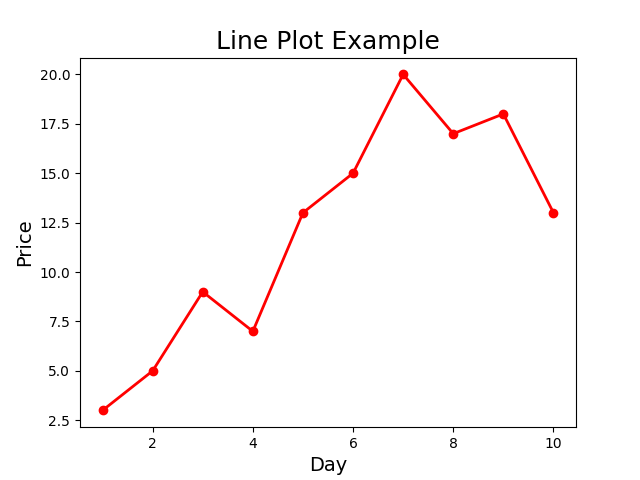
## **Biểu đồ mô tả mối liên hệ giữa các thuộc tính**

### Giới thiệu

Khi chúng ta có nhiều thuộc tính, chúng ta thường quan tâm đến việc mô tả hoặc nắm bắt các mối quan hệ giữa các thuộc tính. Trong bài đọc này, chúng ta sẽ tìm hiểu về cách biểu diễn mối quan hệ giữa các thuộc tính bằng biểu đồ đường và biểu đồ scatter.

### Biểu đồ đường



Biểu đồ đường thường được sử dụng để hiển thị sự thay đổi theo thời gian dưới dạng một chuỗi các điểm dữ liệu được kết nối bởi các đoạn thẳng trên hai trục. Do đó, biểu đồ đường giúp xác định mối quan hệ giữa hai thuộc tính, với một thuộc tính thay đổi theo thời gian và luôn phụ thuộc vào thuộc tính còn lại và có thể giúp đưa ra dự đoán về kết quả của dữ liệu chưa được ghi lại. Trong cùng một biểu đồ cũng có thế biểu thị nhiều đường khác nhau (nếu các trục tuân theo cùng một tỷ lệ) giúp dễ dàng so sánh khi tất cả dữ liệu đều được nằm trên cùng một đồ thị.

#### Thành phần chính trong biểu đồ đường

* Trục X: thể hiện các giá trị của thuộc tính thời gian.
* Trục Y: thể hiện giá trị của dữ liệu tương ứng điểm thời gian.
* Mỗi điểm tương ứng với một giá trị thuộc tính Y tại thời điểm X.
* Các điểm liên tiếp được nối với nhau bởi các đoạn thẳng.
* Không vẽ quá nhiều đường trong cùng một biểu đồ để tránh phiền nhiễu thị giác và sử dụng màu hoặc marker để phân biệt các đường.
* Sử dụng tỉ lệ trục Y phù hợp, luôn bắt đầu tại 0.

#### Ưu điểm và nhược điểm của biểu đồ đường

*Ưu điểm*

* Trình bày đơn giản; dễ đọc và tạo.
* Có thể trình bày trực quan nhiều chuỗi dữ liệu khác nhau với các điểm dữ liệu nhất định.
* Phù hợp để biểu thị một xu hướng trong một khoảng thời gian, có thể sử dụng để dự đoán dữ liệu bị thiếu, dữ liệu chưa xảy ra.
* Xử lý được những dữ liệu tích cực và tiêu cực.

*Nhược điểm*

* Khó đọc hơn khi những dữ liệu chồng lấp lên nhau
* Có thể khó xác định các giá trị chính xác tại một điểm nhất định của biểu đồ.
* Không phù hợp khi khoảng thời gian hay phạm vi dữ liệu quá lớn.

#### Vẽ biểu đồ đường bằng thư viện Matplotlib

* Câu lệnh vẽ biểu đồ đường:

plt.plot(x, y, ’…’)

Trong đó:

x: tọa độ trục X.

