

Th.S LÊ NGỌC THẠCH

**ỨNG DỤNG
PHÂN TÍCH DỮ LIỆU
VÀ
TRÍ TUỆ NHÂN TẠO
VỚI PYTHON**

2021

Đây là eBook của riêng bạn – đề nghị không chia sẻ cho ai khác nhé!

Lời nhắn

eBook "ỨNG DỤNG PHÂN TÍCH DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO VỚI PYTHON" này dự kiến phát hành vào tháng 12/2021. Bạn có thể đặt hàng ngay bây giờ với ưu đãi giảm 50% bằng 2 cách sau:

Cài **App MinePI** cho điện thoại tại theo link:

<https://minepi.com/thachln>

Sử dụng invitation code: **thachln**

Thử dùng điện thoại để đào Pi Coin. eBook được chấp nhận thanh toán Pi Coin với giá tương đương 399K đồng (Xem hình thức thanh toán tiền mặt).

Phiên bản bạn đang nhận là bản nháp trong quá trình hoàn thiện.

Bạn được gọi riêng để tham khảo hoặc để góp ý. Vì thế bạn được toàn quyền sử dụng và **KHÔNG** chia sẻ với bất kỳ ai khác nhé, **KHÔNG** lưu trữ trên internet nói chung để hạn chế đến tay người không thật sự cần nó!

Về nội dung bạn thu lượm được từ eBook dưới dạng các bài tóm tắt, đánh giá, hoặc đề nghị bổ sung thì rất được **KHUYẾN KHÍCH** chia sẻ công khai.



Đặc biệt khuyến khích bạn chia sẻ link:

<https://ThachLN.github.io>

Lê Ngọc Thạch

Hãy cài app [MinePI](https://minepi.com) ngay với Invitation Code là **thachln** để nhận ngay bản nháp (hơn 600 trang) nhé!

Hình thức thanh toán tiền mặt – Đặt hàng ngay bây với 199K, tiết kiệm 200K qua:

① MoMo	② Chuyển khoản
<div><p>Thanh toán qua MoMo</p><p>0908550642 Lê Ngọc Thạch</p><p>Nội dung tin nhắn: email sdt Ví dụ: abc@gmail.com 0908550642 AIPYTHON Email và sdt của người nhận eBook.</p><p>Trường hợp tặng bạn bè thì ghi thông tin email và sdt của bạn.</p><p>Quét mã QR thanh toán 199K.</p><p>199.000đ</p></div>	<div><p>Thanh toán qua NH Tiên Phong</p><p>Lê Ngọc Thạch, Ngân Hàng Tiên Phong, CN HCM Số tài khoản: 00002888001 Nội dung tin nhắn: email sdt AIPYTHON Vd tin nhắn: abc@gmail.com 0908456321 AIPYTHON</p><p>Quét mã QR để thanh toán cho:</p><p>Quét mã vạch này để giao dịch</p></div>

Mục lục

Quy ước	7
Ngày 1 – Chủ đề: Giới thiệu về ngôn ngữ thống kê, ngôn ngữ lập trình	10
Bài 1: Tóm tắt về thống kê (Statistics)	12
Bài 2: Ngôn ngữ lập trình Python	19
Bài 3: Ngôn ngữ Python và phần mềm Anaconda	27
Bài 4: Cài đặt thêm phần mềm	48
Bài 5: Nhập liệu, biên tập, lưu trữ dữ liệu với Python	53
Bài tập ngày 1	70
Thử thách cho bạn!	72
Ngày 2 – Chủ đề: Biểu đồ	73
Bài 6: Các loại biểu đồ	75
Bài 7: Vẽ biểu đồ trong Python	81
Bài 8: Nguyên tắc soạn biểu đồ	96
Bài 9: Giới thiệu Matplotlib	98
Bài 10: Giới thiệu Bokeh	112
Bài 11: Khai phá Bokeh	120
Ngày 3 – Phân tích mô tả	146
Bài 12: Phân tích mô tả dữ liệu Bank Marketing	148
Bài 13: Phân tích dữ liệu Marketing #2	159
Bài 14: So sánh 2 tỉ lệ	166
Bài 15: Mô hình kiểm định giả thuyết	177
Bài 16: Ứng dụng minh họa kiểm định giả thuyết	178
Bài 17: Phân tích mối tương quan	188
Ngày 4 – Chủ đề: Dữ liệu lớn	196
Bài 18: Cách xử lý tập hợp dữ liệu lớn	197
Bài 19: Sử dụng Ubuntu	232
Bài 20: Cài đặt Hadoop 3.2	241
Bài 21: Trải nghiệm Hadoop với Python	249
Ngày 5 – Chủ đề: Dự báo bằng mô hình hồi qui tuyến tính	255

Đây là eBook của riêng bạn – đề nghị không chia sẻ cho ai khác nhé!

Bài 22: Giới thiệu mô hình hồi qui tuyến tính	256
Bài 23: Diễn giải mô hình hồi qui tuyến tính.....	260
Bài 24: Mô hình hồi qui tuyến tính đa biến.....	273
Bài 25: Dự báo bằng mô hình hồi qui tuyến tính	277
Ngày 6 – Chủ đề: Dự báo bằng mô hình hồi qui logistic	281
Bài 26: Giới thiệu mô hình hồi qui logistic.....	282
Bài 27: Mô hình hồi qui logistic đa biến (Multiple logistic regression model).....	286
Bài 28: So sánh mô hình.....	290
Bài 29: Dự báo bằng mô hình hồi qui logistic	296
Ngày 7 – Chủ đề: Phân tích đa biến	303
Bài 30: Xử lý giá trị trống	304
Bài 31: Mô hình phân tích phân định (Linear discriminant analysis) ..	308
Bài 32: Mô hình thành phần (Principal Component Analysis)	316
Bài 33: Mô hình phân tích cụm/nhóm (cluster analysis)	324
Ngày 8 – Chủ đề: Machine Learning	332
Bài 34: Giới thiệu Machine learning	333
Bài 35: Mô hình SVM.....	335
Bài 36: Mô hình Random Forest	343
Bài 37: Mô hình Artificial Neural Network	347
Bài 38: Machine Learning với Python Tensorflow.....	353
Ngày 9 – Chủ đề: Recommendation.....	382
Bài 39: Giới thiệu phương pháp gợi ý Collaborative filtering	383
Bài 40: Triển phương pháp gợi ý Collaborative filtering bằng R	393
Ngày 10 – Chủ đề: Natural Language Processing.....	399
Bài 41: Các kỹ thuật cơ bản	400
Bài 42: Trích đặc trưng (Feature extraction).....	405
Bài 43: Giới thiệu ứng dụng phân tích cảm xúc (Sentiment Analysis) ..	415
Bài 44: Giới thiệu ứng dụng phân tích từ vựng (Word Embedding) ...	426
Bài 45: Giới thiệu ứng dụng xác định chủ đề (Topic Modeling)	438
Ngày 11 – Chủ đề: Computer Vision	449
Bài 46: Giới thiệu Face recognition	450

Đây là eBook của riêng bạn – đề nghị không chia sẻ cho ai khác nhé!

Bài 47: Giới thiệu mô hình CNN	465
Ngày 12 – Chủ đề: Nhận diện tiếng nói (Speech Recognition)	484
Bài 48: Giới thiệu đặc trưng của âm thanh	485
Bài 49: Các thao tác cơ bản với file âm thanh	491
Bài 50: Mô hình Chuyển giọng nói thành văn bản	495
Ngày 13 – Chủ đề: Phân tích dữ liệu theo trường phái Bayes	498
Bài 51: Nhập môn.....	499
Tạm kết thúc	499
Phụ lục	500
Quan sát giao dịch cổ phiếu VNM (Vinamilk)	501
Đọc và vẽ tín hiệu âm thanh.....	511
Tải sách nói “Từ tốt đến vĩ đại”	514
Đọc ảnh y khoa DiCOM.....	517
Áp dụng biến đổi Fourier cho ảnh.....	520
Sử dụng Git.....	524
Khảo sát ảnh và ma trận	553
Phát triển ứng dụng với Python.....	555
Xử lý file pdf	562
Khảo sát file âm thanh.....	570
Phân tích âm thanh với thư viện mutagen	573
Khám phá Python trong WSL2	574
Crawl dữ liệu bằng Selenium	576
Sử dụng OpenCV để phân tích dữ liệu ảnh.....	577
Cài đặt OpenCV	578
Đóng gói chương trình Python	579
Tải file video từ Youtube	582
Sinh code Restful API từ database	583
Trải nghiệm Restful API với Flask	585
Trải nghiệm Kafka.....	586
Trải nghiệm Apache NiFi.....	589
Giới thiệu superset.....	594
Cài đặt.....	595

Đây là eBook của riêng bạn – đề nghị không chia sẻ cho ai khác nhé!

Khởi động lại superset.....	600
Cài đặt và cấu hình Nginx	601
Truy cập nginx trên Ubuntu từ máy Windows.....	602
Khai thác superset	604
Cài đặt Ubuntu Server 20.04.2	608
Superset introduction.....	609
Giới thiệu PowerBI.....	609
Import data	612
Tạo biểu đồ	615
Bài bổ sung	617
Bài 101: Trải nghiệm ứng dụng với Flask	618
Bài 102: Xử lý dữ liệu	624

Quy ước

Một số nội dung trong tài liệu được trình bày với các định dạng khác nhau thì có ý nghĩa của nó, bạn đọc nên nắm thông tin này để tiện theo dõi.

Mã nguồn

Mã lệnh được viết và đóng khung với font chữ Courier New như sau:

```
print('Xin chào!')
print('Welcome!')
print('{} + {} = {}'.format(1, 2, (1 + 2)))
print('%d + %d = %d' % (1, 1, 4))
```

Bạn có thể sao chép và dán (đôi khi trong tài liệu viết luôn tiếng Anh: copy & paste) vào phần mềm để chạy.

