Tổng quan về Data Visualization

Trần Nhật Khoa , Trường Đại học Công nghệ thông tin – ĐHQG TP.HCM

1. Lời mở đầu

Bạn đã bao giờ biết cách trình bày một bộ dữ liệu (Dataset) hay cách phân tích sơ bộ về nó thông qua đồ thị chưa? Nếu chưa thì trong bài viết này, mình sẽ giới thiệu cho các bạn cách để biểu diễn cơ bản dữ liệu bằng đồ thị thông qua thư viện *Matplotlib* của ngôn ngữ lập trình *Python*. Let's start !

2. Giới thiệu chung về Data và một số ví dụ đi

Hiện nay với sự phát triển của cách mạng công nghiệp 4.0 thì việc sử dụng *dữ liệu (Data)* là việc hết sức cần thiết và quan trọng. Và khái niệm *Bigdata* thật ra đã có từ rất lâu về trước trong sinh hoạt, công việc hằng ngày của con người. Ví dụ như việc ngân hàng lưu những thông tin về khách hàng , về số tiền gửi, số tiền vay và lãi họ nhận được theo thời gian. Nhưng cứ ngày qua ngày tất cả chỉ dừng lại ở việc ghi chép và lưu trữ. Cho đến thời gian gần đây khi con người nhìn lại vào lượng dữ liệu khổng lồ đó thì họ mới bất giác nhận ra thứ mà họ coi như không có tác dụng ngoài việc lưu trữ lại là một nguồn tài nguyên hữu ích có thể nói lên rất nhiều điều. Và sử dụng *biểu đồ (Chart)* hay *đồ thị (Graph)* là một cách để trực quan những ý nghĩa mà bộ dữ liệu đó mang lại. Ví dụ như bạn thu thập dữ liệu về số ca nhiễm Virus Covid 19 theo ngày thì việc biểu diễn trực quan nó lên đồ thị sẽ dễ dàng khiến cho bạn biết được số ca nhiễm đang tăng nhanh hay giảm đáng kể so với các ngày vừa qua và thậm chí còn có thể dự đoán nó sẽ tăng như thế nào trong những ngày sau. Hay một ví dụ kinh điển trong Machine Learning là bài toán *dự đoán (Prediction)*

,người ta đã biểu diễn bộ dữ liệu lên đồ thị và dùng thuật toán *Hồi quy tuyến tính (Linear Regression)* thông qua các công cụ *Giải tích (Calculus)*, *Đại số tuyến tính (Linear Algebra)*, … để đưa ra một một hàm tuyến tính phù hợp nhất với bộ Data, từ đó mà có thể dự đoán tương đối được một giá trị y theo x. (bạn có thể xem bài viết về Linear Regression của clb AI thông qua liên kết sau đây (*https://tutorials-aiclub-cs-uit-edu-vn.translate.goog/index.php/2021/04/24/linear-regression/?_x_tr_sl=vi&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=sc&_x_tr_sch=http*)

3. **Các khái niệm cơ bản**

   Để cài đặt thư viện Matplotlib thì bạn có thể xử dụng lệnh *pip install matplotlib* trên Command Prompt hay Terminal của trình soạn code mà bạn đang sử dụng.

   Về cơ bản một Matplotlib figure có thể được phân loại thành 4 phần : Figure, Axes, Axis, Artist. Nhưng ở bài viết này ta chỉ trình bày sơ bộ về 1 thành phần đó là Axes

   - Axes: Nơi mà các đối tượng thật sự được vẽ lên

4. **Cách tạo một đồ thị đơn giản bằng Matplotlib**

   Ở đây mình sẽ sử dụng Google Colab để code vì tính tiện lợi và bộ nhớ khủng mà nó cung cấp cho mình (Bạn có thể sử dụng các trình biên dịch khác như *Anaconda* hay *Pycharm* đều được)

   - Trước tiên chúng ta cần phải có một bộ dữ liệu để mà thực hành, và mình có tạo một file dữ liệu đơn giản về nhiệt độ trung bình theo từng tháng trong năm 2022 của hai tỉnh Cà Mau và Bạc Liêu. Đây là link raw *https://raw.githubusercontent.com/trannhatkhoacm1612/Tran-Woffy/main/dataset%20do%20c.csv*

