

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 mời 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** 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 Mono hoặc Ngân hàng (dự kiến 31/12/2021 sẽ có bảng full của eBook):

① MoMo	② Chuyển khoản
<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 APITHON 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>LÊ NGỌC THACH 0908550642</p> 	<p>Thanh toán qua NH Tiên Phong</p> <p>Lê Ngọc Thạch, Ngân Hàng Tiên Phong, CN HCM Số tài khoản: 00002888001</p> <p>Nội dung tin nhắn: email sdt APITHON Vd tin nhắn: abc@gmail.com 0908456321 APITHON</p> <p>Quét mã QR để thanh toán cho:</p> <p>LE NGOC THACH 0000 2888 001</p> 

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

Mục lục

Quy ước	7
Ngày 1 – Chủ đề: Giới thiệu về ngôn ngữ thống kê, ngôn ngữ lập trình	11
Bài 1: Tóm tắt về thống kê (Statistics)	13
Bài 2: Ngôn ngữ lập trình Python	20
Bài 3: Ngôn ngữ Python và phần mềm Anaconda	28
Bài 4: Cài đặt thêm phần mềm	50
Bài 5: Nhập liệu, biên tập, lưu trữ dữ liệu với Python	55
Bài tập ngày 1	73
Thử thách cho bạn!	75
Ngày 2 – Chủ đề: Biểu đồ	76
Bài 6: Các loại biểu đồ	78
Bài 7: Vẽ biểu đồ trong Python	84
Bài 8: Nguyên tắc soạn biểu đồ	101
Bài 9: Giới thiệu Matplotlib	103
Bài 10: Giới thiệu Bokeh	129
Bài 11: Khai phá Bokeh	137
Ngày 3 – Phân tích mô tả	163
Bài 12: Phân tích mô tả dữ liệu Bank Marketing	165
Bài 13: Phân tích dữ liệu Marketing #2	178
Bài 14: So sánh 2 tỉ lệ	189
Bài 15: Mô hình kiểm định giả thuyết	200
Bài 16: Ứng dụng minh họa kiểm định giả thuyết	201
Bài 17: Phân tích mối tương quan	211
Ngày 4 – Chủ đề: Dữ liệu lớn	219
Bài 18: Cách xử lý tập hợp dữ liệu lớn	220
Bài 19: Sử dụng Ubuntu	255
Bài 20: Cài đặt Hadoop 3.2	264
Bài 21: Trải nghiệm Hadoop với Python	272
Ngày 5 – Chủ đề: Dự báo bằng mô hình hồi qui tuyến tính	278

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	279
Bài 23: Diễn giải mô hình hồi qui tuyến tính.....	284
Bài 24: Mô hình hồi qui tuyến tính đa biến.....	297
Bài 25: Dự báo bằng mô hình hồi qui tuyến tính	303
Ngày 6 – Chủ đề: Dự báo bằng mô hình hồi qui logistic	307
Bài 26: Giới thiệu mô hình hồi qui logistic.....	308
Bài 27: Mô hình hồi qui logistic đa biến (Multiple logistic regression model).....	312
Bài 28: So sánh mô hình.....	316
Bài 29: Dự báo bằng mô hình hồi qui logistic	323
Ngày 7 – Chủ đề: Phân tích đa biến	330
Bài 30: Xử lý giá trị trống	331
Bài 31: Mô hình phân tích phân định (Linear discriminant analysis)..	335
Bài 32: Mô hình thành phần (Principal Component Analysis)	343
Bài 33: Mô hình phân tích cụm/nhóm (cluster analysis)	351
Ngày 8 – Chủ đề: Machine Learning	359
Bài 34: Giới thiệu Machine learning	360
Bài 35: Mô hình SVM	362
Bài 36: Mô hình Random Forest	370
Bài 37: Mô hình Artificial Neural Network	374
Bài 38: Machine Learning với Python Tensorflow	380
Ngày 9 – Chủ đề: Recommendation.....	409
Bài 39: Giới thiệu phương pháp gợi ý Collaborative filtering	410
Bài 40: Triển phương pháp gợi ý Collaborative filtering bằng R	420
Ngày 10 – Chủ đề: Natural Language Processing.....	426
Bài 41: Các kỹ thuật cơ bản	427
Bài 42: Trích đặc trưng (Feature extraction).....	432
Bài 43: Giới thiệu ứng dụng phân tích cảm xúc (Sentiment Analysis)	442
Bài 44: Giới thiệu ứng dụng phân tích từ vựng (Word Embedding) ...	453
Bài 45: Giới thiệu ứng dụng xác định chủ đề (Topic Modeling)	465
Ngày 11 – Chủ đề: Computer Vision	476
Bài 46: Giới thiệu Face recognition	477

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

Bài 47: Giới thiệu mô hình CNN	492
Ngày 12 – Chủ đề: Nhận diện tiếng nói (Speech Recognition)	511
Bài 48: Giới thiệu đặc trưng của âm thanh	512
Bài 49: Các thao tác cơ bản với file âm thanh	518
Bài 50: Mô hình Chuyển giọng nói thành văn bản	522
Ngày 13 – Chủ đề: Phân tích dữ liệu theo trường phái Bayes	525
Bài 51: Gia nhập trường phái Bayes	526
Học liệu tham khảo:.....	531
Tạm kết thúc	532
Phụ lục	533
Đóng gói chương trình Python	534
Sử dụng Flask	536
Quan sát giao dịch cổ phiếu VNM (Vinamilk)	540
Đọc và vẽ tín hiệu âm thanh	550
Tải sách nói “Từ tốt đến vĩ đại”	553
Đọc ảnh y khoa DiCOM.....	556
Áp dụng biến đổi Fourier cho ảnh.....	559
Sử dụng Git.....	563
Khảo sát ảnh và ma trận	592
Phát triển ứng dụng với Python.....	594
Xử lý file pdf	601
Khảo sát file âm thanh.....	609
Phân tích âm thanh với thư viện mutagen	612
Khám phá Python trong WSL2	613
Crawl dữ liệu bằng Selenium	615
Sử dụng OpenCV để phân tích dữ liệu ảnh.....	616
Cài đặt OpenCV	617
Đóng gói chương trình Python	618
Tải file video từ Youtube	621
Sinh code Restful API từ database	622
Trải nghiệm Restful API với Flask	624
Trải nghiệm Kafka.....	625

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

Trải nghiệm Apache NiFi.....	628
Giới thiệu superset.....	633
Cài đặt.....	634
Khởi động lại superset.....	639
Cài đặt và cấu hình Nginx	640
Truy cập nginx trên Ubuntu từ máy Windows.....	641
Khai thác superset	643
Cài đặt Ubuntu Server 20.04.2	647
Superset introduction.....	648
Giới thiệu PowerBI.....	649
Bài 1 – Phân tích mô tả với Power BI.....	650
Phân tích biểu đồ với dữ liệu Bank Marketing	662
Minh họa biểu đồ theo thời gian	672
Bài bổ sung	675
Bài 101: Phát triển ứng dụng với PyCharm	676
Bài 102: Xử lý dữ liệu	682
Bài 103: Sử dụng thư viện XGBoost.....	687
Bài 104: Sử dụng môi trường ảo cho Python	688
Bài 105: Sửa code.....	691
Bài 106: Sử dụng Swagger.....	692

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 đọc 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 gấp 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 nháy 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 nháy.

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 ngọn <>. 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

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

“<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 phụ 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 phân đấu đến mức “Tuệ”. Đằng nào thi 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 điều 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.