Kết quả của lệnh, tùy theo phần mềm bạn sử dụng để chạy mã nguồn thì kết quả sẽ hiển thị ở các vị trí khác nhau. Phần văn bản kết xuất của phần mềm sẽ được trình bày theo khung màu đỏ gạch bên dưới:

```
Xin chào độc giả của ebook Chạm tới AI trong 10 ngày.
welcome to ebook Touch on AI in ten days.
1 + 1 = 4
```

Lệnh thực thi trong hệ điều hành

Trường hợp các lệnh thực thi trong môi trường hệ điều hành (phân biệt với các lệnh, hoặc mã nguồn của chương trình thực thi trong môi trường của R hoặc Python như RStudio hoặc Spyder như đã qui ước ở mục Mã nguồn) thì dấu hiệu như sau:

Đối với lệnh thực thi trong dấu nhắc lệnh của Anaconda hoặc trong cửa sổ lệnh CMD của Windows, hoặc trong Terminal của Linux/MacOS thì khung màu vàng có 2 vạch đậm ở cạnh trái và phải như sau:

```
pip install python-docx
```

Cặp dấu nháy

Các dữ liệu dạng chuỗi (string, text, char nói chung là có nghĩa giống nhau trong Python) được bao đóng trong **dấu nháy đơn** hoặc **dấu nháy đôi**. Trên bàn phím máy tính thì dấu **nháy trái** và **phải** là giống nhau. Tuy nhiên trong phần mềm soạn thảo văn bản như Microsoft Word thì gập dấu nháy đơn và đôi được thay thế bằng ‘, ’’ để tăng tính thẩm mỹ. Các dấu nháy thẩm mỹ này khác với kí tự ' và " trên bàn phím (phím bên trái phím Enter).

Đôi khi bạn copy & paste mã nguồn vào các phần mềm như Microsoft Word thì các dấu nháy có thể bị “trang trí” lại như trên. Vì vậy khi copy mã nguồn từ Microsoft vào các phần mềm chạy R hoặc Python thì hãy thay thế lại cho đúng.

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

Một qui ước khác liên quan đến dấu nhảy đôi là khi dùng trong văn bản để bao đóng danh từ riêng, hoặc lệnh như: *Bạn hãy thử gõ lệnh “exit()” trong cửa sổ console để thoát chương trình Python.* Trong câu hướng dẫn này thì lệnh `exit()` được gõ vào console **KHÔNG** bao gồm cặp dấu nhảy.

Cách viết thông tin lặp lại với dấu ba chấm

Khi cần mô tả một lệnh có nhiều thông tin lặp lại thì dùng dấu ba chấm như ví dụ sau.

Khi cần mô tả hàm xóa cột dữ liệu trong tham số thứ nhất của hàm `drop` như:

```
df.drop(['cột 1', 'cột 2', ...], axis =1)
```

thì phần in đậm có nghĩa là có thể gồm 1 hoặc nhiều tên cột dữ liệu. Ví dụ lệnh sau có nghĩa là xóa cột `Fullname` khỏi DataFrame `df`.

```
df.drop(['Fullname'], axis =1)
```

Hoặc lệnh sau sẽ xóa 2 cột `Fullname` và `Year` khỏi DataFrame `df`:

```
df.drop(['Fullname', 'Year'], axis =1)
```

Kí hiệu optional (không bắt buộc)

Khi sử dụng hàm số thì có nhiều tham số (argument, parameter) không bắt buộc (optional) thì sử dụng cặp dấu ngoặc vuông `[]`. Ví dụ hàm `plot` bên dưới không bắt buộc tham số `x` và `format`:

```
plot([x], y, [format])
```

Cách viết in nghiêng cho các biến

Thông thường các biến được mô tả trong các câu lệnh sẽ để trong cặp dấu ngoặc nhọn `<>`. Ví dụ lệnh sau có nghĩa là khi gõ lệnh bạn phải thay nội dung *<tên cột>* thành tên cột cụ thể trong data frame của bạn:

```
df[df['<tên cột>'].notnull()]
```

Trong tài liệu này đôi lúc sẽ không dùng cặp dấu ngoặc nhọn để mô tả lệnh chung như sau:

```
df[df['tên cột'].notnull()]
```

Cách viết trình tự bấm chọn menu

Khi cần trình bày thứ tự các nút bấm, hoặc các mục cần bấm trong các thao tác thì sẽ dùng dấu lớn hơn `>`. Ví dụ khi hướng dẫn bạn vào trang web

“<https://github.com/vncorenlp/VnCoreNLP>”, bấm vào nút “Clone”, sau đó bấm tiếp vào nút hoặc link “Download Zip” thì sẽ viết gọn như sau:

Bấm vào nút Clone > nút Download Zip, hoặc nút Clone > Download Zip.

Các từ tiếng Anh viết tắt thường xuyên được sử dụng trong sách

AI: Artificial Intelligent - **Trí thông minh nhân tạo**. Nhiều người dịch là Trí Tuệ Nhân Tạo. Trong sách này tôi muốn dùng đúng nghĩa Intelligent có nghĩa là Trí thông minh thôi vì khoảng cách từ Thông Minh đến Tuệ thì rất rất là xa. Trí thông minh nhân tạo tôi cho là phù hợp nhất trong bối cảnh hiện nay. Có thể bạn và cả tôi quen với cách đọc Trí Tuệ Nhân Tạo vừa gọn và vừa sang. Tuy nhiên nếu khi cần nói thì vẫn nên dùng từ “Thông minh” để phản ánh đúng mức độ của nó để mà còn phấn đấu đến mức “Tuệ”. Đẳng nào thì tôi cũng viết là AI thay vì viết tiếng Việt nên chắc không nhầm lẫn.

Đường dẫn thư mục (Path)

Trong Windows thì dấu cách thư mục là dấu xuyệt trái (back slash). Ví dụ: D\ai2021\data.

Tuy nhiên ngôn ngữ R hoặc Python được thiết kế tương thích với các hệ điều hành khác như Macintosh, Linux. Các hệ điều hành thì dùng dấu xuyệt phải (right slash) để phân cách thư mục. Ví dụ: /mnt/d/ai2021.

Vì vậy khi trình bày đường dẫn thư mục trong câu văn thì đôi lúc dùng \, hoặc đôi lúc dùng / do dữ liệu được minh họa trên Windows hoặc Linux.

Nhưng trong mã nguồn (R hoặc Python) thì đều thống nhất là dùng dấu xuyệt phải / như sau:

```
read.csv("D:/ai2021/data/test.csv")
```

Trong Windows, code R hoặc Python có một cách khác là dùng hai (double) dấu \. Ví dụ:

```
read.csv("D:\\ai2021\\data\\test.csv")
```

Tuy nhiên code này không tương thích trong Python trên Linux và cả MacOS nên **không** khuyến khích dùng.

Bài 9: Giới thiệu Matplotlib

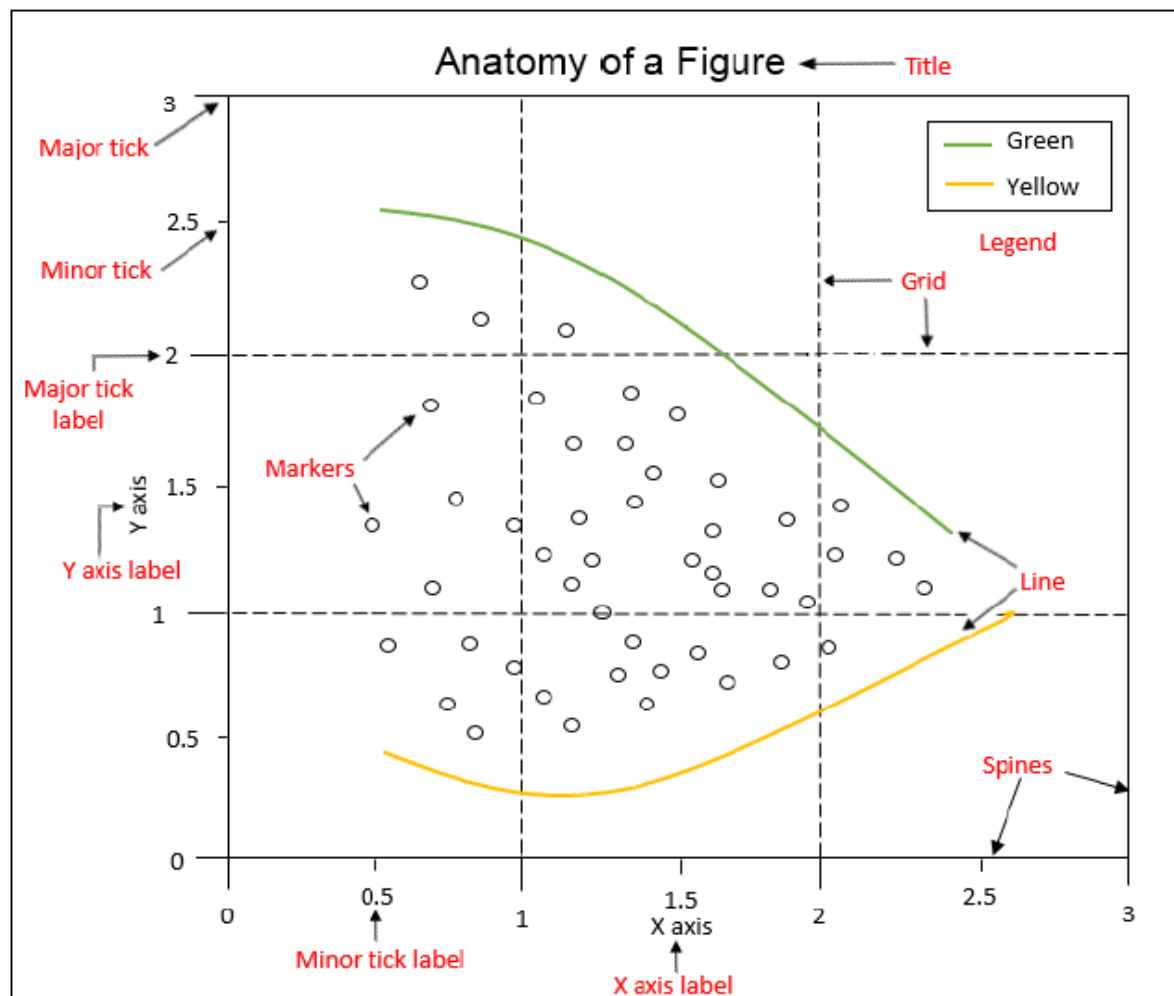
Trong ngày đầu tiên bạn đã làm quen với các loại biểu đồ và đã biết thư viện `matplotlib`. Tuy nhiên bài này sẽ dành riêng để tìm hiểu về thư viện vẽ biểu đồ phổ biến trong Python này. Matplotlib rất phổ biến trong giới data science (khoa học dữ liệu) và machine learning (máy học). Matplotlib được John Hunter phát triển từ năm 2003, lấy ý tưởng từ phần mềm nổi tiếng MATLAB.