Sau đó dùng lệnh *!wget* để upload file lên trên GGC (khi load xong mình sẽ lưu tên file là dataset) :

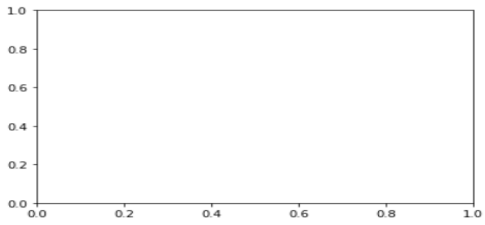
```
!wget
https://raw.githubusercontent.com/trannhatkhoacm1612/Tran-Woffy/main/dataset%20do%20c.csv
```

- Đầu tiên ta sẽ import module của *Matplotlib*, ngoài ra sẽ còn có thêm *Numpy* và *Pandas* cho việc xử lý dữ liệu:

```
# import thư viện và đặt tên nó là plt
import Matplotlib.pyplot as plt
import Numpy as np
import Pandas as pd
```

- Tiếp theo ta sẽ khởi tạo hệ trục bằng phương thức *plt.subplots()* và sẽ biểu diễn thử bằng *plt.show()*:

```
# Khởi tạo hình ảnh (figure) và hệ trục tọa độ ( axes ), mình chỉ khởi tạo thử nên sẽ không truyền dữ liệu vào
fig, ax = plt.subplots()
# Biểu diễn đồ thị
plt.show()
```



- Bước tiếp theo ta đọc dữ liệu của file bằng pandas và sẽ thử biểu diễn dữ liệu lên trên đồ thị bằng phương thức *plt.plot(*index cột dữ liệu x, *index cột dữ liệu y):*

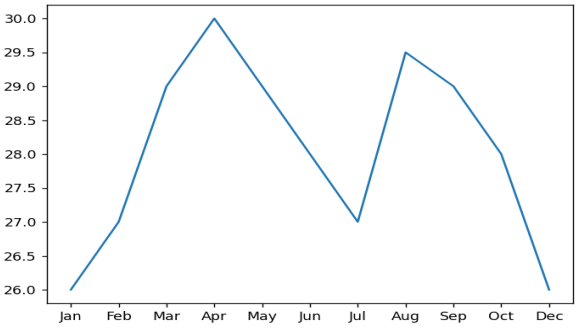
```
# Đọc file csv và đóng gói nó vào một biến bằng pd.read_csv(*string[tên file])
df = pd.read_csv('dataset.csv')
```

- Sử dụng phương thức *np.asarray(*cột cần lấy)* của Numpy để đóng gói một cột của dataset:

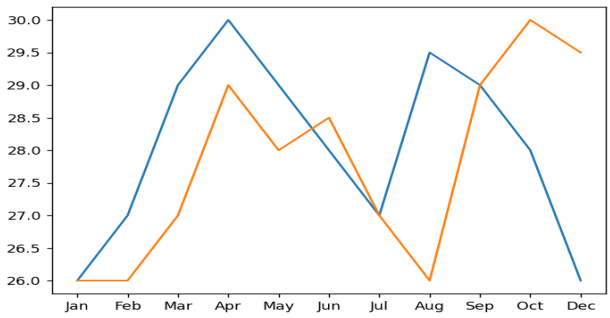
```
x = np.asarray(df['Month'])
y = np.asarray(df['Ca Mau'])
```

- Biểu diễn lên trên đồ thị theo dạng đường thẳng:

```
ax.plot(x,y)
plt.show()
```

