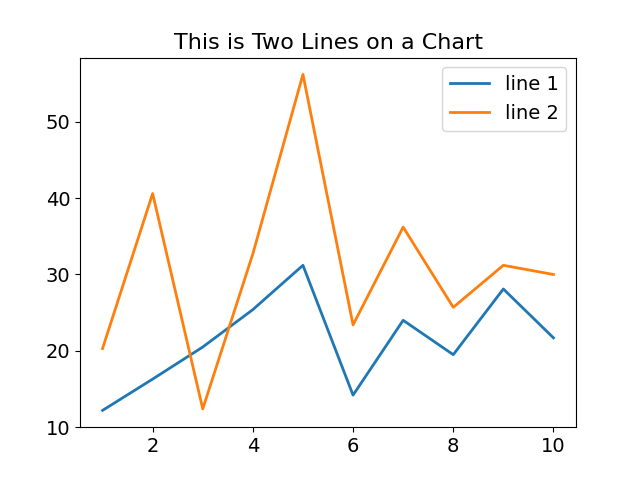
## **Kết hợp nhiều biểu đồ cùng loại**

### Giới thiệu

Trong thực tế, đôi khi chúng ta cần so sánh giá trị trên nhiều bộ dữ liệu có chung thuộc tính nhưng được đo đạc, trích xuất bởi các kĩ thuật khác nhau hay trên các khoảng thời gian khác nhau. Trực quan hóa các bộ dữ liệu này trên cùng một hình, hay trên cùng một biểu đồ sẽ giúp chúng ta dễ dàng nắm bắt thông tin so sánh giữa các bộ dữ liệu, tăng khả năng truyền tải thông tin.

### Nhiều biểu đồ cùng loại trong cùng một khung nhìn



* Nếu từ ba biểu đồ cùng loại trở lên: các bộ dữ liệu phải có cùng miền xác định trục X và Y. Hơn nữa, các bộ dữ liệu phải cùng mô tả chung thuộc tính để tránh đưa ra thông tin giả mạo.
* Nếu có hai biểu đồ cùng loại: hai bộ dữ liệu phải có cùng miền xác định trục X. Nếu miền xác định hay tỉ lệ trục Y của hai bộ dữ liệu khác nhau thì cần sinh thêm trục phụ bên phải.
* Có thể dùng để kiểm tra tính quy luật, tính đồng nhất giữa các bộ dữ liệu.

#### Ý nghĩa sử dụng

* Trực quan các bộ dữ liệu cùng một thuộc tính nhưng mô tả các đối tượng khác nhau, thu thập bởi kĩ thuật khác nhau, trích xuất ở các miền khác nhau,…
* Giúp người xem dễ dàng so sánh giá trị, xu hướng, độ chênh lệch, giá trị thuộc tính giữa các đối tượng, các kĩ thuật, các khoảng thời gian.
* Được sử dụng khi không muốn bố trí quá nhiều biểu đồ trên cùng một Axes.

#### Sử dụng thư viện Matplotlib

* Import thư viện cần thiết

import matplotlib.pyplot as plt  
import pandas as pd  
import numpy as np

* Lấy các bộ dữ liệu:

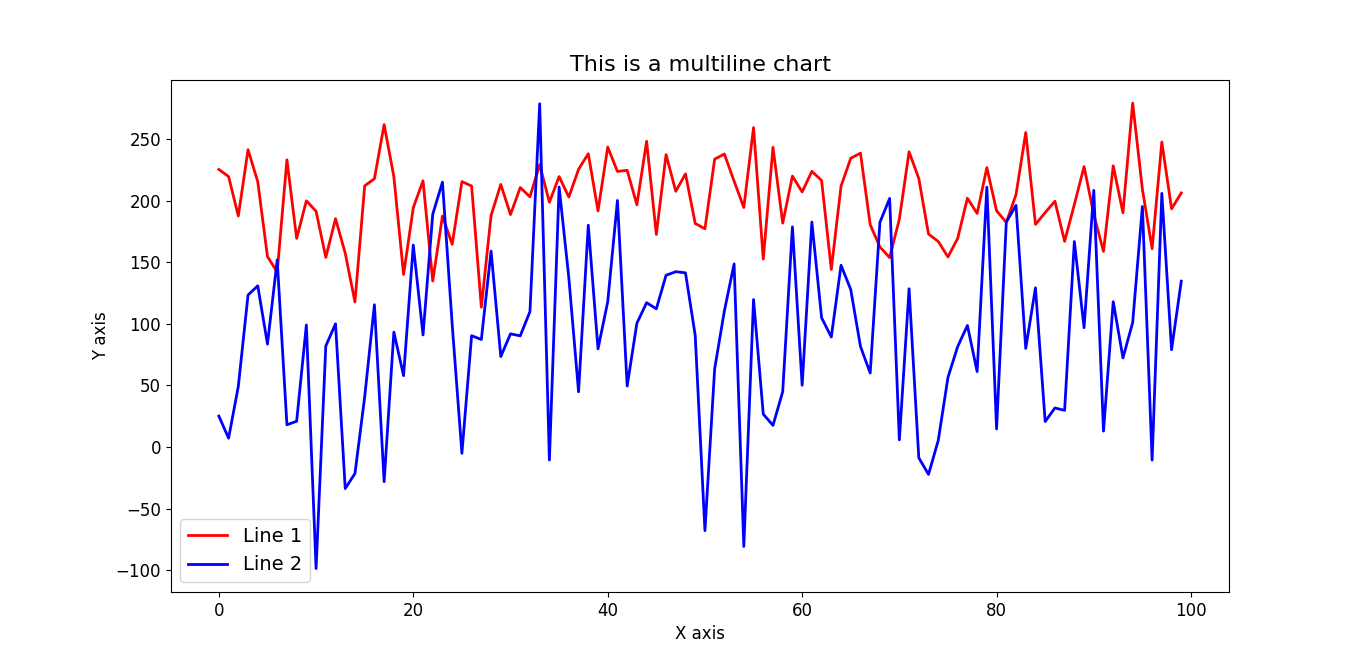
y1 = np.random.normal(200, 30, size = 100)  
y2 = np.random.normal(100,70, size = 100)  
df = pd.DataFrame({'y1': y1, 'y2': y2, 'index': pd.date\_range('1/1/2021', periods=100)})

* Vẽ các biểu đồ đường

plt.plot(df.index, df['y1'], linewidth = 2, color = 'r', label = 'Line 1')  
plt.plot(df.index, df['y2'], linewidth = 2, color = 'b', label = 'Line 2')   
plt.legend(fontsize = 14)  
plt.xticks(fontsize = 12, rotation = 'horizontal')  
plt.yticks(fontsize = 12)  
plt.title('This is a multiline chart', fontsize = 16)  
plt.xlabel('X axis', fontsize = 12)  
plt.ylabel('Y axis', fontsize = 12)

plt.show()

Kết quả:



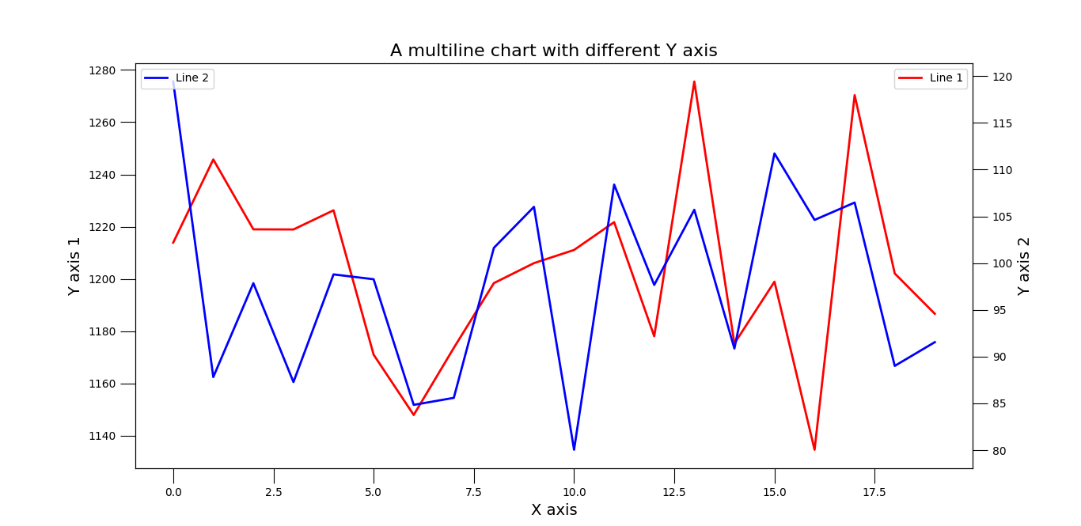
* Trường hợp các bộ dữ liệu không cùng mô tả chung thuộc tính hoặc không có chung độ đo, tỉ lệ trục Y:
  + Chỉ vẽ tối đa 2 bộ dữ liệu trên cùng một biểu đồ.
  + Coi mỗi bộ dữ liệu nằm trên một Axes, Axes 2 có cùng X axis với Axes 1. Tạo thêm trục Y phụ bên phải bằng câu lệnh:

axes2 = axes1.twinx()