Cốt lõi của Matplotlib

Matplotlib xem biểu đồ gồm có 2 thành phần chính:

- **Figure:** Figure được xem như là khung vải mà họa sĩ chuẩn bị để vẽ một bức tranh
- **Axes:** Axes là đối tượng cần vẽ, giống như nội dung bức tranh. Trong bức tranh này có **trục x**, **trục y**, các giá trị cần thể hiện trong không gian x,y (**Markers, Lines, Grid**); các thành phần khác để trang trí như: **tên trục x** (x axis label), **tên trục y** (y axis label), **tiêu đề** bức tranh (title), **ghi chú** (Legend).

Trên hai trục xy thì có thêm các **kí hiệu chia đơn vị chính** (Major stick), **nhãn giá trị đơn vị chính** (Major stick label); **kí hiệu chia đơn vị phụ** (Minor stick), **nhãn giá trị đơn vị phụ** (Minor stick label)



Sub module pyplot của Matplotlib

Module `pyplot` sẽ giúp chúng ta vẽ các biểu đồ mà không cần tốn nhiều thì giờ cho việc trang trí (sử dụng `Figure` và `Axes`).

Để sử dụng sub module `pyplot` của `matplotlib` thì dùng cú pháp sau:

```
import matplotlib.pyplot as plt
```

Nạp thư viện `pyplot` với tên viết tắt (alias) `plt`.

Tạo figure - tạo khung tranh

Đầu tiên là gọi hàm `.figure()` để tạo ra đối tượng `Figure`:

```
fig = plt.figure()
```

<Figure size 432x288 with 0 Axes><Figure size 432x288 with 0 Axes>

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

Mặc định Python sẽ tạo ra bức tranh có kích thước 432 x 288 (tương ứng width x height). Kích thước này tương ứng với 6.4 inches chiều rộng và 4.8 inches chiều ngang với dpi là 100⁽⁵⁾.

Để thay đổi kích thước của biểu đồ thì truyền thêm tham số `figsize`:

```
# Thiết lập bề rộng và chiều cao
fig = plt.figure(figsize=(10, 5))
# Thiết lập dpi: Số điểm ảnh trên một đơn vị Inch
fig = plt.figure(dpi=300)
```

Out: <Figure size 1800x1200 with 0 Axes><Figure size 720x360 with 0 Axes>
<Figure size 1800x1200 with 0 Axes>

Đóng figure - đóng khung tranh

Đối tượng `figure` được tạo dùng để vẽ tiếp chi tiết bức tranh. Khi không dùng nữa thì gọi hàm `close()` để đóng đối tượng. Tức là hủy đối tượng `figure`.

Ví dụ bạn là họa sĩ trên Python, chuẩn bị bày tám vải ra chuẩn bị vẽ biểu đồ thì có ai đó rủ đi café, đá bóng, tám chuyện thì vội đóng lại. Code Python như sau:

```
import matplotlib.pyplot as plt

# Thiết lập bề rộng và chiều cao
plt.figure(figsize=(10, 5))
# Thiết lập dpi: Số điểm ảnh trên một đơn vị Inch
plt.figure(dpi=300)

plt.close()
```

Lệnh `plt.close()` không có tham số thì mặc định cái `figure` hiện tại sẽ bị hủy (đóng). Nếu có nhiều `figure` được tạo và muốn hủy tất cả thì truyền thêm tham số chuỗi `'all'`, hoặc `"all"` `"all"`)

```
plt.close('all')
```

Nếu muốn đóng một `figure` cụ thể thì chỉ rõ số thứ tự trong tham số `num`:

```
import matplotlib.pyplot as plt
```

⁵ Để chuyển đổi độ phân giải màn hình và dpi (Dots per Inch) sang kích thước thật thì không quá phức tạp. Tuy nhiên, bạn hãy tạm bỏ qua cái này.

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
# Tạo Figure với số 1. Thiết lập bề rộng và chiều cao
plt.figure(num=1, figsize=(10, 5))
# Thiết lập dpi: Số điểm ảnh trên một đơn vị Inch
plt.figure(dpi=300)

# Đóng Figure với số 1
plt.close(1)
```

Cấu trúc format

Format ở đây là định dạng cho biểu đồ.

Format này gồm 3 phần [color][marker][line] được trình bày theo 2 dạng:

- Dạng gọn: 'bo--'
- Dạng đầy đủ: color='blue', marker='o', linestyle='dashed'

Ví dụ Format được sử dụng trong tham số thứ ba của hàm `plot([x], y, [format])` sẽ được giải thích ở mục tiếp theo.

Định dạng marker

Marker là kí hiệu để vẽ biểu đồ. Marker phổ biến là điểm (point) tại giá trị của (x_i, y_j) :


Tra cứu kí hiệu marker tại link:

https://matplotlib.org/stable/api/markers_api.html

Vài marker tiêu biểu:

marker	symbol	description
"."	•	point
", "	.	pixel
"o"	●	circle
"v"	▼	triangle_down
"^"	▲	triangle_up
"<"	◀	triangle_left
">"	▶	triangle_right
"1"	⋿	tri_down
"2"	⋿	tri_up
"3"	⋿	tri_left
"4"	⋿	tri_right
"8"	●	octagon





marker	symbol	description
"s"	■	square
"p"	⬠	pentagon
"P"	⬢	plus (filled)
"*"	★	star
"h"	⬡	hexagon1
"H"	⬢	hexagon2
"+"	+	plus
"x"	×	x
"X"	⊠	x (filled)
"D"	◆	diamond
"d"	◇	thin_diamond
" "		vline

marker	symbol	description
" - "		hline

Định dạng màu sắc

Kí hiệu	Màu	Kí hiệu	Màu
b	blue	c	cyan
r	red	b	black
g	green	w	white
m	magenta	y	yellow

Định dạng loại đường kẻ (styleline) – nối các point

Kí hiệu	Mô tả	Ví dụ
' - '	solid line style	
' - - '	dashed line style	
' - . '	dash-dot line style	
' : '	dotted line style	

Các biểu đồ cơ bản

Vẽ biểu đồ với 2 dãy x, y

Hàm `plot([x], y, [format])` sẽ vẽ biểu đồ gồm các điểm theo tọa độ của x, y. Nếu không có x thì mặc định x sẽ là dãy số 0, 1, 2, ...

[format] là định dạng 3 thông tin: color, marker và styleline. Định dạng này có thể viết tắt gồm các kí hiệu đã mô tả trong phần trên:

```
plt.plot(x, y, 'bo--')
```

hoặc viết đầy đủ theo dạng truyền tham số thông thường:

```
plt.plot(x, y, color='blue', marker='o', linestyle='dashed')
```

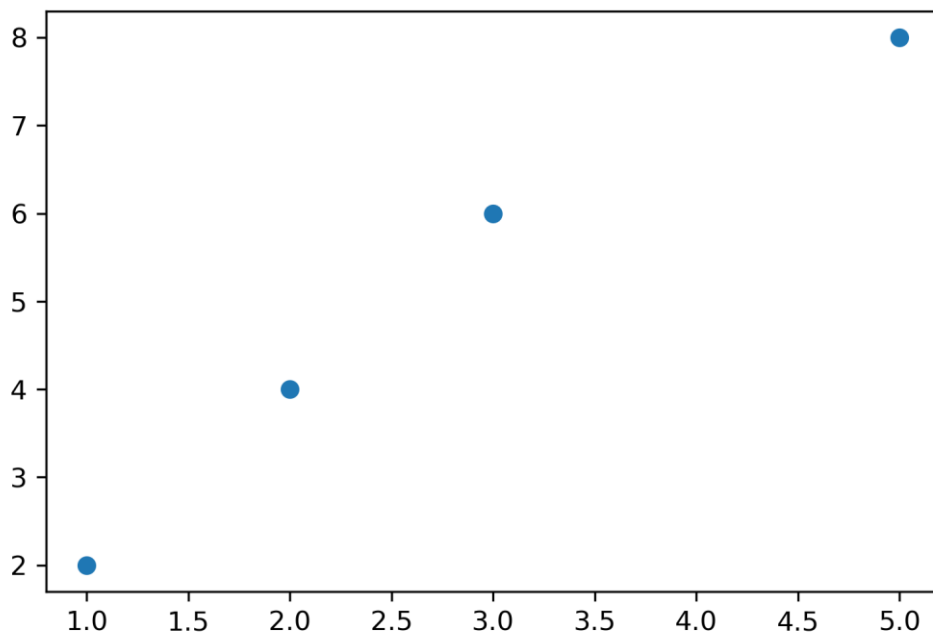
Nhắc lại: kí hiệu [] bao đóng tham số cho biết là không bắt buộc được chỉ định (sẽ sử dụng giá trị mặc định).

```
import matplotlib.pyplot as plt
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

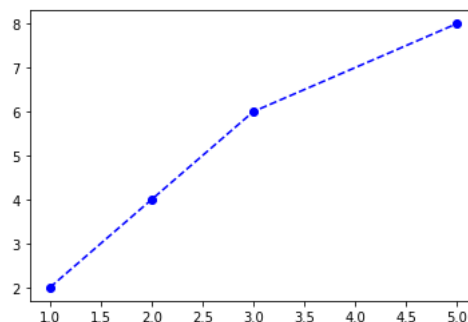
```
# Tạo Figure với số 1. Thiết lập bề rộng và chiều cao
plt.figure(num=1, figsize=(10, 5))
# Thiết lập dpi: Số điểm ảnh trên một đơn vị Inch
plt.figure(dpi=300)

x = [1, 2, 3, 5]
y = [2, 4, 6, 8]
plt.plot(x, y, 'o')
```

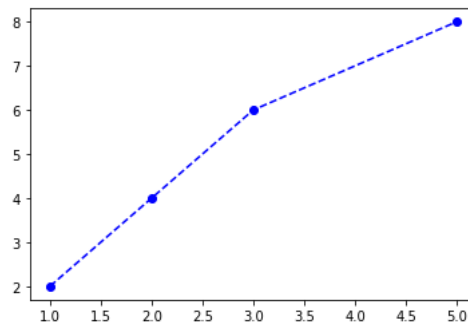


Thử plot với format khác nhau để tự khám phá ý nghĩa các kí hiệu:

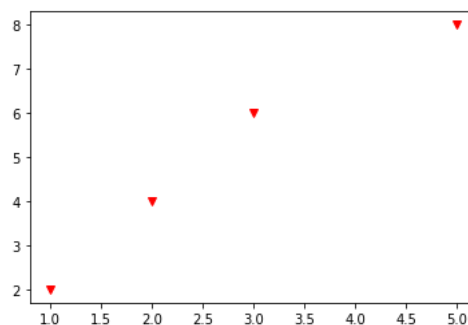
```
plt.plot(x, y, 'bo--')
```



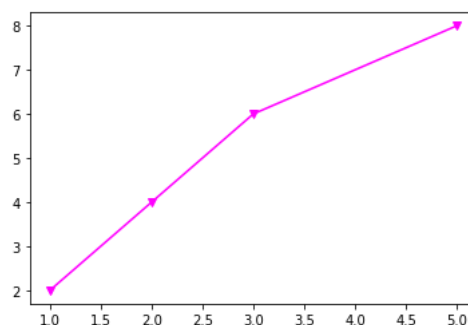
```
plt.plot(x, y, color='blue', marker='o', linestyle='dashed')
```



```
plt.plot(x, y, 'rv')
```



```
plt.plot(x, y, color='magenta', marker='v')
```



Lệnh này truyền 2 tham số `color` và `marker`, không truyền tham số `styleline` thì mặc định có kẻ đường nối giữa các point kề nhau.