Chú ý khi copy lệnh từ file pdf

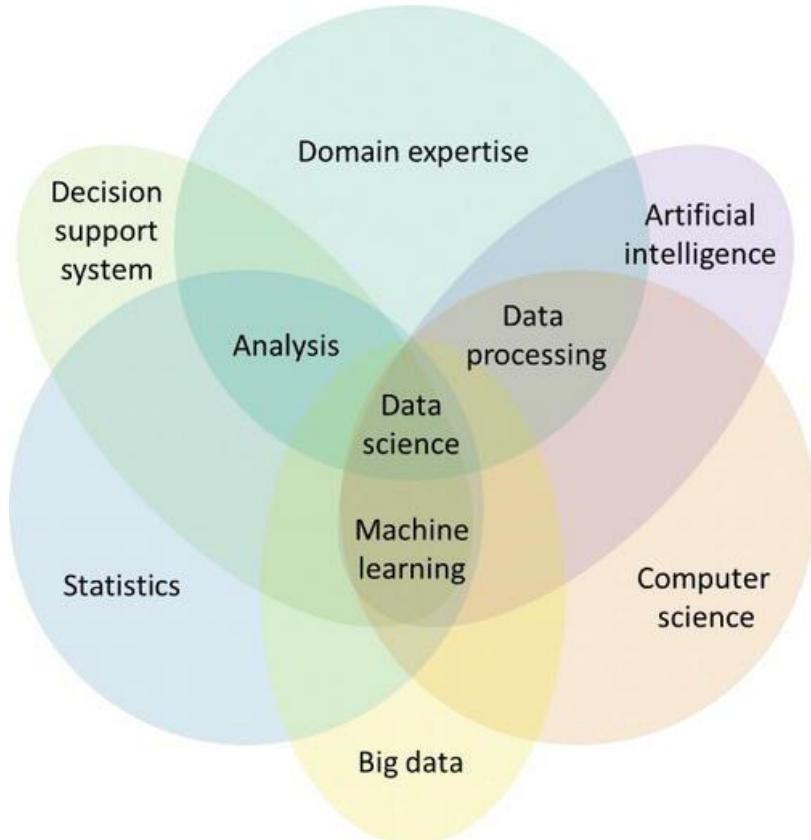
Nếu bạn đang đọc tài liệu này dạng pdf thì khi copy mã nguồn có thể gặp tình trạng sau:

- Đối với một lệnh dài trong tài liệu khi bị xuống dòng thì khi copy và paste từ file pdf ra thì cuối dòng sẽ có thêm khoảng trắng. Vì vậy các từ khi bị rót xuống hàng ở giữa từ thì bị thêm khoảng trắng. Chú ý kiểm tra lại sau khi paste.

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

Ngày 1 - Chủ đề: Giới thiệu về ngôn ngữ thống kê, ngôn ngữ lập trình

Một bức tranh tôi cho là giúp chúng ta có cái nhìn khái quát về các lĩnh vực AI là từ cuốn sách Artificial Intelligence - Scope and Limitations, tạm dịch là “Trí thông minh nhân tạo – Năng lực và Giới hạn”.



Hình 1: Bức tranh về các lĩnh vực liên quan đến AI

(Nguồn: Artificial Intelligence - Scope and Limitations)

AI là một ngành giao thoa giữa rất nhiều lĩnh vực. Trong đó sử dụng phần lớn các kỹ thuật của Khoa học máy tính (Computer science) kết hợp Thống kê (Statistics) và Phân tích dữ liệu (Analysis/Analytics).

Ngày đầu tiên tôi sẽ thảo luận với các bạn vài điểm cơ bản về thống kê để các bạn có cơ hội ôn lại. Đối với các bạn mới thì cũng có đủ kiến thức cơ bản để hiểu và làm quen được nội dung trong ngày này.

Tôi không đi sâu vào các khái niệm về toán học – vốn rất nhức đầu, dành cho giới hàn lâm mà sẽ tập trung vào các khái niệm cơ bản, rất cơ bản để chúng ta làm quen với các công cụ phần mềm như Python và R. Sau ngày đầu tiên này chúng ta sẽ biết hoặc làm được các việc sau:

① Biết hoặc tự mô tả được các khái niệm thống kê vốn được sử dụng phổ biến trong đời sống như:

Giá trị trung bình

Giá trung vị

Giá trị mode

Phương sai

Mối tương quan (correlation)

Các cách để mô tả dữ liệu (**data types**)

② Tự cài phần mềm để thực hành với R hoặc Python. Làm việc với các lệnh cơ bản trong R hoặc Python.

③ Đọc dữ liệu vào Python

Ngày đầu tiên sẽ gồm 5 bài:

Bài 1: Tóm tắt và giúp các bạn nhớ lại, hoặc làm quen với vài khái niệm thống kê đơn giản.

Bài 2: Giới thiệu ngắn gọn về ngôn ngữ Python.

Bài 3: Hướng dẫn làm quen với ngôn ngữ Python và phần mềm để thực hành Anaconda, Spyder.

Bài 4: Chia sẻ thêm các trải nghiệm thực tế để giúp các bạn làm việc trên máy tính thuận tiện hơn.

Bài 5: Hướng dẫn chuẩn bị dữ liệu, các thao tác biên tập cơ bản và lưu trữ dữ liệu với Python. Đặc biệt trong bài 5 là phần “Tinh huống xử lý dữ liệu thường gặp” với Pandas để giúp bạn tra cứu khi cần.

Với mục tiêu của tài liệu là giúp các bạn tiếp cận, ứng dụng Python để phân tích dữ liệu trong một thời gian rất ngắn – 10 ngày.

Bây giờ chúng ta có thể bắt đầu.

Bài 1: Tóm tắt về thống kê (Statistics)

Ôn tập khái niệm

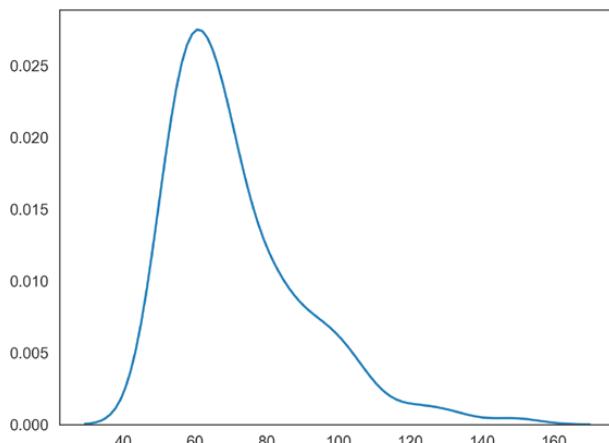
Thống kê (statistics) là công việc tổng hợp gồm nhiều việc nhỏ hơn như: **đặt câu hỏi, thu thập dữ liệu, trình bày dữ liệu, phân tích dữ liệu, diễn giải dữ liệu và suy diễn** (ra thông tin mới).

Xác suất (**probability**) là một cách đo khả năng xảy ra của một biến cố và được ước lượng bằng một con số từ 0 đến 1 (tương ứng từ 0% đến 100%).

Phân phối xác suất (**probability distribution**) là cách mô tả tất cả các khả năng xảy ra của biến cố.

Phân phối xác suất rời rạc (**discrete probability distribution**) thể hiện tất cả giá trị mà một biến ngẫu nhiên có thể có cùng với xác suất của nó.

Phân phối xác suất liên tục (**continuous probability distribution**): biểu diễn xác suất của mỗi giá trị có thể có của một biến ngẫu nhiên liên tục. Ví dụ hình bên dưới minh họa phân phối thời gian di chuyển từ chỗ làm về nhà. Trong đa số trường hợp thì mất khoảng 60 phút, nhưng thỉnh thoảng nhanh hơn vì không có kẹt xe, và thỉnh thoảng mất nhiều thời gian hơn nếu có kẹt xe.



Đánh giá dữ liệu

Một trong các cách để cảm nhận được dữ liệu là đánh giá chúng. Bạn nên làm quen với các khái niệm để đánh giá hoặc đo đạc (measure) dữ liệu như:

- ① **Đo sự tập trung của dữ liệu** (hoặc sự cô đặc của dữ liệu)
- ② Ngược lại với sự tập trung là sự phân tán của dữ liệu. Vì vậy ta cũng cần biết các khái niệm để đo **sự phân tán của dữ liệu**

Đo sự tập trung dữ liệu (Measure of Central Tendency)

Sự tập trung dữ liệu thường được đo bằng giá trị trung bình (average). Có 3 loại giá trị trung bình thường được sử dụng:

Mean: Giá trị trung bình được tính bằng tổng của các giá trị chia cho số lượng các quan sát.

Median

Median gọi là trung vị. Đây chính là giá trị của phần tử ở chính giữa một dãy giá trị có xếp theo thứ tự. Trong trường hợp dãy có số phần tử là chẵn thì trung vị được tính là trung bình của 2 phần tử ở giữa của dãy có thứ tự.

Mở rộng một chút: thay vì quan tâm đến phần tử chính giữa một tập dữ liệu, tức là phần tử mà tại đó chia đôi tập dữ liệu (có thứ tự) thì nếu bạn quan tâm đến phần tử mà tại đó chia $\frac{1}{4}$, $\frac{3}{4}$ hoặc 1 tỉ lệ bất kỳ mà bạn muốn thì xem khái niệm [Quantile](#).