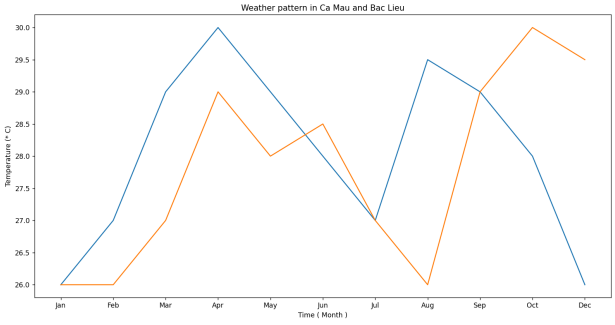



```
# Có thể vẽ chồng thêm một đồ thị nhiệt độ trung bình của Bạc Liêu bằng cách tương tự

# Tạo một biến lưu dữ liệu cột
z = np.asarray(df['Bac Lieu'])

# Biểu diễn hai đồ thị liên tiếp thì cứ gọi hai phương thức của nó ra liên tục
ax.plot(x,y)
ax.plot(x,z)
plt.show()
```



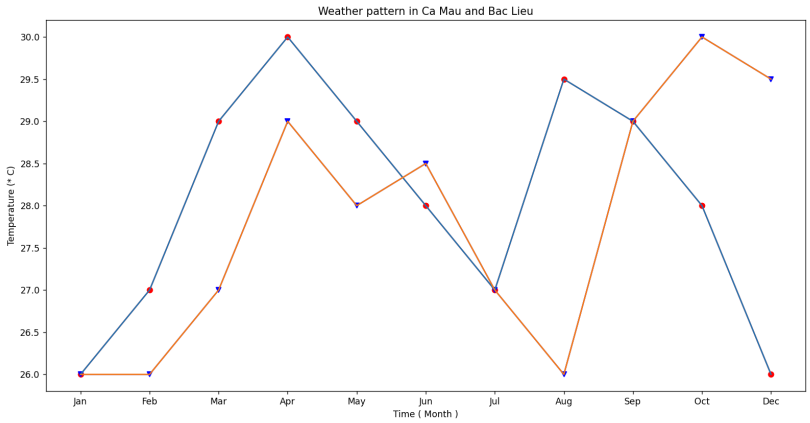
- Ngoài ra việc đặt nhãn (Label) và tùy chỉnh (Customize) cho các đối tượng trên đồ thị là một việc hết sức quan trọng

```
# Ta sẽ đặt nhãn cho cả đồ thị lớn bằng phương thức plt.set_title(*string), cho trục hoành (x-axis) , trục tung(y-axis) bằng 2 phương thức là plt.set_xlable(*string) và plt.set_ylable(*string)

ax.set_title('Nhiệt độ trung bình từng tháng của tỉnh Cà Mau và Bạc Liêu')

ax.set_xlable('Tháng')

ax.set_ylable('Nhiệt độ (độ c)')
```



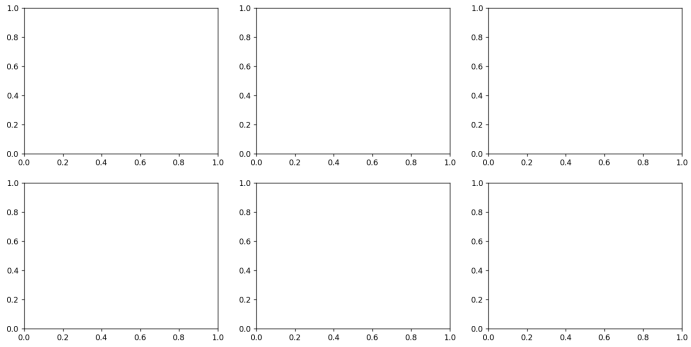

```
# Để nhận diện hai đồ thị của hai tỉnh hay đơn giản là muốn đồ thị đẹp hơn ta tùy chỉnh chấm các điểm theo từng tọa độ (x,y), đổi
```

```
màu đường hay thậm chí là thay đổi kiểu của
đồ thị mà ta muốn biểu diễn.

# dùng các phần tử (parameters) như color,
market, linestyle để tùy chỉnh đồ thị

ax.plot(x,y,color = 'r', marker = 'o',
linestyle = '-' )

ax.plot(x,z,color = 'b', marker = 'v',
linestyle = None )
```