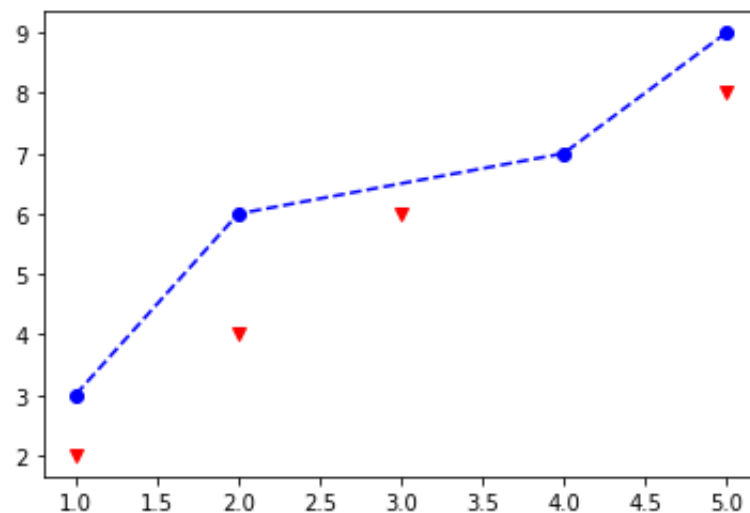
Vẽ biểu đồ với nhiều dãy x, y

Có thể mở rộng gồm 2 bộ tham số:

```
x1 = [1, 2, 3, 5]
y1 = [2, 4, 6, 8]

x2 = [1, 2, 4, 5]
y2 = [3, 6, 7, 9]

plt.plot(x1, y1, 'rv', x2, y2, 'bo--')
```

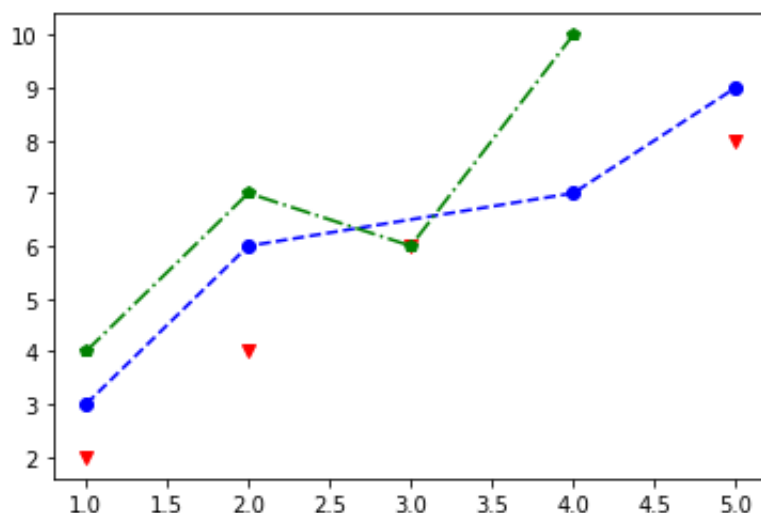
Với 3 bộ tham số:

```
x1 = [1, 2, 3, 5]
y1 = [2, 4, 6, 8]

x2 = [1, 2, 4, 5]
y2 = [3, 6, 7, 9]

x3 = [1, 2, 3, 4]
y3 = [4, 7, 7, 10]

plt.plot(x1, y1, 'rv', x2, y2, 'bo--', x3, y3, 'gp-.')
```



Vẽ biểu đồ line với data frame

Dùng thư viện `matplotlib.pyplot` kết hợp với thư viện `pandas.DataFrame` với cấu trúc sau:

```
plt.plot(x_key, y_key, data=df)
```

Quay lại ví dụ trong Bài 7, vẽ biểu đồ tăng trưởng GDP và CPI bằng cách tự tạo data frame như sau:

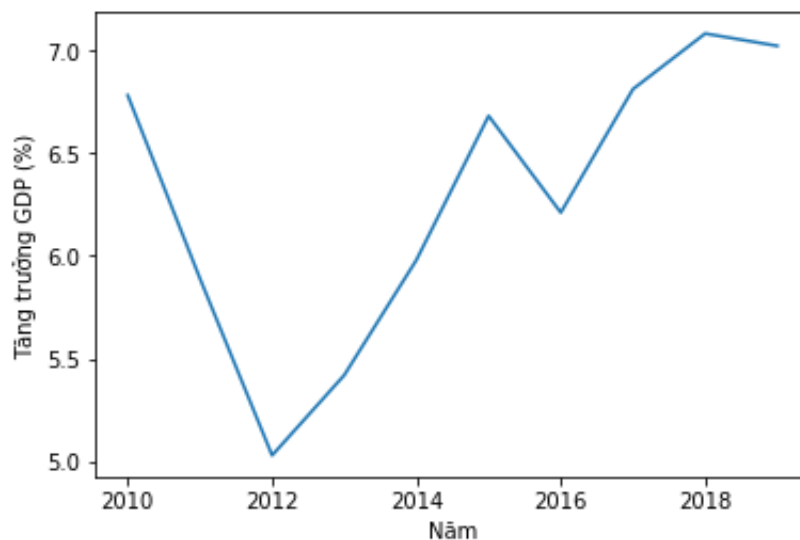
```
import pandas as pd
import matplotlib.pyplot as plt

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year})

plt.figure().suptitle('Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019')
plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP (%)')
plt.plot('year', 'gdp', data = df)
```

Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019



Bạn để ý so với Bài 10 thì khác ở dòng cuối cùng: vẽ với trục x là giá trị cột tên 'year', trục y là giá trị cột tên 'gdp', với data frame là **df**.

Vẽ nhiều biểu đồ

Để hỗ trợ vẽ nhiều biểu đồ trong một “bức tranh” thì Matplotlib cung cấp hàm `.subplots()`.

Khởi tạo Figure và Axe

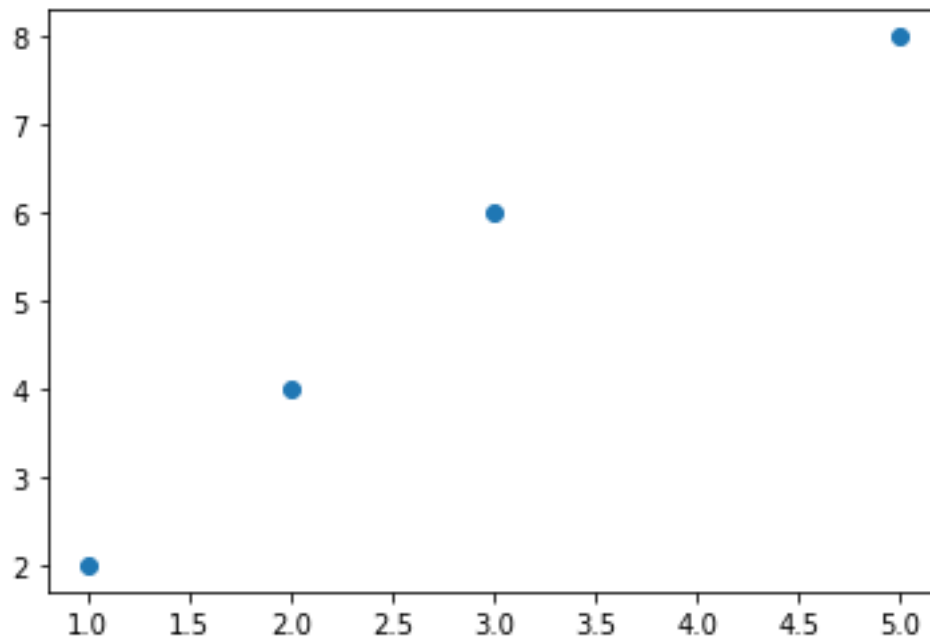
Lệnh Python để gọi tạo Figure và Axe từ hàm `.subplots()` như sau:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
```

Dùng Axe để vẽ 1 biểu đồ

Vẽ biểu đồ từ 2 dãy số:

```
x = [1, 2, 3, 5]
y = [2, 4, 6, 8]
ax.plot(x, y, 'o')
```



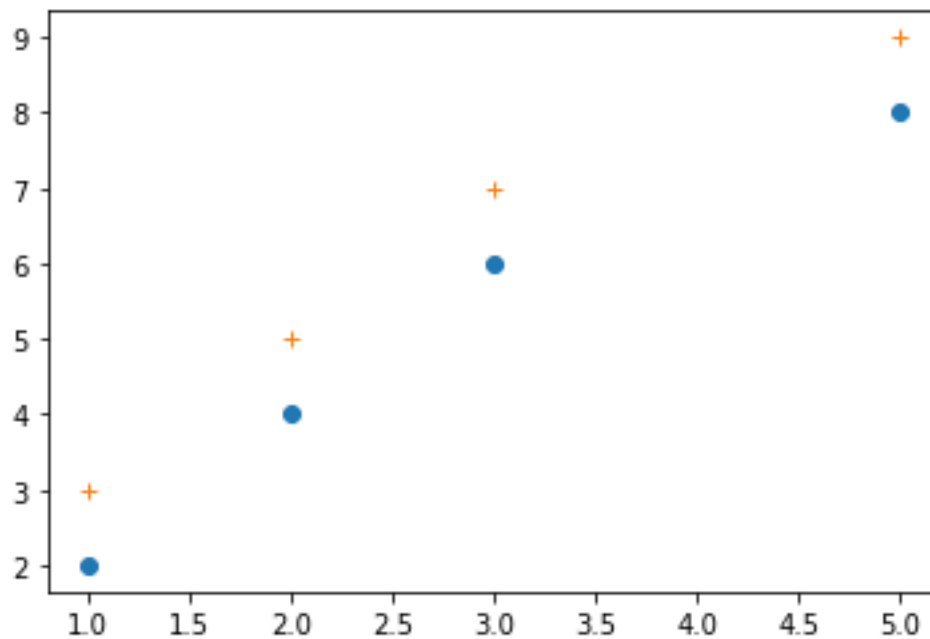
Dùng Axe để vẽ nhiều biểu đồ

Bạn hình dung có một bức tranh (Figure) mà trong đó có 2 biểu đồ được vẽ từ 2 bộ dãy số (x, y) và (x1, y1). Bạn có thể tự thêm các dãy số mới và gọi hàm plot với mark

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()

x = [1, 2, 3, 5]
y = [2, 4, 6, 8]
ax.plot(x, y, 'o')
```

```
x1 = x
y1 = [3, 5, 7, 9]
ax.plot(x1, y1, '+')
```



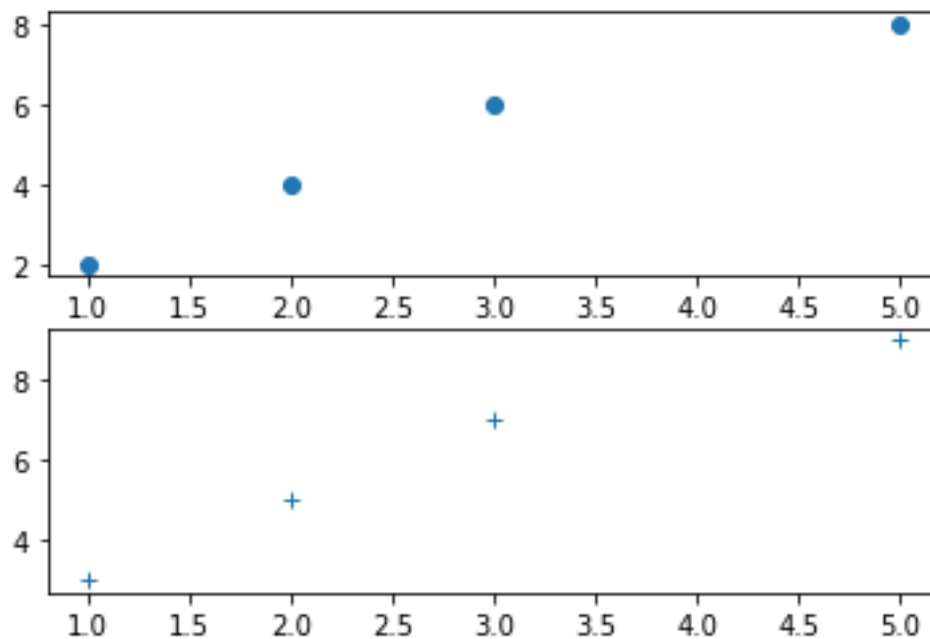
Vẽ biểu đồ trên nhiều Figure

Đoạn code sau tạo ra biểu đồ gồm 2 Figures bằng cách gọi hàm `.subplot (số lượng figure)` từ module `matplotlib.pyplot`:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(2)

x = [1, 2, 3, 5]
y = [2, 4, 6, 8]
ax[0].plot(x, y, 'o')

x1 = x
y1 = [3, 5, 7, 9]
ax[1].plot(x1, y1, '+')
```



Trang trí biểu đồ

Thêm tên biểu đồ

Có thể dùng một trong 2 lệnh sau:

- `plt.figure().suptitle('Tiêu đề')`
- `plt.title('Tiêu đề')`

Thêm tên trục x, trục y

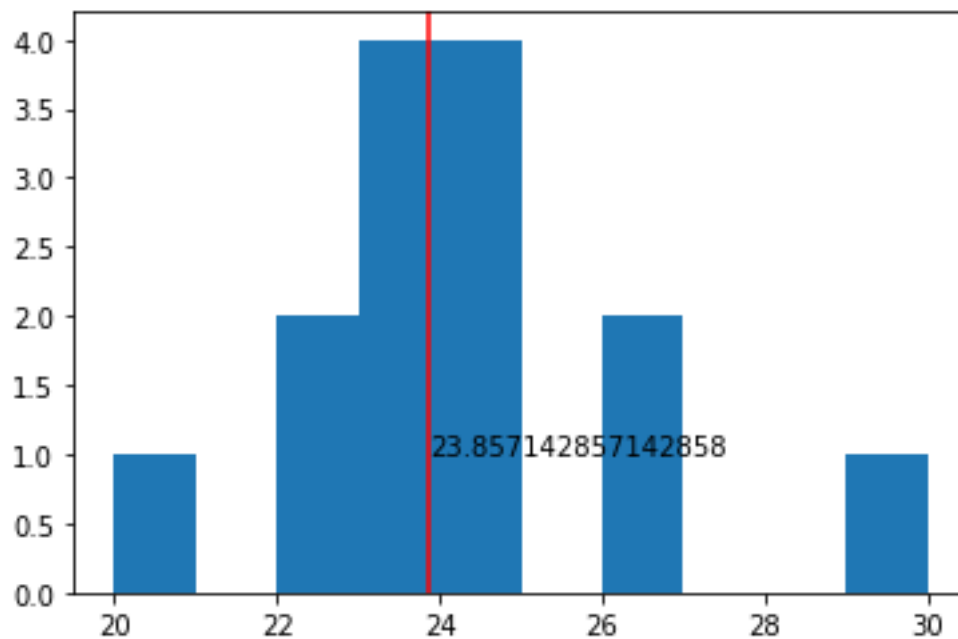
```
plt.xlabel('Nhãn trục x')  
plt.ylabel('Nhãn trục y')
```

Vẽ thêm text tại vị trí x, y

Sử dụng hàm `.text(x, y, value)` để hiển thị một giá trị tại vị trí x,y:

Ví dụ vẽ thêm giá trị trung bình

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')  
plt.hist(df['age'])  
plt.axvline(df['age'].mean(), color='red')  
plt.text(df['age'].mean(), 1, df['age'].mean())
```



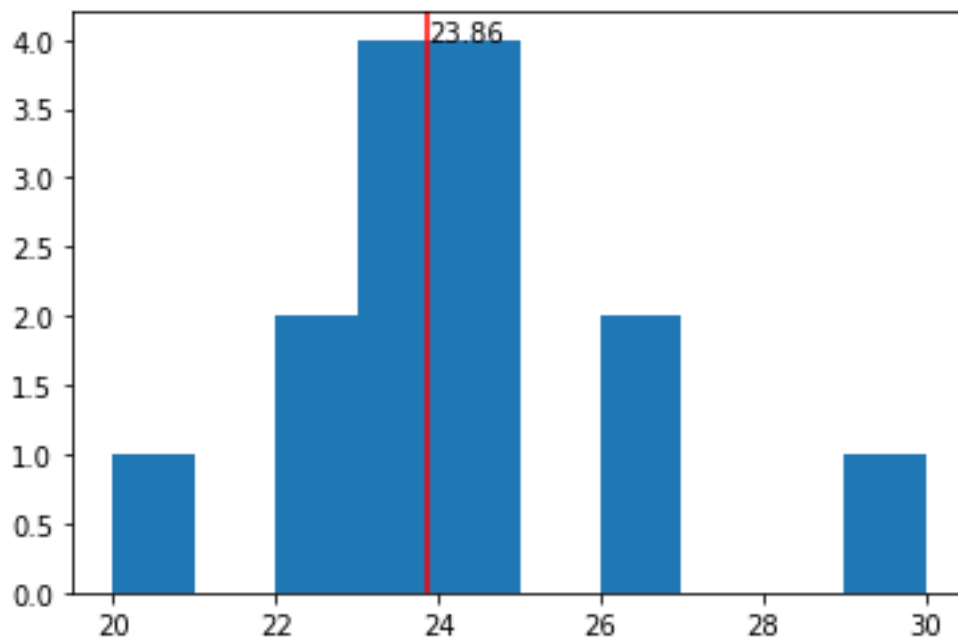
Cải tiến một chút cho biểu đồ:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')
hist = plt.hist(df['age'])

age_mean = df['age'].mean()
plt.axvline(age_mean, color='red')

plt.text(age_mean, hist[0].max(), '%.2f' % age_mean)
```



Việc giải thích những chỗ thay đổi (bôi đậm) thì dành cho bạn nhé!

Thêm legend

Để giải thích thêm cho từng loại dữ liệu trong biểu đồ thì khi plot kèm theo tham số label. Tiếp theo gọi hàm legend(). Xem phần in đậm trong đoạn chương trình sau:

```
import pandas as pd
import matplotlib.pyplot as plt

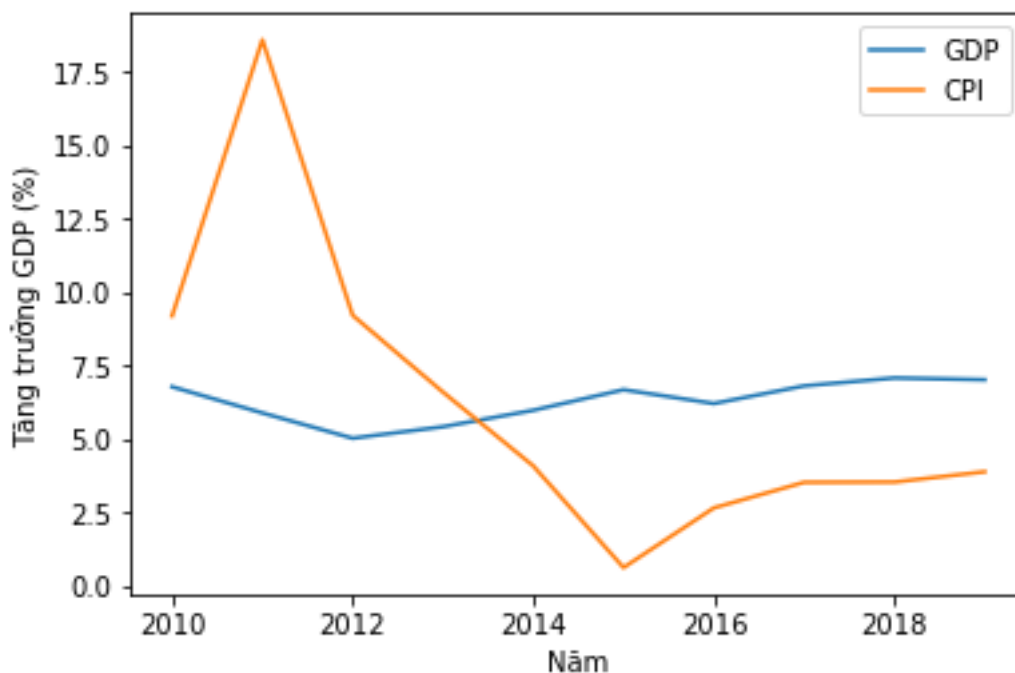
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]
cpi = [9.19, 18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54, 3.89]

# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year, 'cpi': cpi})

plt.figure()
plt.title('Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019')

plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP (%)')
plt.plot(df['year'], df['gdp'], label = 'GDP')
plt.plot(df['year'], df['cpi'], label = 'CPI')
plt.legend()
```

Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019



Vẽ biểu đồ line với data frame từ file CSV

Trong trường hợp bạn có sẵn file CSV thì có thể đọc dữ liệu vào data frame với thư viện pandas và vẽ biểu đồ cho hai cột dữ liệu đơn giản như sau:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')
df.head()

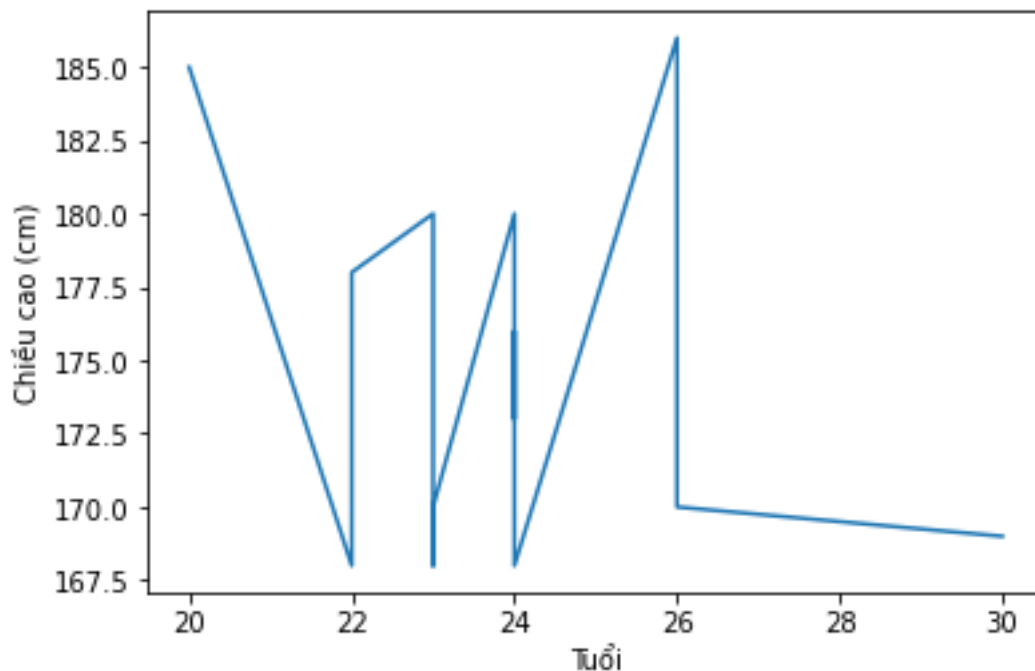
# Sắp xếp dữ liệu theo tuổi
df = df.sort_values(by = 'age')

plt.figure().suptitle('Biểu đồ tuổi và chiều cao của nam tuyển thủ bóng đá.')
plt.xlabel('Tuổi')
plt.ylabel('Chiều cao (cm)')

plt.plot('age', 'height', data = df)
```

Trong đoạn code trên có dùng hàm `df.sort_values(by = 'age')` để sắp xếp lại dữ liệu theo cột age. Kết quả biểu đồ:

Biểu đồ tuổi và chiều cao của nam tuyển thủ bóng đá.



Vẽ biểu đồ theo nhu cầu quan sát dữ liệu

Đọc dữ liệu

Đọc dữ liệu từ nghiên cứu về sức khỏe, luyện tập lại một số lệnh, kỹ thuật:

- Thêm cột mới cho DataFrame.
- Dùng lệnh `print` để hiển thị thông tin về vài dòng dữ liệu, thông tin về tên cột.

```
import pandas as pd
fp = 'https://thachln.github.io/datasets/sample_health_vn.csv'
df = pd.read_csv(fp)
df['whtr'] = df['waist'] / df['height']

print(df.head())
print(df.columns)
```

```
   id  age  sex  height  waist  risk  weight  hit  life
0   1   23    0   148    69    0.0    38    77  77.50
1   2   26    1   171    82    0.0    57    95  75.37
2   3   66    1   164    86    0.0    77    89  66.09
3   4   55    1   170    89    0.0    73    90  69.59
4   5   30    0   154    76    0.0    59    83  70.01
Index(['id', 'age', 'sex', 'height', 'waist', 'risk', 'weight', 'hit', 'life',
      'whtr'],
      dtype='object')
```

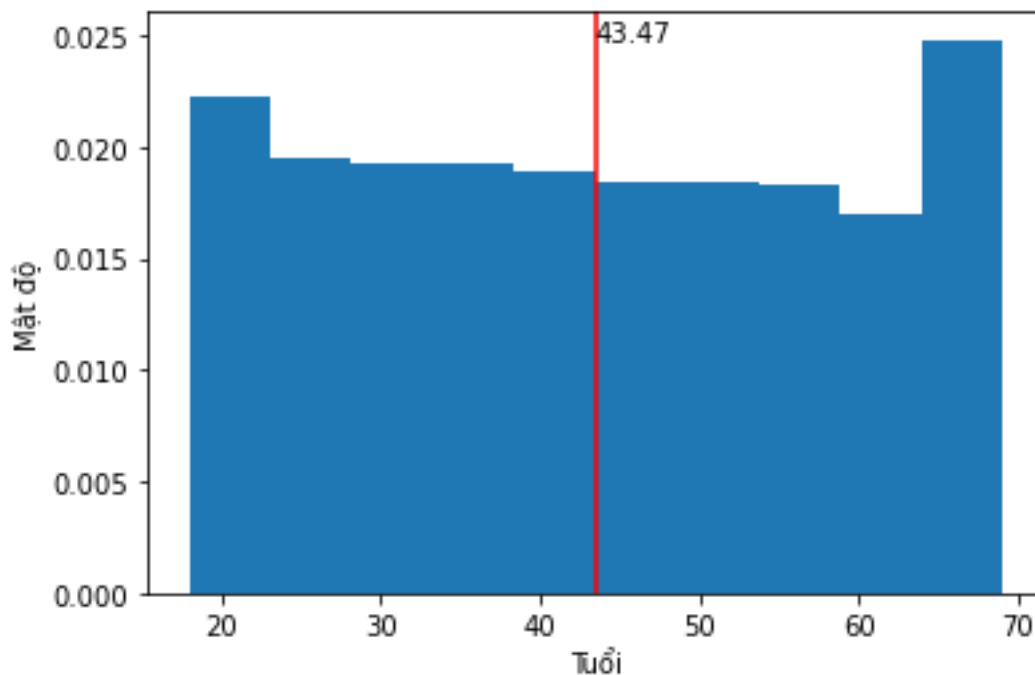
Xem phân bố của 1 biến với hàm .hist

```
import matplotlib.pyplot as plt
hist = plt.hist(df['age'], density=True)
plt.xlabel('Tuổi')
plt.ylabel('Mật độ')

age_mean = df['age'].mean()
plt.axvline(age_mean, color='red')

plt.text(age_mean, hist[0].max(), '%.2f' % age_mean)

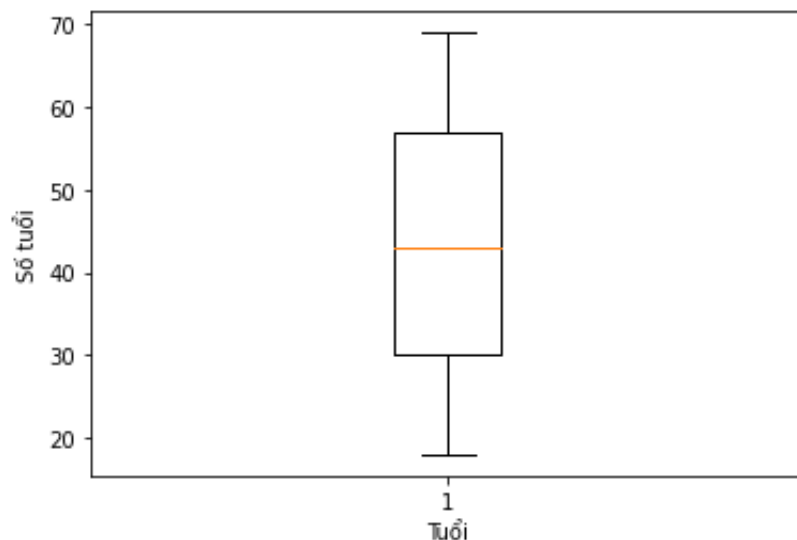
plt.show()
```



Xem dữ liệu tuổi với boxplot

```
import matplotlib.pyplot as plt
plt.boxplot(df['age'])
plt.xlabel('Tuổi')
plt.ylabel('Số tuổi')

plt.show()
```



Tham khảo thêm:

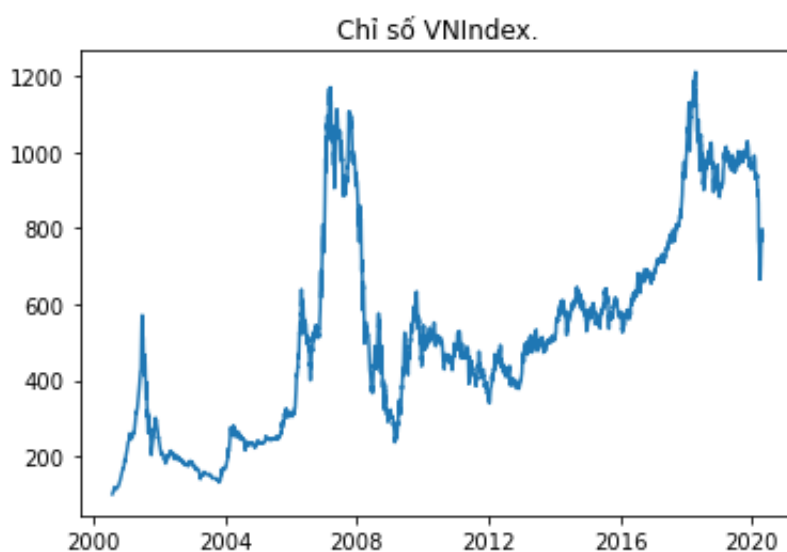
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.boxplot.html

Xem dữ liệu theo thời gian

Sử dụng hàm `.plot` với trục x là thời gian

```
import pandas as pd
import matplotlib.pyplot as plt
fp = 'https://thachln.github.io/datasets/vnindex_20200424.txt'
df1 = pd.read_csv(fp)

df1['date'] = pd.to_datetime(df1['<DTYYYYMMDD>'], format='%Y%m%d')
plt.plot(df1['date'], df1['<High>'])
plt.title('Chỉ số VNIndex.')
```



Sử dụng hàm `.plot_date`

```
import pandas as pd
import matplotlib.pyplot as plt

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

# Tạo data frame
df1 = pd.DataFrame({'gdp': gdp, 'year': year})
df1.index = df1['year']
plt.title('Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019')
plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP (%)')

df1['date'] = pd.to_datetime(df1['year'], format='%Y')

plt.gcf().autofmt_xdate()
plt.plot_date(df1['date'], df1['gdp'], linestyle='solid')
```



Ghi chú:

- Lệnh `plt.gcf().autofmt_xdate()` sẽ tự động định dạng độ nghiêng cho nhãn trên trục x sao cho không chồng lên nhau.

So sánh 2 biến với biểu đồ bar

Đếm số lượng nam và nữ trong dữ liệu của một nghiên cứu và vẽ biểu đồ bar:

```
import pandas as pd
fp = 'https://thachln.github.io/datasets/sample_health_vn.csv'
df = pd.read_csv(fp)
df['whtr'] = df['waist'] / df['height']

df.loc[df['sex'] == 0, 'sex_label'] = 'Nữ'
df.loc[df['sex'] == 1, 'sex_label'] = 'Nam'

print(df.head())
print(df.columns)

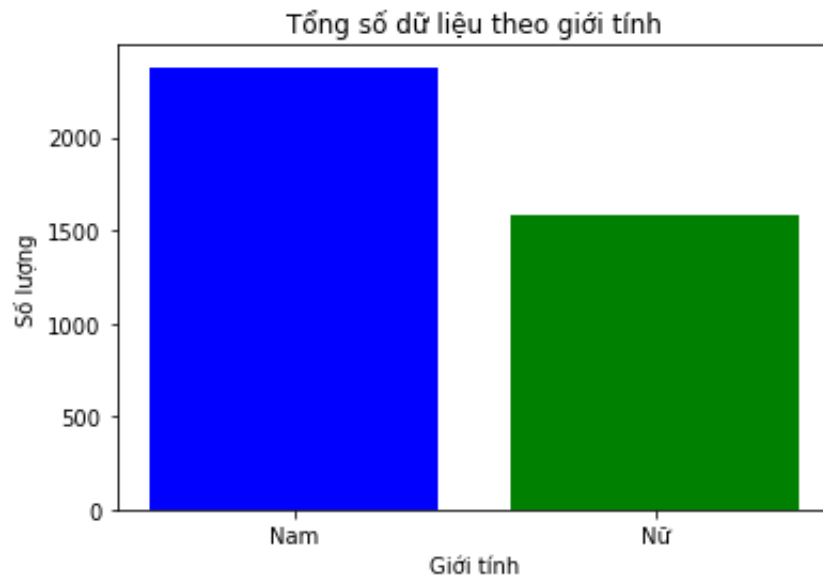
group_sex_age_count =
df.groupby('sex_label')['age'].count().reset_index()
print(group_sex_age_count)

import matplotlib.pyplot as plt

plt.bar(group_sex_age_count['sex_label'], group_sex_age_count['age'],
color=['b', 'g'])
plt.title('Tổng số dữ liệu theo giới tính')
plt.xlabel('Giới tính')
plt.ylabel('Số lượng')

plt.show()
```

```
id age sex height waist risk weight hit life whtr sex_label
0 1 23 0 148 69 0.0 38 77 77.50 0.466216 Nữ
1 2 26 1 171 82 0.0 57 95 75.37 0.479532 Nam
2 3 66 1 164 86 0.0 77 89 66.09 0.524390 Nam
3 4 55 1 170 89 0.0 73 90 69.59 0.523529 Nam
4 5 30 0 154 76 0.0 59 83 70.01 0.493506 Nữ
Index(['id', 'age', 'sex', 'height', 'waist', 'risk', 'weight', 'hit', 'life',
      'whtr', 'sex_label'],
      dtype='object')
sex_label age
0 Nam 2377
1 Nữ 1583
```



So sánh 1 biến theo thời gian

Đoạn chương trình bên dưới vẽ 2 bức tranh trong cùng một biểu đồ. Mỗi bức tranh là một biểu đồ line để theo dõi giá trị (GDP hoặc CPI) theo thời gian.

```
import pandas as pd
import matplotlib.pyplot as plt

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
cpi = [9.19, 18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54, 3.89]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

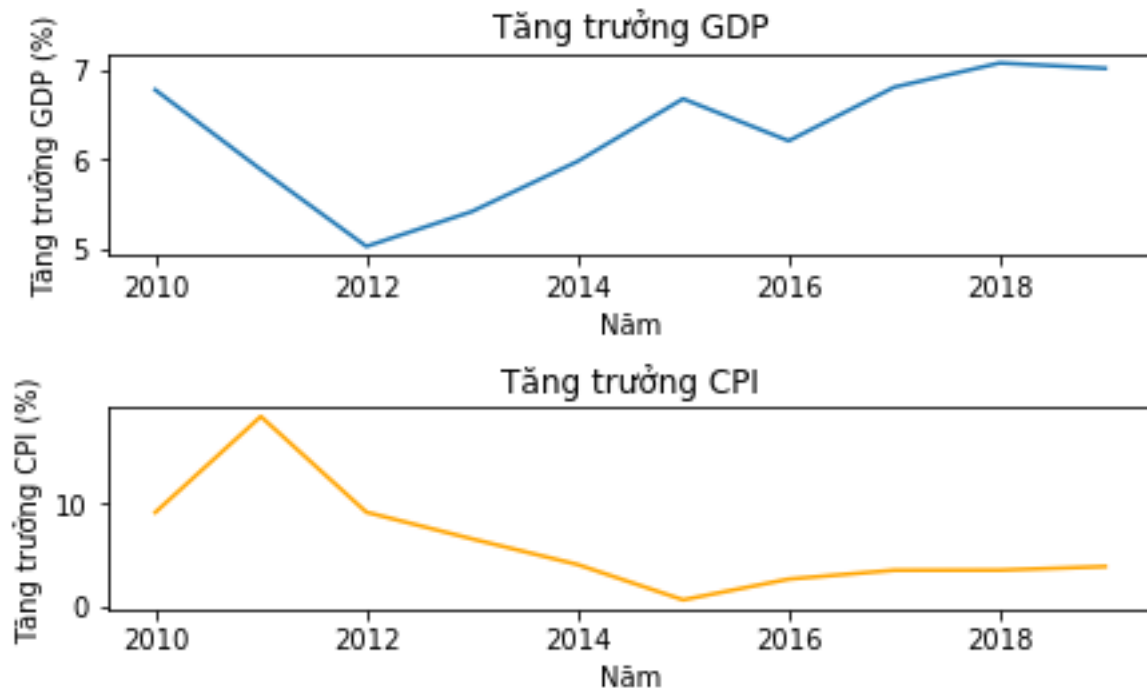
# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year, 'cpi': cpi})

fig, (ax1, ax2) = plt.subplots(2, constrained_layout=True)
ax1.set_title('Tăng trưởng GDP')
ax1.set_xlabel('Năm')
ax1.set_ylabel('Tăng trưởng GDP (%)')
ax1.plot(df['year'], df['gdp'])

ax2.set_title('Tăng trưởng CPI')
ax2.set_xlabel('Năm')
ax2.set_ylabel('Tăng trưởng CPI (%)')
ax2.plot(df['year'], df['cpi'], color='orange')
```

```
fig.suptitle('Biểu đồ tăng trưởng GDP và CPI của Việt Nam từ năm 2010  
đến 2019')  
  
plt.show()
```

Biểu đồ tăng trưởng GDP và CPI của Việt Nam từ năm 2010 đến 2019

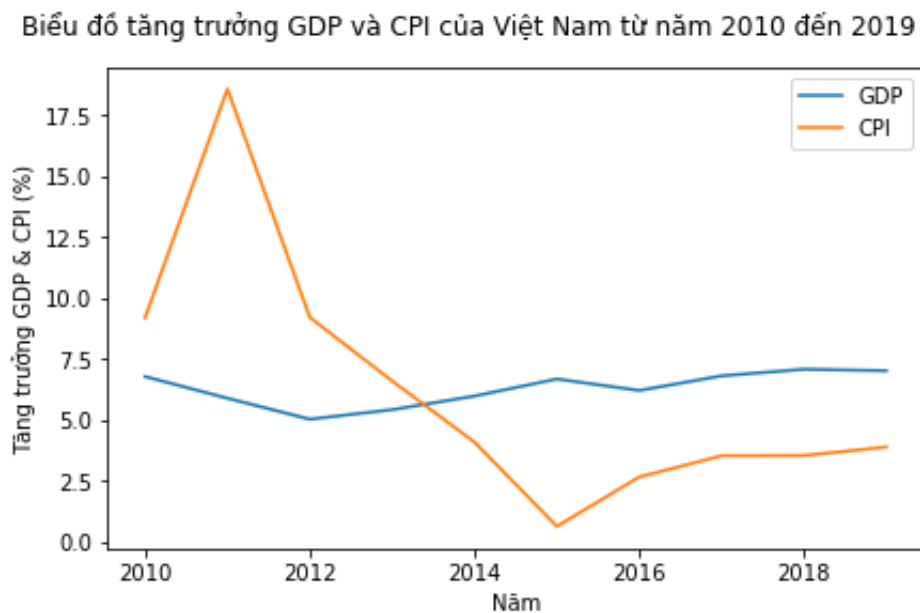


So sánh 2 biến theo thời gian

Cải tiến đoạn chương trình ở trên 1 chút để vẽ 2 đường tăng trưởng GDP, CPI trong cùng một biểu đồ.