Ví dụ

y1 = np.random.normal(1200, 30, size = 20)  
y2 = np.random.normal(100, 10, size = 20)  
df = pd.DataFrame({'y1': y1, 'y2': y2, 'index': pd.date\_range('1/1/2021', periods=20)})  
plt.plot(df.index, df['y1'], linewidth = 2, color = 'r', label = 'Line 1')  
axes1 = plt.gca() #  
axes2 = axes1.twinx()  
axes2.plot(df.index, df['y2'], linewidth = 2, color = 'b', label = 'Line 2')  
axes1.set\_xlabel('X axis', fontsize = 14)  
axes1.set\_ylabel('Y axis 1', fontsize = 14)  
axes2.set\_ylabel('Y axis 2', fontsize = 14)  
plt.legend(fontsize = 14)  
plt.xticks(fontsize = 14)  
plt.yticks(fontsize = 14)  
plt.title('A multiline chart with different Y axis', fontsize = 16)  
plt.show()

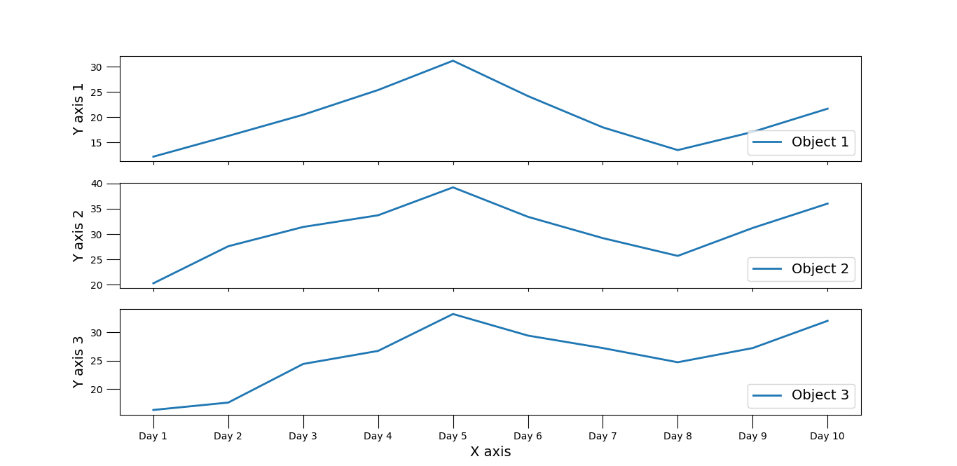
Kết quả:



#### Một số lưu ý khi biểu diễn nhiều biểu đồ trong cùng một khung nhìn

* Dùng marker, color và legend để giúp người xem dễ dàng phân biệt các biểu đồ.
* Không vẽ quá nhiều biểu đồ trong cùng một biểu đồ khiến người xem cảm thấy khó nhìn và phức tạp trong việc xử lý nhiều thông tin cùng lúc.
* Hạn chế sử dụng biểu đồ phải vẽ thêm trục Y phụ do biểu đồ này dễ bị thao túng để truyền đạt thông tin giả mạo.

### Nhiều biểu đồ cùng loại trên cùng một Figure



* Có thể biểu diễn chung các thuộc tính nhưng khác nhau về giới hạn miền giá trị, hay đối tượng mô tả, kĩ thuật thu thập, trích xuất, xử lý dữ liệu.
* Nếu biễu diễn các thuộc tính khác nhau thì cần có chung một mục tiêu trực quan hóa để tránh gây rắc rối cho người xem.

#### Ý nghĩa sử dụng

* Trực quan các bộ dữ liệu cùng một thuộc tính hay khác thuộc tính, nhưng có chung mục tiêu mô tả.
* Giảm bớt độ phức tạp, rắc rối so với việc biểu diễn quá nhiều biểu đồ trong cùng một khung nhìn.
* Giúp người xem dễ dàng so sánh giá trị, độ chênh lệch giữa các bộ dữ liệu cùng mô tả chung thuộc tính.
* Có thể dùng để kiểm tra tính quy luật, tính đồng nhất và tìm xu hướng chung giữa các bộ dữ liệu.