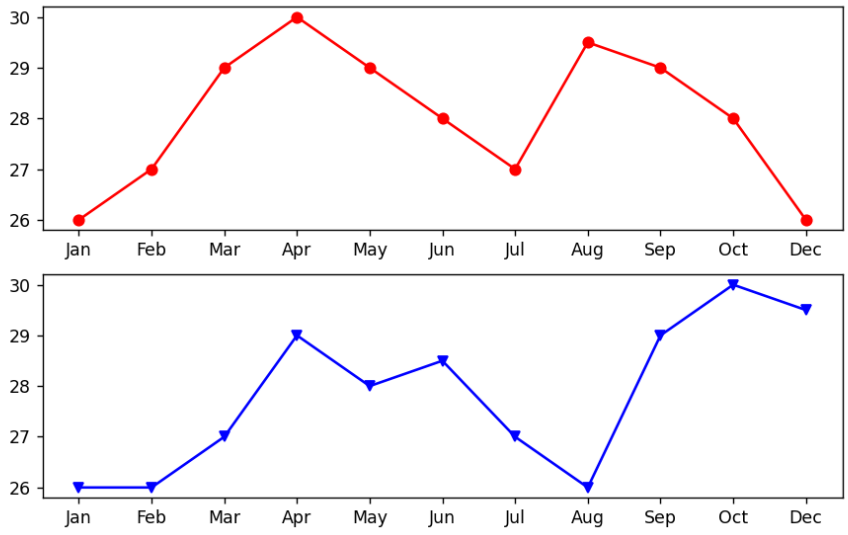


- Nhiều lúc một đồ thị nhỏ có thể chứa nhiều thông tin, nên đôi khi việc đưa nhiều đối tượng lên chung một biểu đồ sẽ làm cho nó thành một đống 'bùi nhùi' khiến chúng ta khó có thể nào phân tích, do đó ta có thể làm như sau để chia ra thành nhiều đồ thị.

```
# Ta sẽ truyền tham số vào phương thức
plt.subplots():

fig, ax = plt.subplots(2,3)

# Khi đó nó sẽ như một ma trận 2 x 3 hay
mảng hai chiều với mỗi phần tử là các đồ thị
```

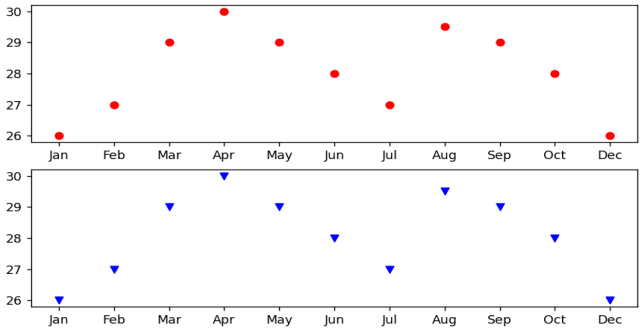


```
# Như ở đây ta sẽ truy xuất và tùy chỉnh từng đồ thị về nhiệt độ trung bình của Cà Mau và Bạc liêu theo từng tháng vào năm 2022 như sau

fig, ax = plt.subplots(2,1)

# Truy suất và tùy chỉnh các đồ thị. Đối với mảng hai chiều thì truy xuất bằng list. Ví dụ với mảng 2 x 3 thì phần tử hàng 2 cột 1 sẽ là ax[1,0]

ax[0].plot(x,y,color = 'r', marker = 'o', linestyle = '-' )

ax[1].plot(x,z,color = 'b', marker = 'v', linestyle = None)

plt.show()
```

