## **MẢNG ĐỒ THỊ**

### Giới thiệu

Mảng đồ thì cho phép trực quan hóa dữ liệu đa mục tiêu với các tập dữ liệu khác nhau đến từ nhiều kỹ thuật phân tích, đo lường và công thức tổng hợp trên các phạm vi kích thước khác nhau.

Ví dụ: bạn có thể muốn so sánh giá cổ phiếu của một mã cổ phiếu với khối lượng giao dịch của nó hoặc tổng doanh thu của công ty với số lượng sản phẩm đã bán.

### Mảng đồ thị là gì?

Mảng đồ thị là một tập hợp các đồ thị, biểu đồ được hiển thị trong cùng một biểu đồ hay trong một hình.

### Tại sao cần mảng đồ thị?

* Xác định bức tranh toàn cảnh: dễ dàng xem và phân tích nhiều yếu tố của cùng một thuộc tính hoặc nhiều thuộc tính trên cùng một hình ảnh.
* Hỗ trợ tìm kiếm đặc trưng, quy luật của dữ liệu thông qua các biểu đồ cùng loại hay khác loại, dựa trên các bộ dữ liệu khác nhau hoặc được thu thập bởi các kỹ thuật khác nhau hoặc được biểu diễn để trích xuất thông tin khác nhau.
* Tăng khả năng truyền tải thông tin, giải thích thông tin và phát hiện đặc trưng của dữ liệu tới những đối tượng khác.
* Hỗ trợ đưa ra các quyết định dựa trên dữ liệu.
* Đánh giá, so sánh thông tin dễ dàng thông qua các đồ thị thành phần được biểu diễn.

### Một số phương pháp biểu diễn phổ biến

* Nhiều biểu đồ cùng loại trên cùng một Axes
  + 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.
* Nhiều biểu đồ khác loại trên cùng một Axes:
  + 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.
  + Do dạng biểu này rất dễ gây rắc rối, phức tạp cho người xem nên chỉ sử dụng khi các loại biểu đồ có tác dụng bổ trợ thông tin cho nhau cùng lúc.
* Nhiều biểu đồ cùng loại trên một hình.
  + Có thể biểu diễn chung các thuộc tính nhưng khác nhau về giới hạn miền giá trị, hay 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.
* Nhiều biểu đồ khác loại trên một hình.
  + Cần có chung một mục tiêu trực quan hóa để tránh gây rắc rối cho người xem.

### Sử dụng thư viện Matplotlib

Như chúng ta đã biết, một Matplotlib Figure có thể gồm một hoặc nhiều Axes. Trên mỗi Axes có thể gồm một hoặc nhiều đồ thị. Dưới đây sẽ trình bày một số ví dụ vẽ các mảng đồ thị đơn giản.

#### Tạo biểu đồ gồm 2 đồ thị đường cùng tỉ lệ trục trên một Axes

* Import thư viện Matplotlib để sử dụng bằng câu lệnh:

import matplotlib.pyplot as plt

* Tạo dữ liệu mẫu

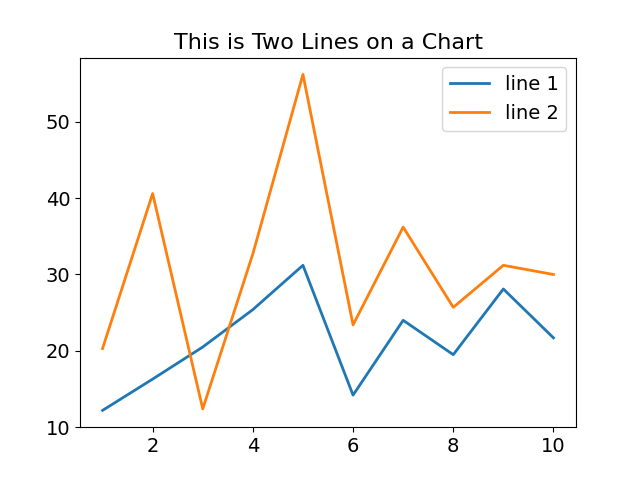
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
y1 = [12.2, 16.3, 20.5, 25.4, 31.2, 14.2, 24, 19.5, 28.1, 21.7]  
y2 = [20.3, 40.6, 12.4, 32.7, 56.2, 23.4, 36.2, 25.7, 31.2, 30]

Chú ý: kích thước của y1, y2 có cùng đơn vị đo và cùng tỉ lệ.