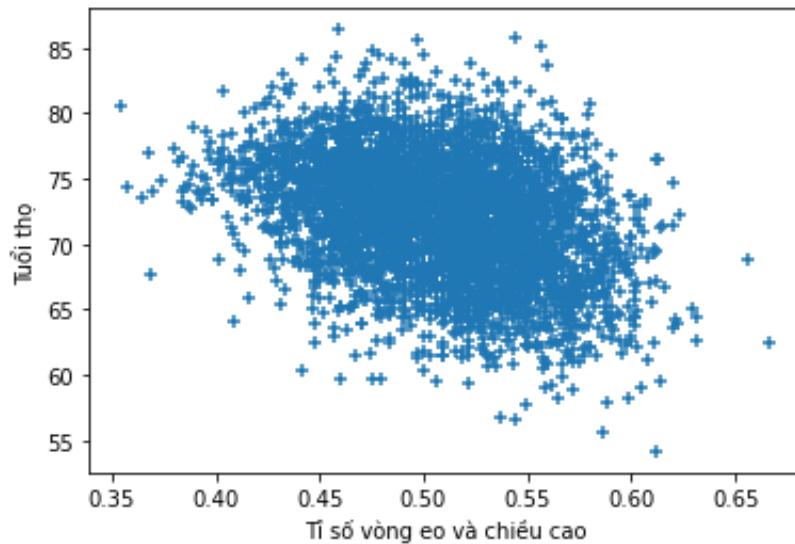
```
import pandas as pd  
import matplotlib.pyplot as plt  
  
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]  
cpi = [9.19, 18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54, 3.89]  
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]  
  
# Tạo data frame  
df = pd.DataFrame({'gdp': gdp, 'year': year, 'cpi': cpi})  
  
fig, ax = plt.subplots(1, constrained_layout=True)  
ax.plot(df['year'], df['gdp'], label = 'GDP')  
ax.plot(df['year'], df['cpi'], label = 'CPI')
```

```
fig.suptitle('Biểu đồ tăng trưởng GDP và CPI của Việt Nam từ năm 2010  
đến 2019')  
plt.xlabel('Năm')  
plt.ylabel('Tăng trưởng GDP & CPI (%)')  
  
plt.legend()  
plt.show()
```



Xem tương quan giữa 2 biến với hàm .scatter

```
import pandas as pd  
import matplotlib.pyplot as plt  
fp = 'https://thachln.github.io/datasets/sample_health_vn.csv'  
df = pd.read_csv(fp)  
df['whtr'] = df['waist'] / df['height']  
  
plt.scatter(df['whtr'], df['life'], marker='+')  
# Gọi scatter với DataFrame: x, y là tên cột  
# plt.scatter('whtr', 'life', data=df, marker='+')  
plt.xlabel('Tỉ số vòng eo và chiều cao')  
plt.ylabel('Tuổi thọ')  
plt.show()
```

Tham khảo thêm:

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.scatter.html

Lưu biểu đồ

Hàm `plt.savefig(fname)` sẽ lưu Figure hiện hành. Các tham số tùy chọn gồm `dpi`, `format`, `transparent`.

```
import pandas as pd
import matplotlib.pyplot as plt

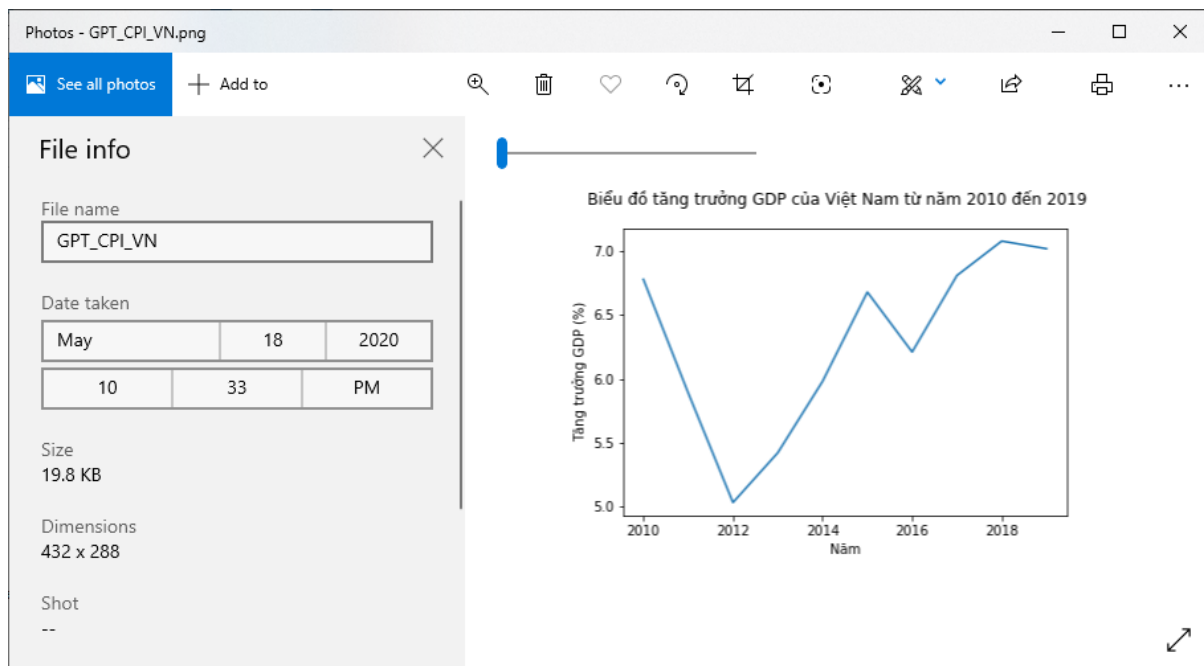
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year})

plt.figure().suptitle('Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019')
plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP (%)')
plt.plot('year', 'gdp', data = df)
plt.savefig("D:/GPT_CPI_VN.png")
```

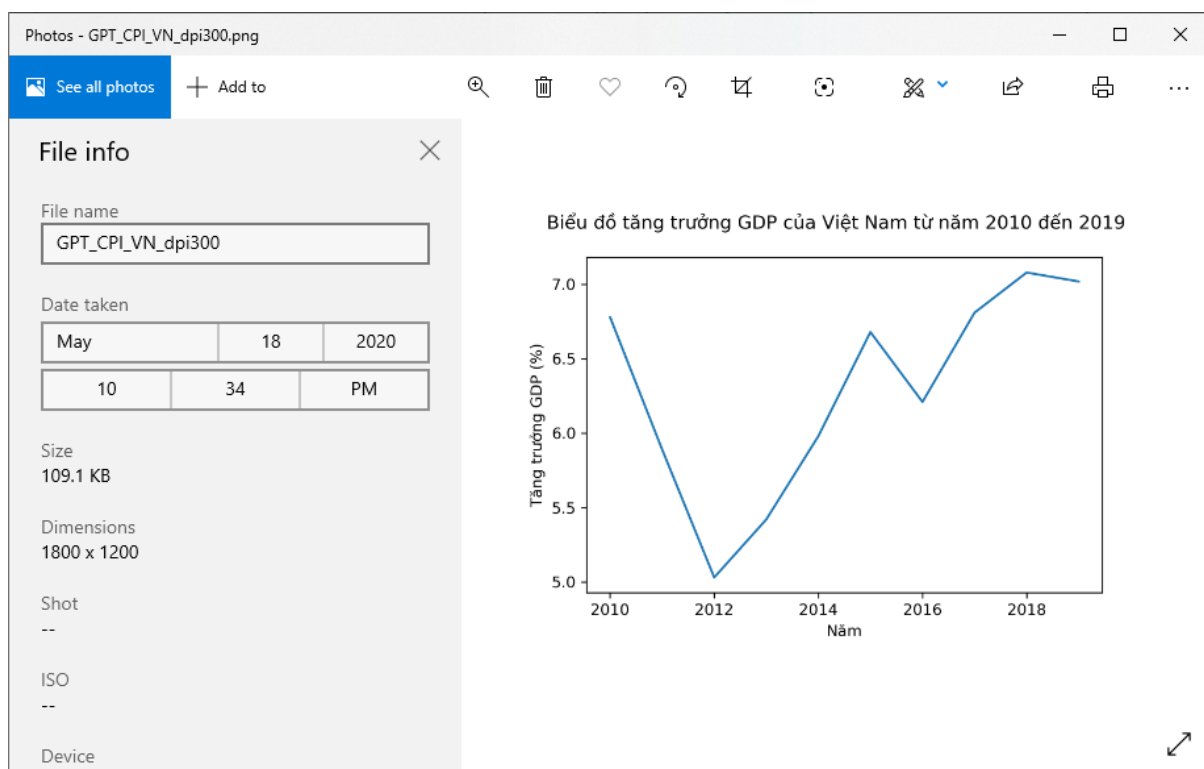
Kết quả:

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



Thử sửa lại dòng cuối cùng

```
plt.savefig("D:/GPT_CPI_VN_dpi300.png", dpi=300)
```



Hãy quan sát kích thước ảnh và Dimensions khác nhau giữa 2 lệnh trên.

Thử thêm tham số `transparent` và kiểm tra tính trong suốt (`transparent`) của ảnh:

```
plt.savefig("D:/GPT_CPI_VN_0.png", transparent = 0)
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
plt.savefig("D:/GPT_CPI_VN_1.png", transparent = 0.1)
plt.savefig("D:/GPT_CPI_VN_2.png", transparent = 0.5)
plt.savefig("D:/GPT_CPI_VN_3.png", transparent = 1)
```