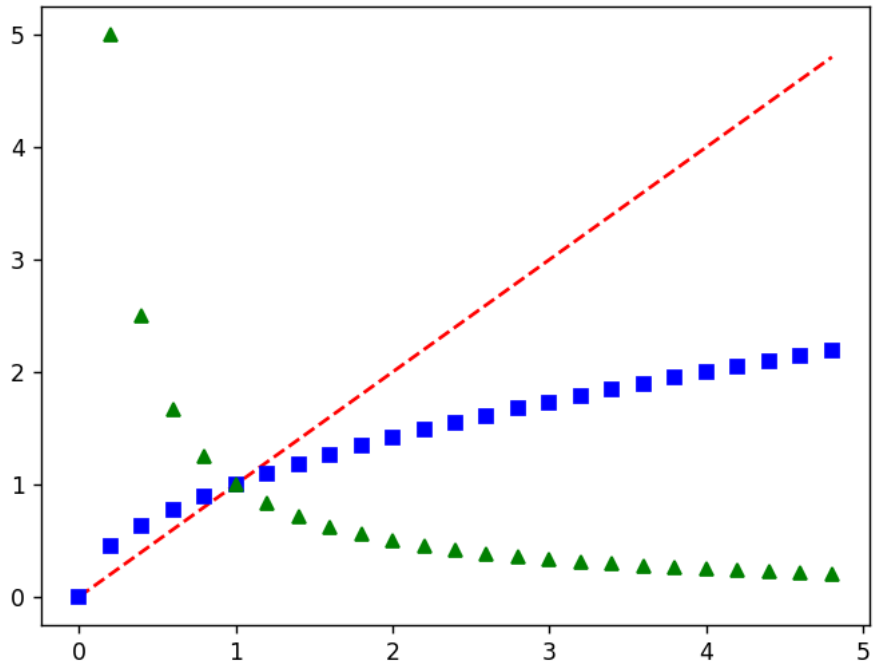


- Ngoài ra nếu bạn chỉ muốn biểu diễn các điểm lên trên đồ thị thì ta có thể làm như sau

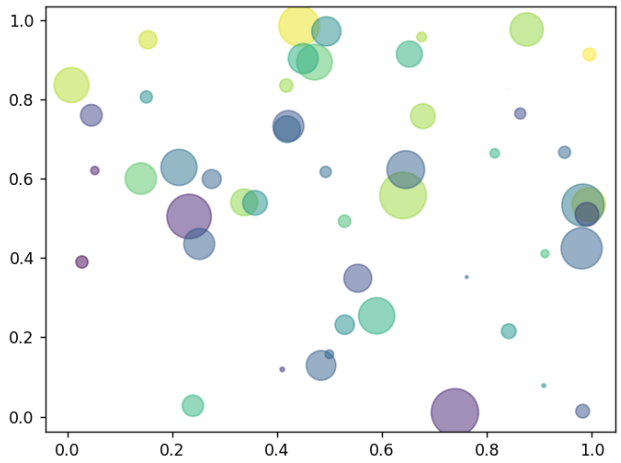
```
fig, ax = plt.subplots(2,1)

# Truy suất và tùy chỉnh các đồ thị. Đối với
mảng hai chiều thì truy xuất bằng list. Ví
dụ với mảng 2 x 3 thì phần tử hàng 2 cột 1
sẽ là ax[1,0]

ax[0].scatter(x,y,color = 'r', marker = 'o')
ax[1].scatter(x,z,color = 'b', marker = 'v')
plt.show()
```



- Đây là một số đoạn code tham khảo về các phương thức trên
  - o
  - o
```
t = np.arange(0., 5., 0.2)
# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**(1/2), 'bs', t, t**-1, 'g^') # ghép 3 đồ thị
plt.show()
```



```
o N = 50
  x = np.random.rand(N)
  y = np.random.rand(N)
  colors = np.random.rand(N)

  area = (30 * np.random.rand(N))**2  # Ngẫu
  nhiên từ 0 tới 30

  plt.scatter(x, y, s=area, c=colors,
  alpha=0.5)

  plt.show()
```



Vậy về cơ bản chúng ta đã biểu diễn được một bộ dữ liệu cơ bản lên trên một đồ thị, rất đơn giản đúng không các bạn. Và qua đó thấy được rằng ta dễ dàng khai thác được rất nhiều thứ thay vì nhìn và phân tích trên bộ dữ liệu khô khan đó, ví dụ như bạn có thể biết tháng nhiệt độ trung bình cao nhất , tháng có nhiệt độ trung bình thấp nhất, tháng nào cả hai tỉnh có cùng nhiệt độ,…

Mình mong nó sẽ hữu ích với mọi người, xin cảm ơn đã xem.