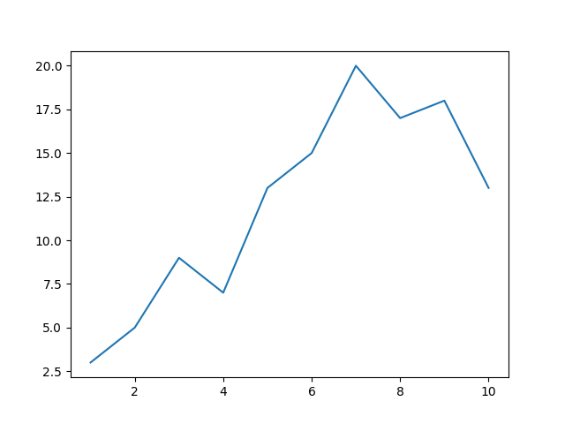
y: tọa độ trụcY.

‘…’: các tham số cài đặt thuộc tính biểu đồ.

Ví dụ

import matplotlib.pyplot as plt  
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
y = [3, 5, 9, 7, 13, 15, 20, 17, 18, 13]  
plt.plot(x, y)  
plt.show()

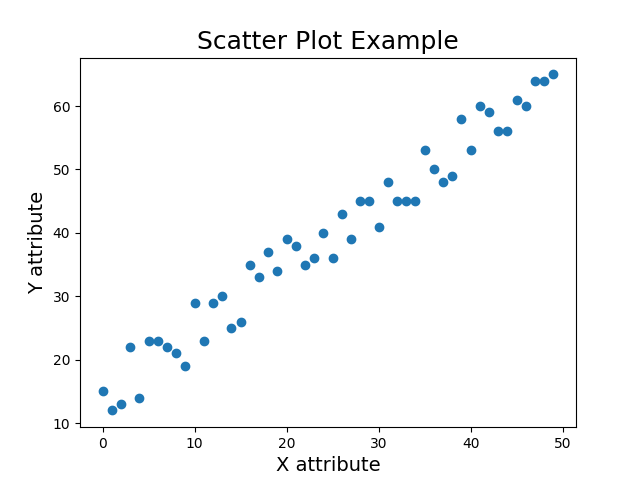
Kết quả:



* Các tham số cài đặt thuộc tính biểu đồ đường:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tham số** | **Giá trị** | **Ý nghĩa sử dụng** | **Ví dụ minh họa** |
| linewidth | float | Cài đặt độ rộng của đường kẻ | plt.plot(x, y, linewidth=3) |
| linestyle | '-' or 'solid' '--' or 'dashed' '-.' or 'dashdot' ':' or 'dotted'  … | Kiểu đường vẽ | plt.plot(x, y, '-') |
| marker | "." point "," pixel "o" circle "1" tri\_down "2" tri\_up "3" tri\_left "4" tri\_right "s" square "p" pentagon "P" plus (filled) "\*" star  '+' plus marker 'x' x marker 'D' diamond marker  … | Đánh dấu đường kẻ | plt.plot(x, y, marker='s', markersize=10, markerfacecolor='red', markeredgecolor = 'blue', markeredgewidth=2) |
| label | str | Nhãn của đường | x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] y1 = [3, 5, 9, 7, 13, 15, 20, 17, 18, 13] y2 = [2, 7,12, 17, 14, 17,18, 23, 22, 28] plt.plot(x, y1, label = 'line 1') plt.plot(x, y2, label = 'line 2') plt.legend(loc='upper right') |
| color | 'b' blue 'g' green 'r' red 'c' cyan 'm' magenta 'y' yellow 'k' black 'w' white  … | Cài đặt màu cho đường kẻ | plt.plot(x, y, 'r') |

### Biểu đồ Scatter

******

Để biểu diễn mối liên hệ giữa hai thuộc tính, người ta dùng biểu đồ Scatter (biểu đồ phân tán). Biểu đồ phân tán sử dụng các dấu chấm (điểm) để thể hiên giá trị (điểm giao nhau) của hai thuộc tính khác nhau. Vị trí của mỗi điểm tương ứng với một quan sát riêng lẻ. Biểu đồ phân tán thường được sử dụng để quan sát mối tương quan giữa hai thuộc tính khác nhau. Và mối tương quan của chúng có thể được diễn tả theo nhiều cách như: mối quan hệ tỉ lệ thuận hay nghịch (độ dốc), mức độ tương quan nhiều hay ít (độ tập trung), và xu hướng của toàn bộ biểu đồ là tuyến tính hay phi tuyến.