Mode

Mode là giá trị được lặp lại nhiều nhất.

Ví dụ theo dõi giá trị một cổ phiếu được giao dịch theo lô trong một ngày gồm có các mức giá tại mươi thời điểm như sau: 127, 128, 128, 126, 127, 128, 129, 128, 127, 126.

Giá trị mean được tính bằng:

$$(127 + 128 + 127 + 126 + 128 + 128 + 129 + 128 + 127 + 126) / 10 = 127.4$$

Để tìm giá trị median thì ta cần sắp lại thứ tự của mươi mức giá:

126, 126, 127, 127, **127, 128**, 128, 128, 128, 129

Nếu số phần tử là lẻ thì sau khi sắp thứ tự thì median sẽ là giá trị của phần tử chính giữa dãy. Tuy nhiên trong ví dụ này có 10 phần tử, nên median được tính bằng trung bình của 2 phần tử thứ 5 và 6 trong dãy đã xếp thứ tự: $(127 + 128) / 2 = 127.5$

Mode là giá trị 128 (được lặp lại 4 lần)

Đo sự phân tán (Dispersion)

Sự phân tán còn được gọi là tính dao động, hoặc mức độ dao động (varibility) của các giá trị.

Phương sai

Phương sai (variance) dùng để đo độ lệch, hoặc là mức độ cách biệt của các giá trị so với giá trị mean. Quay lại mươi mức giá của cổ phiếu ở trên thì câu hỏi đặt ra là các giá trị dao động như thế nào? Cụ thể trong bảng bên dưới chúng ta tính độ lệch bằng cách đo khoảng cách của Giá cổ phiếu và Giá trị trung bình ở dòng 3. Vì giá trị này có thể là số âm nên tính tổng các độ lệch thì sẽ không phản ảnh được tổng các độ lệch của tất cả giá trị so với giá trung bình. Vì thế phải lấy bình phương các độ lệch sau đó chia cho 10. Kết quả phương sai là 0.84.

Các mức giá cổ phiếu	126	126	127	127	127	128	128	128	128	129
Giá trị mean	127.4	127.4	127.4	127.4	127.4	127.4	127.4	127.4	127.4	127.4

Khoảng cách của Giá và Mean	-1.4	-1.4	-0.4	-0.4	-0.4	0.6	0.6	0.6	0.6	1.6
Bình phương của khoảng cách	1.96	1.96	0.16	0.16	0.16	0.36	0.36	0.36	0.36	2.56
0.84										

Công thức tổng quát để tính phuơng sai là:

$$\text{Var}(X) = \frac{1}{N} \sum_{i=0}^n (x_i - \mu)^2$$

Độ lệch chuẩn (standard deviation)

Độ lệch chuẩn (**standard deviation**): được tính bằng căn bậc hai của phuơng sai. Như vậy Độ lệch chuẩn là một giá trị mà nó có ý nghĩa cũng tương tự phuơng sai. Nó phản ánh mức độ chênh lệch của các giá trị quan sát so với giá trị trung bình. Độ lệch chuẩn càng nhỏ thì cho thấy dữ liệu tập trung càng dày đặc xung quanh giá trị trung bình.

Range

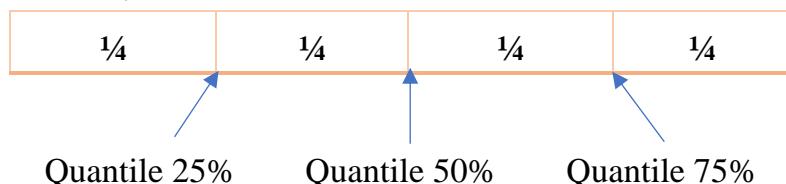
Range là giá trị khác biệt giữa phần tử lớn nhất và phần tử nhỏ nhất.

Quantile

Quantile là dạng tổng quát của Median. Tức là những giá trị (hay điểm cắt (cut points)) chia tập dữ liệu có thứ tự thành p phần có số phần tử bằng nhau. Khi đó ta có thể gọi các điểm này là p-quantiles. Median là 2-quantiles.

Một Quantile phổ biến khác thường được dùng là Tứ phân vị (4-quantiles). Tức là các điểm mà tại đó tập dữ liệu được chia làm 4 phần.

Quantile giúp chúng ta hình dung được sự phân bố, hoặc phân tán dữ liệu. Quantile có thể được xem là mở rộng khái niệm Trung vị (median). Cụ thể tìm Trung vị là tìm giá trị của phần tử để phân tách tập giá trị quan sát thành 2 nửa 50-50; 4-Quantiles sẽ cho biết các giá trị mà tại đó tập giá trị quan sát được tách thành các đoạn $\frac{1}{4}$ như hình bên dưới.



- Bách phân vị 50% chính là median (Trung vị)

Interquartile range

Interquartile range là khoảng giữa hai Bách phân vị 25% và 75%.

Correlation

Mối tương quan (Correlation): Các khái niệm đã thảo luận ở phần trước dùng để đánh giá các biến đơn lẻ (single variable). Để đánh giá mối quan hệ xác suất giữa hai hay nhiều biến thì người ta dùng khái niệm **correlation**.

Bài tập

Nếu tra Internet có thể bạn sẽ thấy thông báo tuyên dụng lập trình viên của các công ty đưa ra nhiều mức lương tháng khác nhau như: \$300, \$400, \$1000, \$1200 và \$700.

Bạn có thể tính các giá trị sau:

$$\text{Giá trị trung bình (Mean)} = \frac{\$300 + \$400 + \$1000 + \$1200 + \$700}{5} = \$720$$

$$\text{Trung vị (Median)} = \$700$$

$$\begin{aligned}\text{Độ lệch chuẩn (SD)} &= \sqrt{\frac{(300-720)^2 + (400-720)^2 + (1000-720)^2 + (1200-720)^2 + (700-720)^2}{5}} \\ &= 343\end{aligned}$$

$$\text{Range: } \$1200 - \$300 = \$900$$

Kiểu dữ liệu (Data Types)

Tùy theo đối tượng và thông tin chúng ta cần đo đạc, quan sát và phân tích thì có nhiều dạng thông tin khác nhau gọi Data Types.

Có thể chia Data Types thành các nhóm như:

① Các thông tin mô tả về đặc tính, đặc trưng của đối tượng như **màu sắc**, giới tính, v.v... gọi là kiểu dữ liệu Danh mục (Categorical data) hay còn gọi là dữ liệu **Định tính** (Qualitative data).

- Các dữ liệu dạng Danh mục không có ý nghĩa về thứ tự (vd: **màu sắc**, giới tính) gọi là **Nominal data**.

- Các dữ liệu về Danh mục nhưng có thêm ý nghĩa thứ tự như: các bậc học (Tiểu học, Phổ thông, Trung học, Đại học, Sau đại học) thì gọi là **Ordinal data**.

② Các thông tin mô tả về đối tượng dưới dạng con số như chiều cao, cân nặng, giá trị cổ phiếu, v.v... thì gọi là Numerical data hay còn gọi là dữ liệu **Định lượng** (Quantitative data).

- Giá trị định lượng có thể là liên tục (Continuous data) như chiều cao, cân nặng.

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

- Giá trị định lượng có thể là rời rạc (Discrete data) như số chân của con vật.