#### Sử dụng thư viện Matplotlib

* Import thư viện cần thiết

import matplotlib.pyplot as plt

* Lấy các bộ dữ liệu:

x1 = ['Day 1','Day 2','Day 3','Day 4','Day 5','Day 6','Day 7','Day 8','Day 9','Day 10']  
y1 = [12.2, 16.3, 20.5, 25.4, 31.2, 24.2, 18, 13.5, 17.1, 21.7]  
y2 = [20.3, 27.6, 31.4, 33.7, 39.2, 33.4, 29.2, 25.7, 31.2, 36]  
y3 = [16.3, 17.6, 24.4, 26.7, 33.2, 29.4, 27.2, 24.7, 27.2, 32]

* Khai báo ma trận các Axes

fig, ax = plt.subplots(r, c)

trong đó:

r: số hàng

c: số cột

ví dụ:

fig, ax = plt.subplots(3, 1)

* Vẽ các biểu đồ trên mỗi Axes

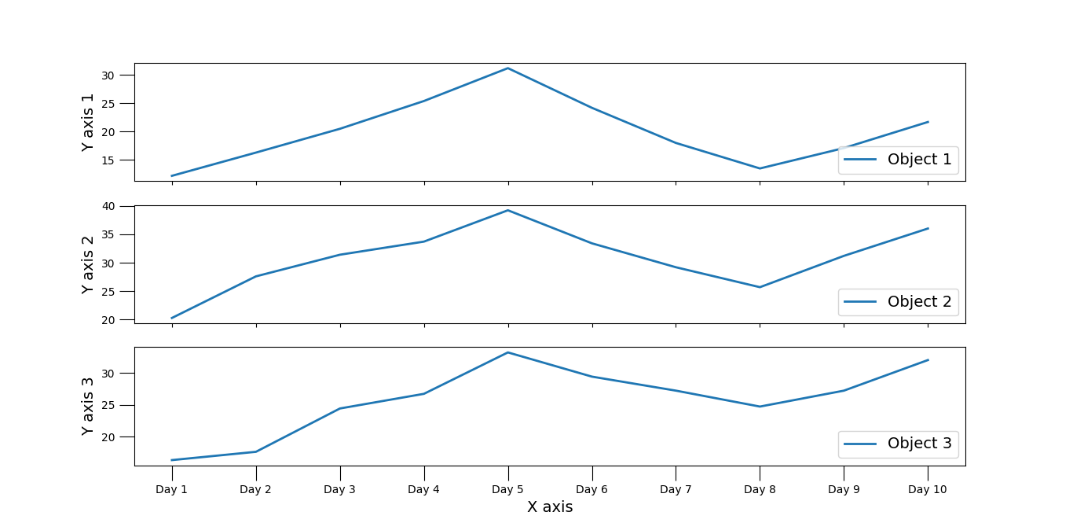
ax[i].plot(x1, y1, label = 'Object 1')

Nếu ax là một ma trận thì truy cập từng phần tử như đối với ma trận thông thường (ví dụ a[i][j]).

Ví dụ:

ax[0].plot(x1, y1, label = 'Object 1', linewidth = 2)  
ax[1].plot(x1, y2, label = 'Object 2', linewidth = 2)  
ax[2].plot(x1, y3, label = 'Object 3', linewidth = 2)  
  
ax[0].set\_ylabel('Y axis 1', fontsize = 14)  
ax[1].set\_ylabel('Y axis 2', fontsize = 14)  
ax[2].set\_xlabel('X axis', fontsize = 14)  
ax[2].set\_ylabel('Y axis 3', fontsize = 14)  
  
ax[0].legend(fontsize = 14, loc = 'lower right')  
ax[0].tick\_params(axis='y', size = 14)  
ax[0].set\_xticklabels([])  
  
ax[1].legend(fontsize = 14, loc = 'lower right')  
ax[1].tick\_params(axis='y', size = 14)  
ax[1].set\_xticklabels([])  
  
ax[2].legend(fontsize = 14, loc = 'lower right')  
ax[2].tick\_params(axis='x', size = 14)  
ax[2].tick\_params(axis='y', size = 14)  
plt.show()

Kết quả:



#### Một số lưu ý khi biểu diễn nhiều biểu đồ trong cùng một khung nhìn

* Không vẽ quá nhiều biểu đồ trong cùng một Figure khiến người xem cảm thấy khó nhìn và phức tạp trong việc xử lý nhiều thông tin cùng lúc.
* Nếu các bộ dữ liệu có chung danh mục trục X thì có thể bố trị dạng n hàng 1 cột và ẩn các X axis của các biểu đồ phía trên để dễ dàng so sánh, tìm quy luật.
* Nếu các bộ dữ liệu dạng time series thì có thể bố trí theo chiều ngang và ẩn các Y axis của các biểu đồ phía phải để thể hiện sự liên tục.

### Tổng kết

Qua bài đọc này, chúng ta đã tìm hiểu được phương pháp trực quan hóa dữ liệu bằng cách kết hợp nhiều biểu đồ cùng loại trên một Axes hay trên một Figure nhằm tăng khả năng so sánh, tìm quy luật, trích xuất thông tin cho người xem.