#### Thành phần chính trong biểu đồ Scatter

* Trục Y, thường được hiển thị các giá trị của biến phụ thuộc (biến cần dự đoán).
* Trục X thường hiển thị giá trị của các biến độc lập (biến dùng để dự đoán).
* Mỗi điểm thể hiện một quan sát từ một tập dữ liệu với giá trị X và Y tương ứng.
* Độ tuyến tính (linearity) thể hiện mối quan hệ tuyến tính hay phi tuyến.
* Độ dốc (slope) thể hiện chiều hướng thay đổi của Y khi X tăng dần. Nếu Y cũng tăng dần, thì độ dốc là dương (tỉ lệ thuận), nhưng nếu Y giảm dần, thì độ dốc là âm (tỉ lệ nghịch).
* Độ mạnh (strength) thể hiện mức độ tập trung hay phân tán của các điểm. Nếu các điểm cách xa nhau, thì mối quan hệ giữa các biến là yếu. Nếu các điểm càng gần sát vào nhau xung quanh một đường, thì mối quan hệ này là chặt chẽ.

#### Ưu điểm và nhược điểm của biểu đồ Scatter

*Ưu điểm:*

* Phát hiện mối quan hệ giữa hai thuộc tính.
* Phát hiện các điểm ngoại lệ.
* Cung cấp một vài mô tả ban đầu về mối quan hệ giữa hai thuộc tính như: tỉ lệ thuận/nghịch, mức độ tương quan,…

*Nhược điểm:*

* Không dễ để phân tích như các loại biểu đồ khác.
* Chỉ thích hợp cho các loại dữ liệu cần đi tìm mối tương quan.

#### Vẽ biểu đồ Scatter bằng thư viện Matplotlib

* Câu lệnh vẽ biểu đồ Scatter:

plt.scatter(x, y, '…')

trong đó:

x: danh sách giá trị thuộc tính độc lập.

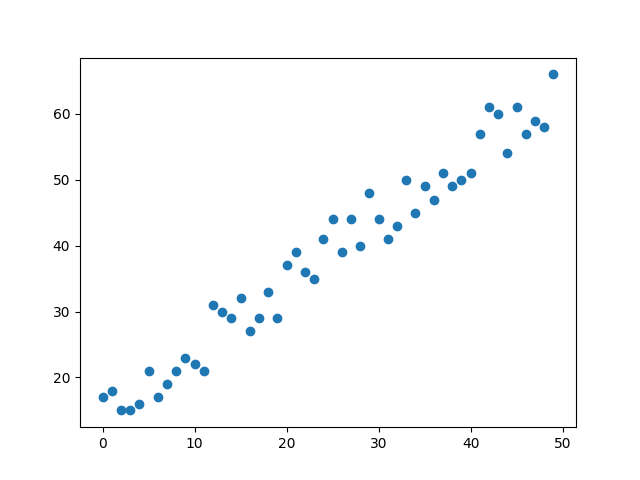
y: danh sách giá trị thuộc tính phụ thuộc.

‘…’: các tham số cài đặt thuộc tính biểu đồ

Ví dụ

import matplotlib.pyplot as plt  
import numpy as np  
x = range(50)  
y = range(50) + np.random.randint(10,20,50)  
plt.scatter(x, y)  
plt.show()

Kết quả:



* Các tham số cài đặt thuộc tính biểu đồ Scatter:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tham số** | **Giá trị** | **Ý nghĩa sử dụng** | **Ví dụ minh họa** |
| s | float or array-like | Thay đổi kích thước điểm | sizes = np.random.randint(5, 200, 50) plt.scatter(x, y, s=sizes) |
| c | color or list of colors | Thay đổi màu sắc điểm | colors = np.random.randint(0, 100, 50) plt.scatter(x, y, s=sizes, c = colors) |
| marker | "o" circle "1" tri\_down "2" tri\_up "3" tri\_left "4" tri\_right "s" square "p" pentagon "P" plus (filled) "\*" star  '+' plus marker 'x' x marker 'D' diamond marker  … | Thay đổi kiểu đánh dấu điểm | plt.scatter(x, y, s=sizes, c = colors, marker='\*') |
| alpha | float or None | Hòa trộn giữa độ trong suốt (0) và rõ ràng (1) | plt.scatter(x, y, s=sizes, c = colors, alpha=0.5) |

### Tổng kết

Qua bài đọc này, chúng ta đã tìm hiểu được các định nghĩa, ý nghĩa sử dụng, các thành phần chính, cách sử dụng thư viện Matplotlib để vẽ và tinh chỉnh một số biểu đồ:

* Biểu đồ đường
* Biểu đồ Scatter