Ví dụ mô tả con mèo hàng xóm có các thông tin sau:

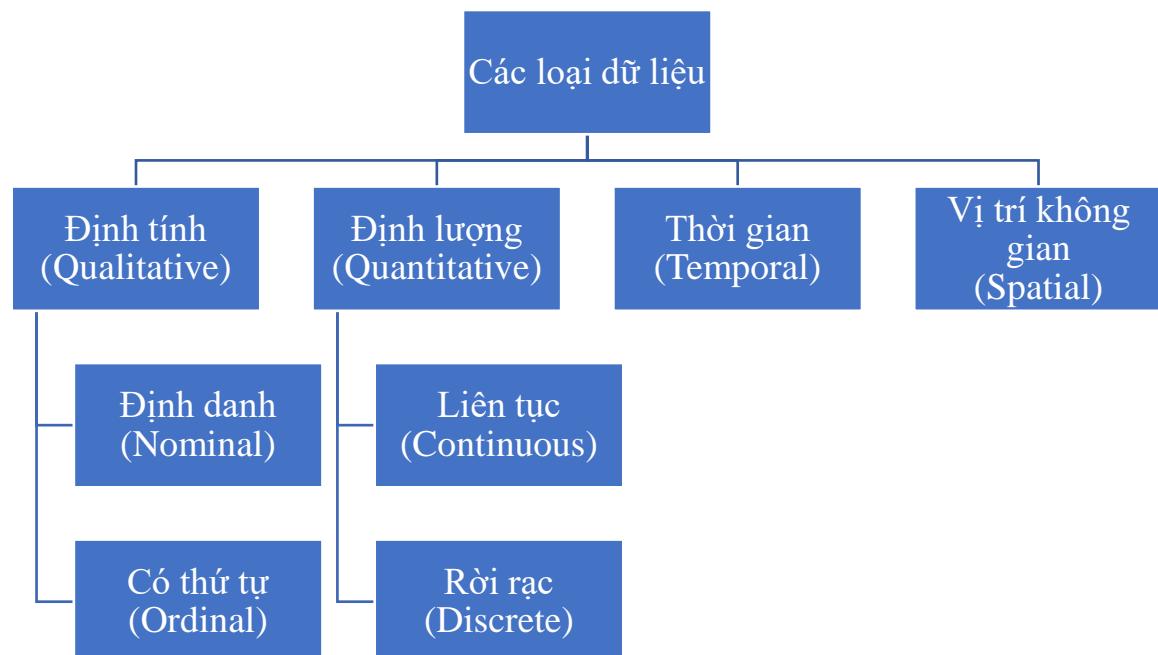
- Màu sắc: đen (Nominal data)
- Giống: cái (Nominal data)
- Nặng: 1.3 kg (Quantitative data - Continuous data)
- Số chân: 4 (Quantitative data – Discrete data)

Ngoài hai dạng dữ liệu Định tính và Định lượng mà bạn thường gặp ở trên thì còn có hai loại khác như:

③ Dạng dữ liệu có đi kèm thêm yếu tố thời gian (Temporal data). Ví dụ giá cổ phiếu. Khi nói đến giá cổ phiếu thì phải nói thêm giá vào ngày nào. Ví dụ giá cổ phiếu của VNM ngày 10/10/2019 là 127 nghìn đồng.

④ Dạng dữ liệu liên quan đến vị trí địa lý (Spatial data). Ví dụ vị trí vật lý trên bản đồ (gồm có kinh độ và vĩ độ), hoặc đơn giản hơn là ví trí gồm x và y trong một hệ trục hai chiều.

Sơ đồ bên dưới tổng hợp các loại dữ liệu:



Hình 2: Sơ đồ các loại dữ liệu

Trên đây là các khái niệm phân loại dữ liệu ở mức trừu tượng.

Để biểu diễn dữ liệu trong máy vi tính và để cho các phần mềm có thể xử lý được dễ dàng thì bạn cần nắm các loại dữ liệu cơ bản sau:

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

- Số nguyên (Integer)
- Số thực (Real / Double number)
- Kí tự (Character)
- Luận lý (Logical)
- Chuỗi (String)
- Thời gian (Date, Time)
- Mảng (Array)

Dùng khái niệm / thước đo nào để quan sát dữ liệu?

Một trong các cách để có cái cảm nhận nhanh về dữ liệu là đo sự tập trung của dữ liệu (central tendency). Như phần trên đã trình bày thì có nhiều thước đo như Mean, Median, Mode.

Cần ôn lại một chút là:

- Mean để thấy giá trị trung bình
- Median để thấy trung vị
- Mode để thấy sự lặp lại của dữ liệu

Câu hỏi đặt ra là với loại dữ liệu nào thì cần dùng thước đo nào?

Bảng bên dưới sẽ gợi ý cho bạn nên dùng thước đo nào cho các kiểu dữ liệu khác nhau.

Kiểu dữ liệu	Đo sự tập trung của dữ liệu	Ghi chú
Nominal	Mode	Nominal là dữ liệu định danh không có thứ tự. Vì vậy cần biết là có bao nhiêu dữ liệu được lặp lại. Ví dụ hôm nay ra đường bạn thấy xe hơi màu nào nhiều nhất. Tức ra đường bạn sẽ thấy rất nhiều xe với nhiều màu khác nhau. Nhưng tựu trung lại ngày hôm nay bạn thấy màu nào nhiều nhất! Nên dùng Mode để tính. <i>Biết đâu màu xe mà bạn gặp nhiều nhất có tác động đến kết quả làm việc của ngày hôm đó?</i>
Ordinal	Median	Ordinal là dữ liệu danh mục có tính thứ tự. Ví dụ trong một doanh nghiệp thì có 5 cấp

Numerical Mean/Median

độ nhân viên lập trình (Dev 1, Dev 2, Dev 3, Dev 4, Dev 5) thì Median của Cấp độ lập trình viên là Dev 3.

Đối với dữ liệu số thì dễ dàng tính giá trị trung bình và trung vị. Ví dụ trong nhóm bạn học của mình thì trung bình chiều cao là bao nhiêu? Nếu đứng xếp hàng theo thứ tự chiều cao thì bạn nào sẽ bạn đứng giữa cao bao nhiêu? (nếu số người là chẵn thì lấy chiều cao trung bình của 2 bạn đứng giữa).

Biến phụ thuộc và biến tiên lượng

Phần lớn các nghiên cứu, mô hình phân tích dữ liệu phân biệt hai loại biến số:

- Biến độc lập (independent variable). Đôi khi gọi là biến tiên lượng (predictor variable), hoặc đặc trưng (feature)
- Biến phụ thuộc (dependent variable). Đôi khi gọi là outcome.

Bài 2: Ngôn ngữ lập trình Python

Để thực hành và trải nghiệm các nội dung trong sách này thì các bạn cần làm quen với Ngôn ngữ thống kê hoặc Ngôn ngữ lập trình trong máy tính và vài công cụ phần mềm. Phần này tôi sẽ giới thiệu cho các bạn ngôn ngữ Python vừa đủ để các bạn trải nghiệm các khái niệm về thống kê, về kiểu dữ liệu đã học trong ngày hôm nay.

Biến (variable) và Đối tượng (Object)

Nếu bạn đã học lập trình thì Variable là một cái tên dùng để chỉ một vùng nhớ trong máy tính. Để đơn giản, bạn hãy tưởng tượng cái máy vi tính giống như não người, trong đó có vùng nhớ (memory) để lưu thông tin tạm thời (lúc máy tính đang bật). Một variable được xem như một cái ô nhớ để chứa một giá trị nào đó.

Hình bên dưới là một thiết bị điện tử có trong máy tính của các bạn. Nó là một bản mạch gồm nhiều con chip có thể lưu trữ lại thông tin (bao gồm cả dữ liệu và lệnh) trong lúc máy tính có điện. Mọi người thường gọi ngắn gọn nó là thanh RAM.



Hình 3: Thanh RAM – nơi lưu "Trí nhớ" tạm thời của máy tính

Để các bạn hiểu hơn một chút về việc khai thác bộ nhớ của máy tính thì hãy tưởng tượng làm cách nào mà bạn bắt cái máy tính của bạn nhớ thông tin của một người bạn thân gồm các thông tin như sau:

Tên	Lê Ngọc Thạch
Chiều cao	165 cm
Cân nặng	72.5 kg
Giới tính	Nam
Ngày sinh	29/9/1977
Các chữ số yêu thích	1, 2, 5, 10, 20, 50, 100
Các môn thể thao yêu thích	Bóng bàn, bóng đá, Quần vợt

(Bạn có thể thay bằng thông tin của chính mình cho chính xác hơn nhé!)

Mỗi thông tin ở cột bên trái được gọi là một **biến** (variable). Bạn tưởng tượng là trong thanh RAM ở phần trước có rất nhiều ô nhỏ li ti. Mỗi ô nhỏ như vậy máy tính (*cụ thể các phần mềm mà chúng ta sẽ thực hành ở phần tiếp theo*) được đặt cho một cái tên (name) – gọi là **tên biến** (variable name). Mỗi biến như vậy sẽ có một vùng nhớ khác nhau để chứa thông tin. Để đơn giản cho máy tính thì chúng ta nên sử dụng tên tiếng Anh để đặt cho tên biến.