* Vẽ biểu đồ:

axes1 = plt.plot(x, y1)  
axes2 = plt.plot(x, y2)  
plt.title('This is Two Lines on a Chart')  
plt.show()

Kết quả:



#### Tạo biểu đồ đường và biểu đồ cột khác tỉ lệ trục Y trên cùng một Axes

* Import thư viện Matplotlib để sử dụng bằng câu lệnh:

import matplotlib.pyplot as plt

* Tạo dữ liệu mẫu

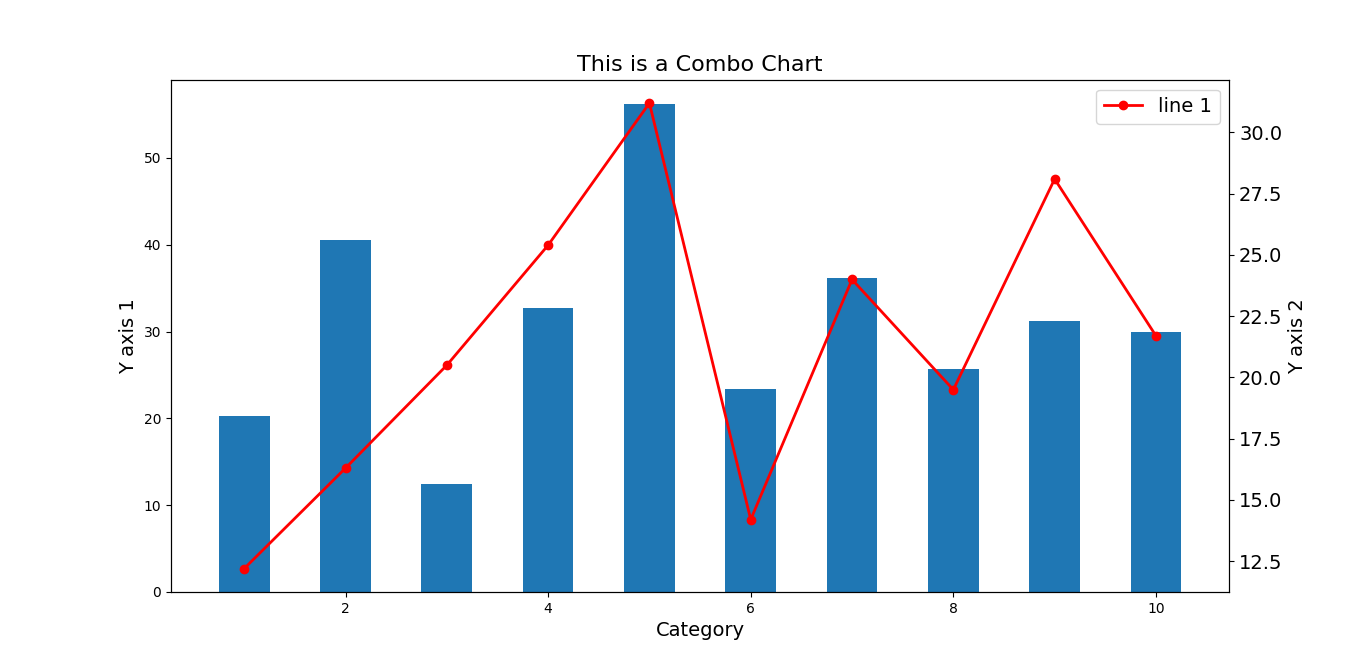
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
y1 = [12.2, 16.3, 20.5, 25.4, 31.2, 14.2, 24, 19.5, 28.1, 21.7]  
y2 = [2000, 4000, 1200, 3200, 5600, 2300, 3600, 2500, 3100, 3000]

Chú ý: y1, y2 không cùng đơn vị đo và tỉ lệ.

* Vẽ biểu đồ:

plt.bar(x, y2, label = 'line 2', width = 0.5) #vẽ biểu đồ cột  
axes1 = plt.gca() #lấy Axes hiện có  
axes2 = axes1.twinx() #khai báo Axes thứ hai có cùng trục X với Axes 1   
axes2.plot(x, y1, label = 'line 1', linewidth = 2, c = 'r', marker = 'o') #vẽ biểu đồ đường  
axes1.set\_xlabel('Category', fontsize = 14) #thay đổi nhãn trục X  
axes1.set\_ylabel('Y axis 1', fontsize = 14) #thay đổi nhãn trục Y trái  
axes2.set\_ylabel('Y axis 2', fontsize = 14) #thay đổi nhãn trục Y phải  
plt.legend(fontsize = 14)  
plt.xticks(fontsize = 14)  
plt.yticks(fontsize = 14)  
plt.title('This is a Combo Chart', fontsize = 16)

Kết quả:



#### Tạo biểu đồ đường và biểu đồ cột trên cùng một Figure

* Import thư viện Matplotlib để sử dụng bằng câu lệnh:

import matplotlib.pyplot as plt

* Tạo dữ liệu mẫu

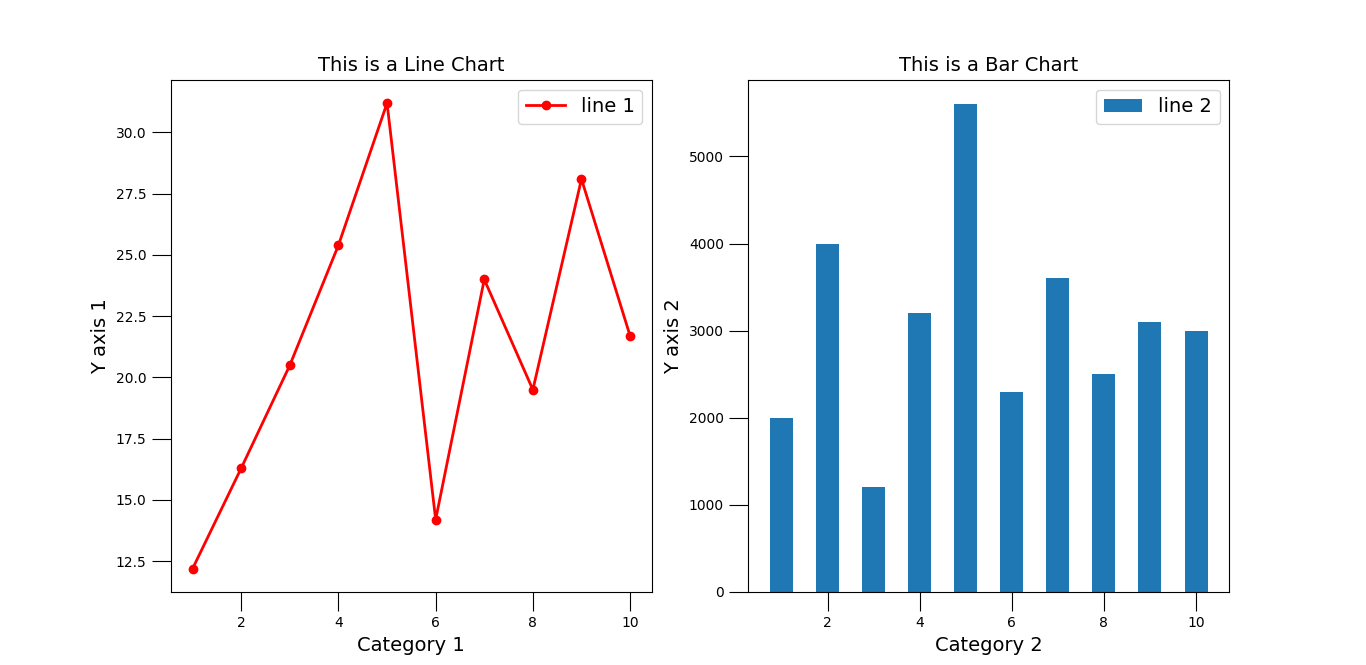
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
y1 = [12.2, 16.3, 20.5, 25.4, 31.2, 14.2, 24, 19.5, 28.1, 21.7]  
y2 = [2000, 4000, 1200, 3200, 5600, 2300, 3600, 2500, 3100, 3000]

Chú ý: y1, y2 không cùng đơn vị đo và tỉ lệ.

* Vẽ biểu đồ:

fig, ax = plt.subplots(1, 2)  
ax[0].plot(x, y1, label = 'line 1', linewidth = 2, c = 'r', marker = 'o')  
ax[1].bar(x, y2, label = 'line 2', width = 0.5)  
  
ax[0].set\_xlabel('Category 1', fontsize = 14)  
ax[0].set\_ylabel('Y axis 1', fontsize = 14)  
ax[0].set\_title('This is a Line Chart', fontsize = 14)  
  
ax[1].set\_xlabel('Category 2', fontsize = 14)  
ax[1].set\_ylabel('Y axis 2', fontsize = 14)  
ax[1].set\_title('This is a Bar Chart', fontsize = 14)  
  
ax[0].legend(fontsize = 14)  
ax[0].tick\_params(axis='x', size = 14)  
ax[0].tick\_params(axis='y', size = 14)  
  
ax[1].legend(fontsize = 14)  
ax[1].tick\_params(axis='x', size = 14)  
ax[1].tick\_params(axis='y', size = 14)  
plt.show()

Kết quả:



### Tổng kết

Qua bài đọc này, chúng ta đã biết các khái niệm về mảng đồ thị và vẽ một số đồ thị kết hợp đơn giản.