y)
plt.savefig("line_chart.png", dpi=300, bbox_inches='tight')
plt.show()
```

**Chú thích:**

- `dpi=300`: Tăng độ phân giải cho hình ảnh.
- `bbox_inches='tight'`: Đảm bảo hình ảnh không bị cắt khi lưu.

5.7 Thực hành

- **Bài tập 1:** Sử dụng dữ liệu doanh số bán hàng hàng tháng để tạo biểu đồ đường và biểu đồ cột.
- **Bài tập 2:** Dùng dữ liệu về số lượng học sinh theo từng khoa để tạo biểu đồ hình tròn.
- **Bài tập 3:** Tạo một bố cục 2x2, hiển thị 4 biểu đồ khác nhau (đường, cột, tán xạ, hình tròn) dựa trên dữ liệu bạn tự tạo.

5.8 Tổng kết chương 5

Nội dung: Trong Chương 5 đã cung cấp nền tảng vững chắc về cách sử dụng Matplotlib để trực quan hóa dữ liệu. Trong chương 6 chúng ta sẽ tiếp tục với Seaborn, một thư viện nâng cao được xây dựng trên Matplotlib.