Tên biến nên gồm các **kí tự chữ cái thường, chữ cái HOA, dấu gạch chân (_)** và có thể có kí số (ở giữa hoặc ở cuối tên biến). Để thống nhất cho các bạn khi thực hành thì tôi sử dụng quy tắc trước theo thông lệ chung như sau:

- Tên biến bắt đầu bằng chữ thường.
- Kí tự Hoa và thường được hiểu là 2 ký tự khác nhau. *Ví dụ tên biến là fullName sẽ khác với tên biến là FullName. Tức là có hai vùng nhớ khác nhau để chứa thông tin của 2 biến này.*
- Tên biến phải ngắn gọn và gợi nghĩa.
- Khi tên biến gồm nhiều từ ghép lại (như Full name – 2 từ trong ví dụ trên) thì hãy viết Hoa kí tự của từ tiếp theo.

Để mô tả thông tin trong ví dụ trên thì chúng ta có thể tự định tên biến như bảng sau:

Thông tin	Tên biến
Họ và Tên	fullName
Chiều cao	height
Cân nặng	weight
Giới tính	sex
Ngày sinh	birthday
Các chữ số yêu thích	favorNumbers
Các môn thể thao yêu thích	favorSports

Trên đây là thông tin của một người, để mô tả thêm một người bạn nữa thì bạn phải làm sao?

Bạn có thể đặt thêm một loạt biến nữa như: fullName1, height1, ... Tức là bạn thêm số thứ tự phía sau để có bộ biến mới cho người mới. Tuy nhiên cách này không hay. Giới khoa học máy tính đưa ra khái niệm **Object** để giúp các bạn giải quyết nhu cầu này.

Object là một khái niệm gom nhiều loại thông tin để mô tả một vật, một người hay nói chung là một đối tượng nào đó. Nói cụ thể hơn là Object sẽ chứa trong nó nhiều

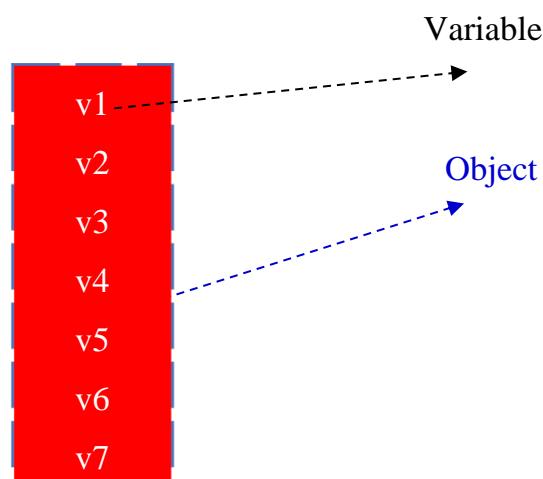
biến. Chúng ta mô tả lại ví dụ trình bày thông tin cho người bạn “Thạch” của chúng ta ở trên dưới dạng một object như sau:

Object: myFriendThach	
fullName	Lê Ngọc Thạch
height	165 cm
weight	72.5 kg
sex	Nam
birthday	29/9/1977
favorNumbers	1, 2, 5, 10, 20, 50
favorSports	Bóng bàn, bóng đá, Quần vợt

Trong bảng trên xuất hiện từ **myFriendThach**, đây là một cái tên (name) được máy tính chỉ định (hoặc là **trỏ tới**) vùng nhớ của tất cả các thông tin về bạn Thạch.

Như vậy đến đây bạn biết được khái niệm biến (**variable**) là một cái tên (name) trỏ tới một vùng nhớ chứa thông tin cơ bản nào đó của bạn Thạch (như tên, cân nặng, v.v....). Toàn bộ các biến liên quan đến bạn Thạch được gom lại trong một vùng nhớ (đương nhiên là rộng hơn) gọi lại **Object**.

Hình minh họa bên dưới gồm 7 ô nhớ tương ứng với 7 biến để mô tả thông tin về bạn Thạch (kí hiệu v1 đến v7 tương ứng với fullName...favorSports). Hình chữ nhật màu xanh được bao gởi đường đứt nét được gọi là một vùng nhớ cũng được đặt tên là một đối tượng (Object) với tên là myFriendThach.



Hình 4: Minh họa khái niệm biến (Variable) và đối tượng (Object)

Variable có nghĩa là gì?

📖 Tra tự điển

Nếu tra tự điển Oxford thì variable có thể là danh từ, có thể là tính từ.

☞ Tính từ variable: *able to be changed or adapted* (có thể được thay đổi hoặc điều chỉnh)

☞ Danh từ variable: *an element, feature, or factor that is liable to vary or change* (một yếu tố, một nét đặc trưng, hoặc một nhân tố có khả năng biến đổi hoặc thay đổi).

Cũng trong Oxford, variable được định nghĩa trong lĩnh vực Computing (điện toán) như sau: *a data item that may take on more than one value during the runtime of a program* (một phần tử dữ liệu có thể mang một hoặc hơn một giá trị trong suốt thời gian thực thi của chương trình).

Như vậy chữ variable có hai nghĩa mà các nhà khoa học máy tính và dịch giả Việt Nam đã dùng từ “biến” đã phản ánh đầy đủ rõ khái niệm “biến” trong máy tính.

Cụ thể là từ **vary** có hàm ý là có thể biến đổi thành đối tượng khác. Đối tượng khác ở đây có nghĩa là bản chất thông tin thay đổi hẳn. Chữ **change** có hàm ý là thay đổi giá trị của ô nhớ. Tức là bản chất, loại thông tin không thay đổi, mà chỉ thay đổi về nội dung, về giá trị của chúng.

Ví dụ:

Biến **height** đang có giá trị là 72.5 thì có thể được thay đổi thành một giá trị khác (tùy theo ngữ cảnh, thời gian như là đo lại tại một thời điểm khác) như là 71, 70 (chúng ta hiểu đơn vị là kg). Sự thay đổi này gọi là **change**.

Tuy nhiên, vì lý do nào đó trong ứng dụng phần mềm chúng ta muốn lưu trữ thông tin không phải là chiều cao nữa mà muốn lưu giá trị là một chức vụ cao nhất mà người đó đã từng làm. Tức là height sẽ được lưu giá trị là một **tên của chức vụ** (chứ không là một con số phản ánh chiều cao nữa). Lúc này biến height được biến đổi từ mục đích lưu con số phản ánh chiều cao thành một tên phản ánh chức vụ cao nhất. Cái này gọi là **vary** theo nghĩa trong tự điển Oxford.

Sau khi bạn hiểu được khái niệm Variable rồi thì câu hỏi tiếp theo là làm sao thiết lập giá trị cho biến. Cụ thể như thiết lập giá trị cho các ô nhớ từ v1 đến v2 trong hình 4.

Để làm được việc này thì bạn cần học thêm khái niệm gán (assign) trong phần tiếp theo.

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

Lệnh gán (assign)

Hình bên dưới minh họa các variable có tên level, score, name, birthday tương ứng với các ô nhớ (hãy xem như là một cái thùng) chứa bên trong nó các thông tin tương ứng.

Để thiết lập thông tin (hay còn gọi là dữ liệu) vào biến thì sử dụng phép gán (assign). Cả Python và R đều dùng chung dấu bằng (=) để thực hiện phép gán.

Trong R, phép gán có thể sử dụng dấu mũi tên (gồm dấu bé hơn và dấu trừ: <-

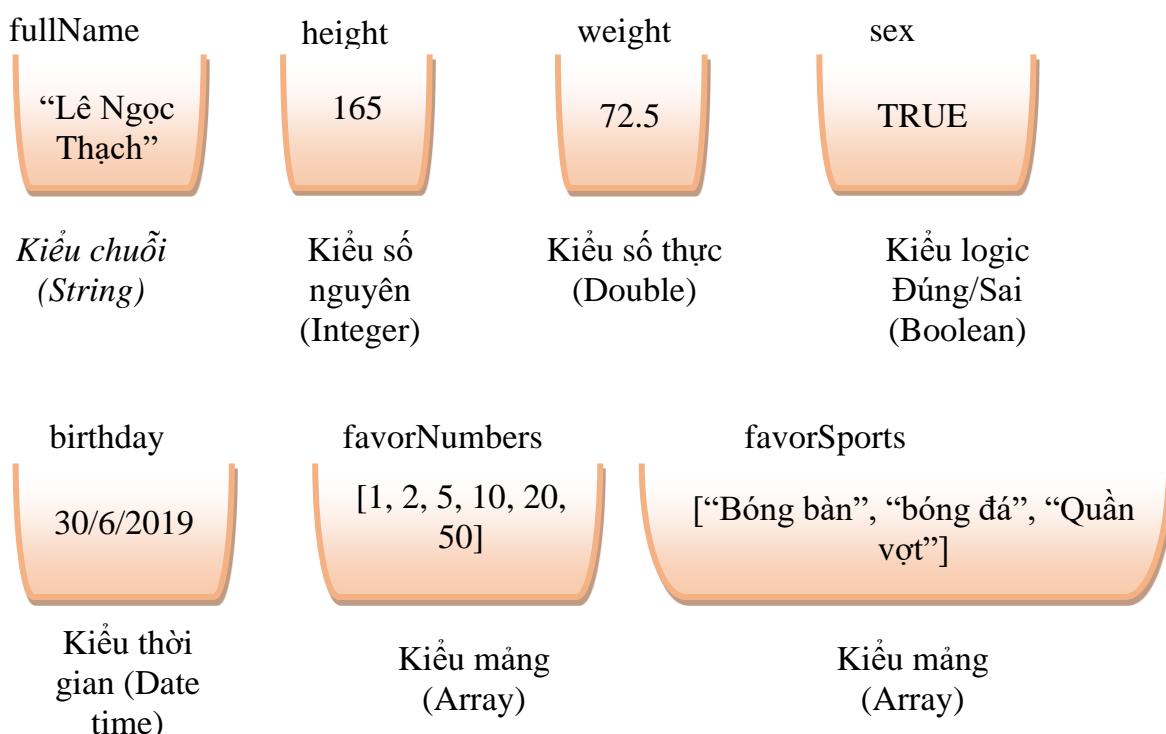
Để thuận tiện thì trong tài liệu này tôi sẽ dùng dấu = làm phép gán.

Để gán một giá trị cho một biến trong R và Python thì dùng dấu bằng “=”.

Vd:

name = "Thạch"

weight = 70



Hình 5: Minh họa biến (variable)

Code Python minh họa:

```
fullName = 'Lê Ngọc Thạch'  
height = 165  
weight = 72.5  
sex = True  
import datetime  
birthday = datetime.datetime.strptime('30/6/2019', '%d/%m/%Y')  
favorNumbers = [1, 2, 5, 10, 20, 50]
```

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

```
favorSports = ['Bóng bàn', 'Bóng đá ', 'Quần vợt']
```

Chú ý:

- Dữ liệu dạng chuỗi thì bắt đầu và kết thúc bởi kí tự dấu nháy đơn hoặc dấu nháy đôi. Thông thường 2 dấu này nằm chung trong một phím phía bên trái phím Enter.

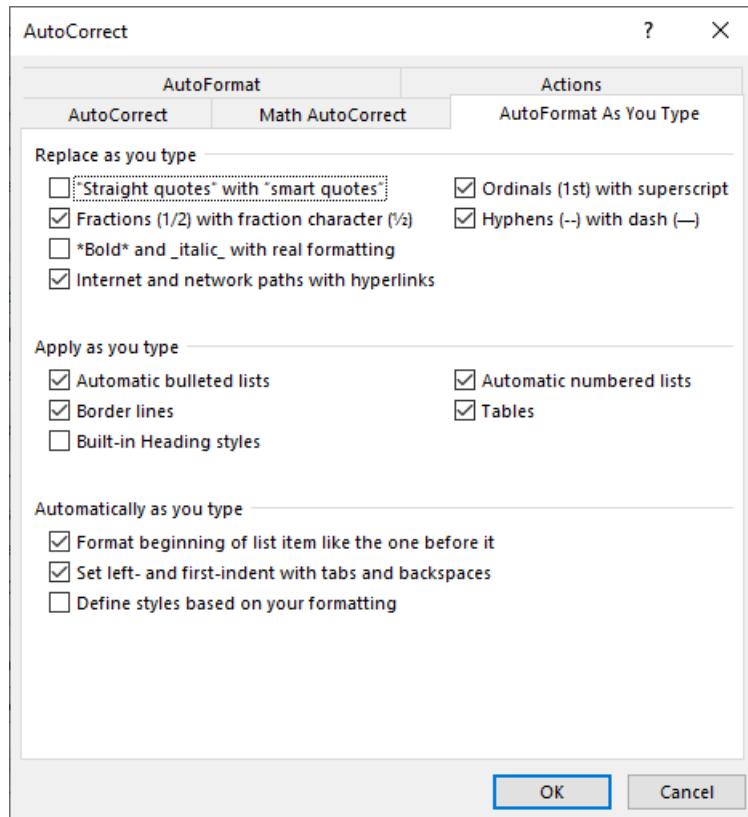


Nhấn phím nháy đơn sẽ nhanh hơn là nhấn phím nháy đôi (phải kèm thêm phím Shift). Vì vậy trong tài liệu này tôi sẽ dùng dấu nháy đơn để bao đóng các dữ liệu dạng chuỗi (string), hay còn gọi là văn bản (text).

Khi chúng ta soạn tài liệu bằng MS Word thì các kí tự nháy đơn và nháy đôi được MS Word thay bằng cách ký tự khác trông đẹp hơn. Điều này sẽ gây ra lỗi nếu chúng ta sao chép mã nguồn từ tài liệu MS Word ra phần mềm thực thi lệnh R hoặc Python hoặc các phần mềm lập trình nói chung.

Để tắt chức năng thay thế thông minh này trong MS Word, bạn tìm vào chỗ cấu hình chức năng Auto Correct rồi bỏ chọn mục "**Straight quotes**" with "**smart quotes**" (tùy theo phiên bản của WS Word thì giao diện có thể khác).

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



- Kiểu số thì cơ bản là giống nhau trong R và Python
- Kiểu luận lý (logical) thì các giá trị ¹ đúng/sai là TRUE/FALSE được viết HOA trong R. Trong Python thì chỉ viết Hoa kí tự đầu tiên như True/False.
- Đối với kiểu dữ liệu ngày tháng thì phức tạp hơn một chút. Việc chúng ta thấy hoặc viết vào máy tính như 30/6/2019 (ngày 30 tháng 6, năm 2019) thì đó là chuỗi các kí tự có ý nghĩa đối với chúng ta. Thật ra máy tính không hiểu nó là giá trị về thời gian. Nếu bạn nào tìm hiểu sâu một chút thì sẽ biết máy tính quản lý biến thời gian là số (trong R là kiểu double).
 - o Để Python hiểu được giá trị thời gian thì dùng thư viện datetime (`import datetime`) và viết lệnh như sau:

```
datetime.datetime.strptime('30/6/2019', '%d/%m/%Y')
```

Chức năng của lệnh này là báo cho phần mềm Python biết chuỗi văn bản "30/6/2019" cần phải chuyển sang dạng thời gian với quy định trong tham số định dạng '%d/%m/%Y'.

¹ Đối với các bạn học lập trình thì nói chính xác hơn là hằng (constant)

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

Định dạng này gồm 3 thành phần ngăn cách nhau bởi dấu xuyệt
phải /:

%d cho biết thành phần đầu tiên có nghĩa là ngày (day)

%m sau dấu / đầu tiên là tháng (month)

%Y sau dấu / thứ hai là năm (Year). Chú ý là chữ Y viết
hoa nhé.

- Kiểu mảng, còn gọi là dãy trong Python.

Trong Python sử dụng cú pháp [] để liệt kê các phần tử của mảng cách
nhau bởi dấu phẩy. Vd:

```
favorNumbers = [1, 2, 5, 10, 20, 50]
```

Bài 3: Ngôn ngữ Python và phần mềm Anaconda

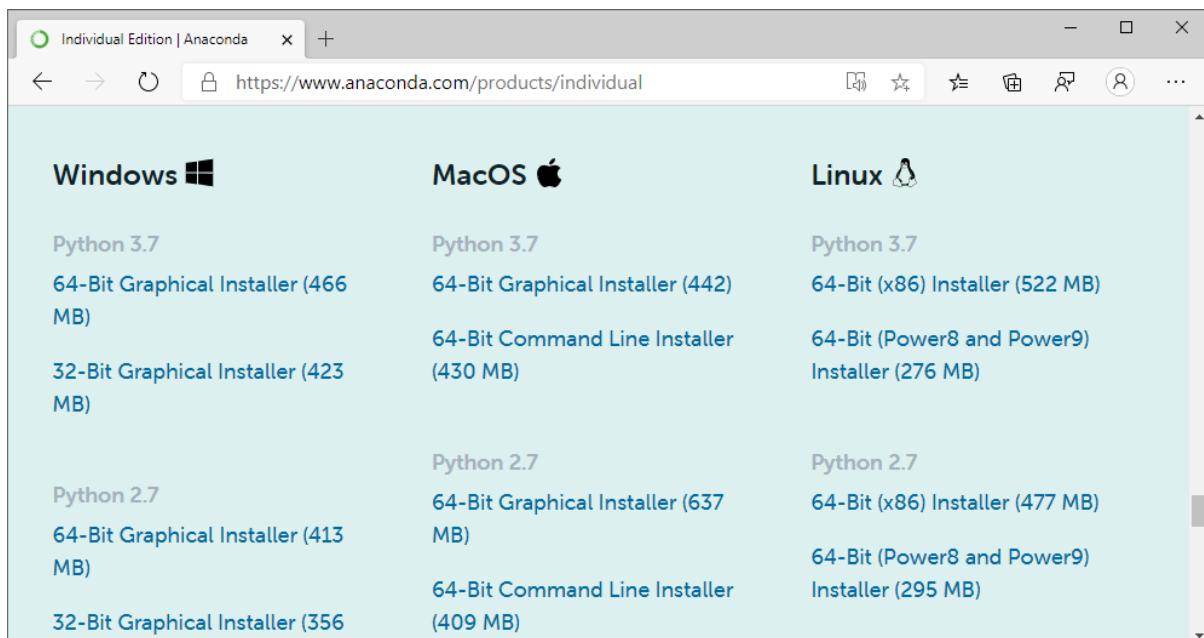
Anaconda

Đối với người mới bắt đầu làm quen với phân tích dữ liệu thì nên cài đặt phần mềm Anaconda tại địa chỉ “<https://anaconda.com>”. Anaconda là bộ quản lý các gói phần mềm (package manager). Trong đó tập trung chủ yếu các gói phần mềm về R và Python. Anaconda miễn phí, dễ sử dụng, có thể chạy được trên các hệ điều hành phổ biến như Windows, Mac, Linux.

Anaconda phù hợp với mọi người để học, thực hiện phân tích dữ liệu, Máy học (Machine learning) bằng ngôn ngữ R và Python.

Cài đặt Anaconda

Vào trang “<https://www.anaconda.com/products/individual>”, bấm vào nút Download:

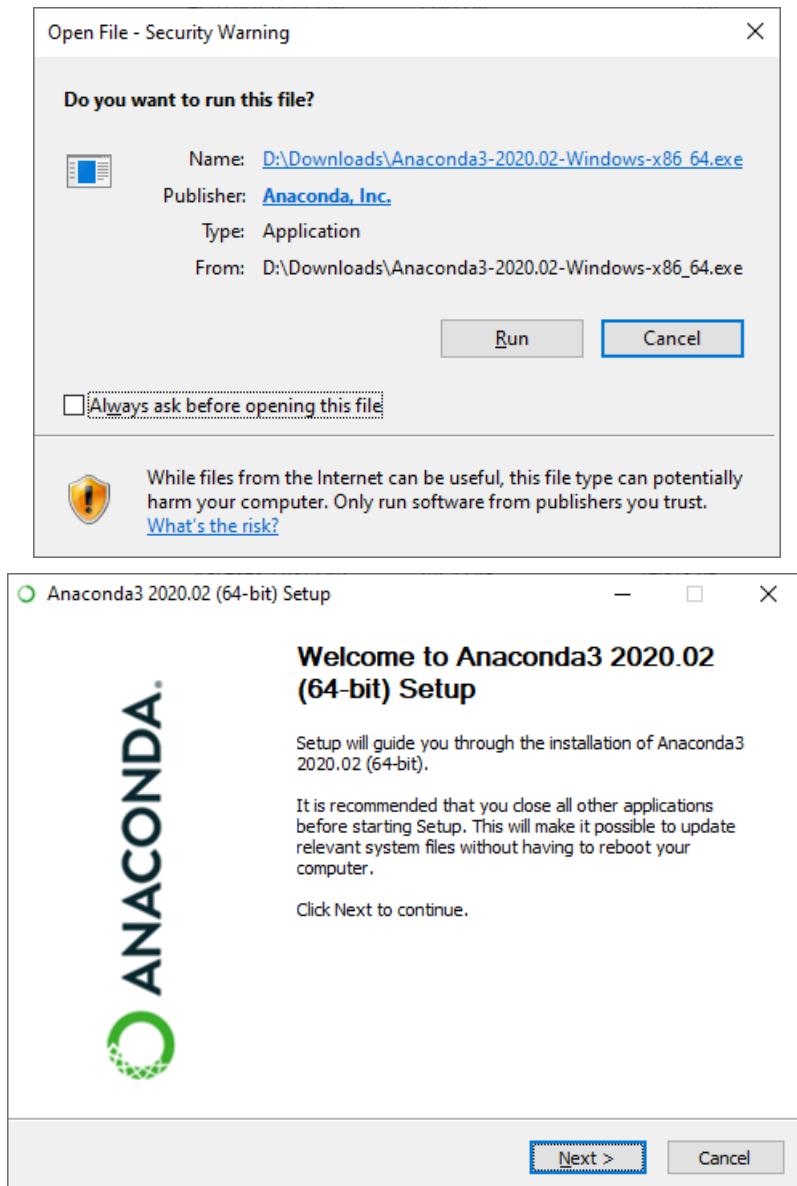


Sau đó bấm mục của gói cài đặt tùy theo máy của bạn. Tại thời điểm viết phần này thì Anconda cung cấp 2 phiên bản phổ biến cho Python 3.7 và Python 2.7. Có nhiều sự khác biệt lớn giữa hai phiên bản Python 3 (gọi chung là 3.x) và Python 2 (gọi chung là 2.x); cũng có nhiều lý do vì sao mọi người đang dùng cả hai phiên bản. Tuy nhiên chi tiết về sự khác biệt này không nằm trong phạm vi của cuốn sách. Và không để bạn mất tập trung thì tạm thời cứ cài đặt phiên bản mới nhất để thực hành. Khi nào gặp vấn đề và cần dùng đến phiên bản cũ (Python 2.7 hoặc 2.x nói chung thì tính sau).

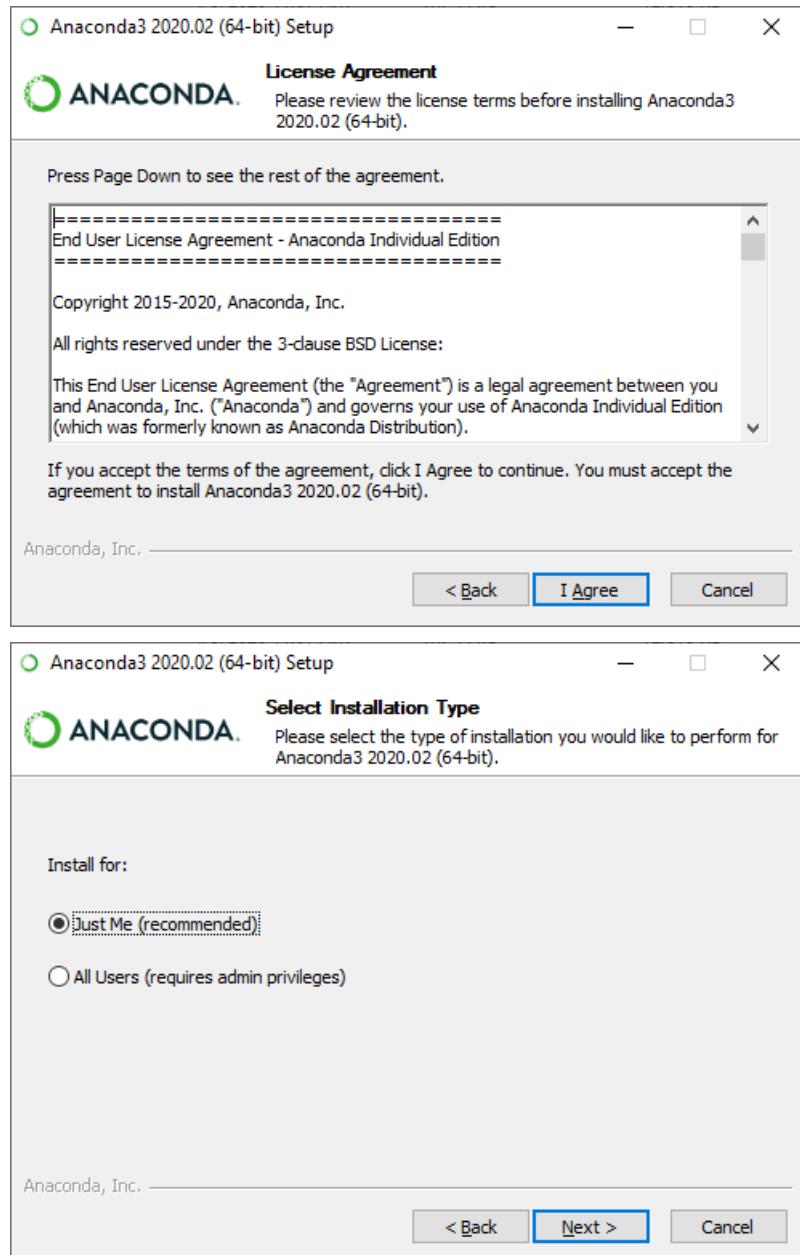
Tôi sẽ dùng phiên bản 3.7 và file tải về là “[64-Bit Graphical Installer \(466 MB\)](#)” do máy tôi dùng Windows 64 bit. Nếu máy bạn đang dùng Windows 32 bit thì tải link “[32-Bit Graphical Installer \(423 MB\)](#)”. Tương tự nếu bạn dùng MacOS hoặc Linux thì bấm vào link tương ứng phía trên màn hình.

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

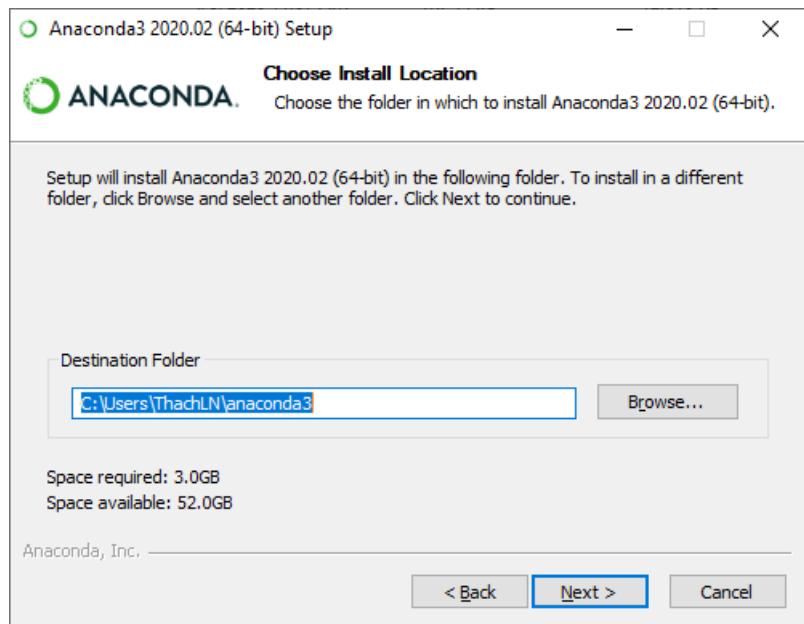
Quá trình cài đặt khá đơn giản. Cơ bản là cứ bấm “Next” và “Agree” rồi làm theo hướng dẫn.



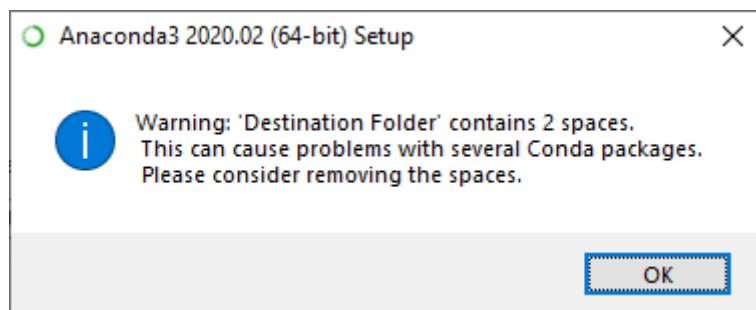
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



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

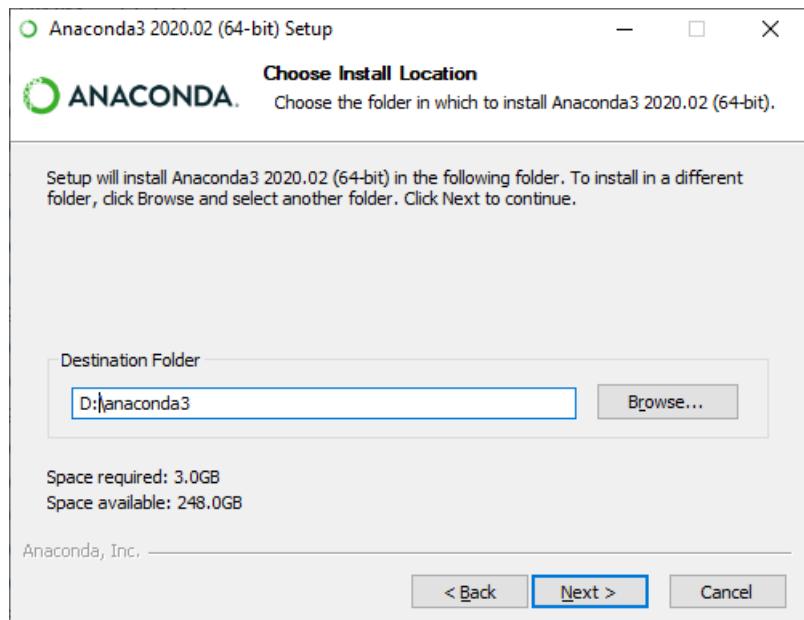


Tới bước chọn thư mục “Destination Folder” thì nếu bạn bấm “Next” mà tên thư mục của bạn có khoảng trắng (ví dụ tôi dùng tên đầy đủ để đăng nhập vào máy nên có khoảng trắng) thì sẽ bị cảnh báo như bên dưới:

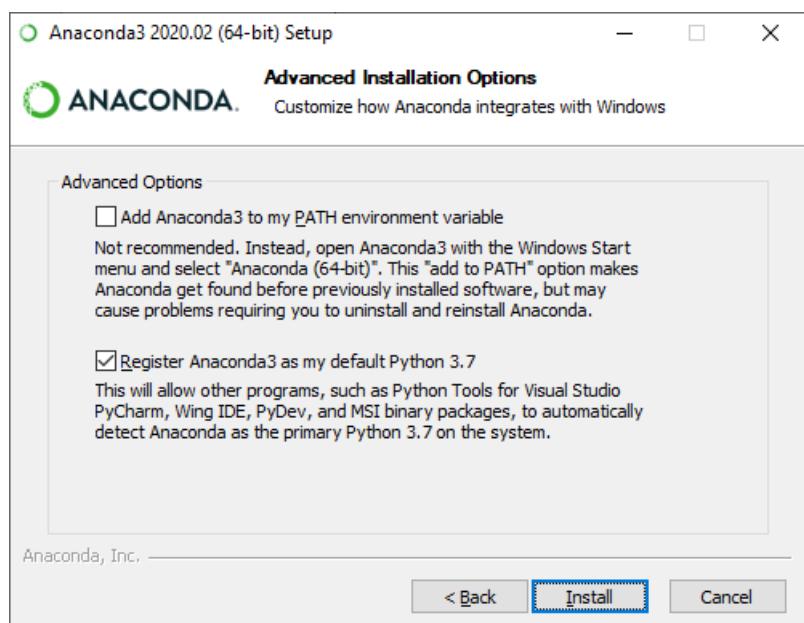


Lúc này bạn nên bấm OK rồi bấm tiếp “Back” trên màn hình tiếp theo để quay lại bước chọn “Destination Folder”. Ví dụ tôi chọn lại thư mục “D:\anaconda3” không có khoảng trắng và dễ quản lý.

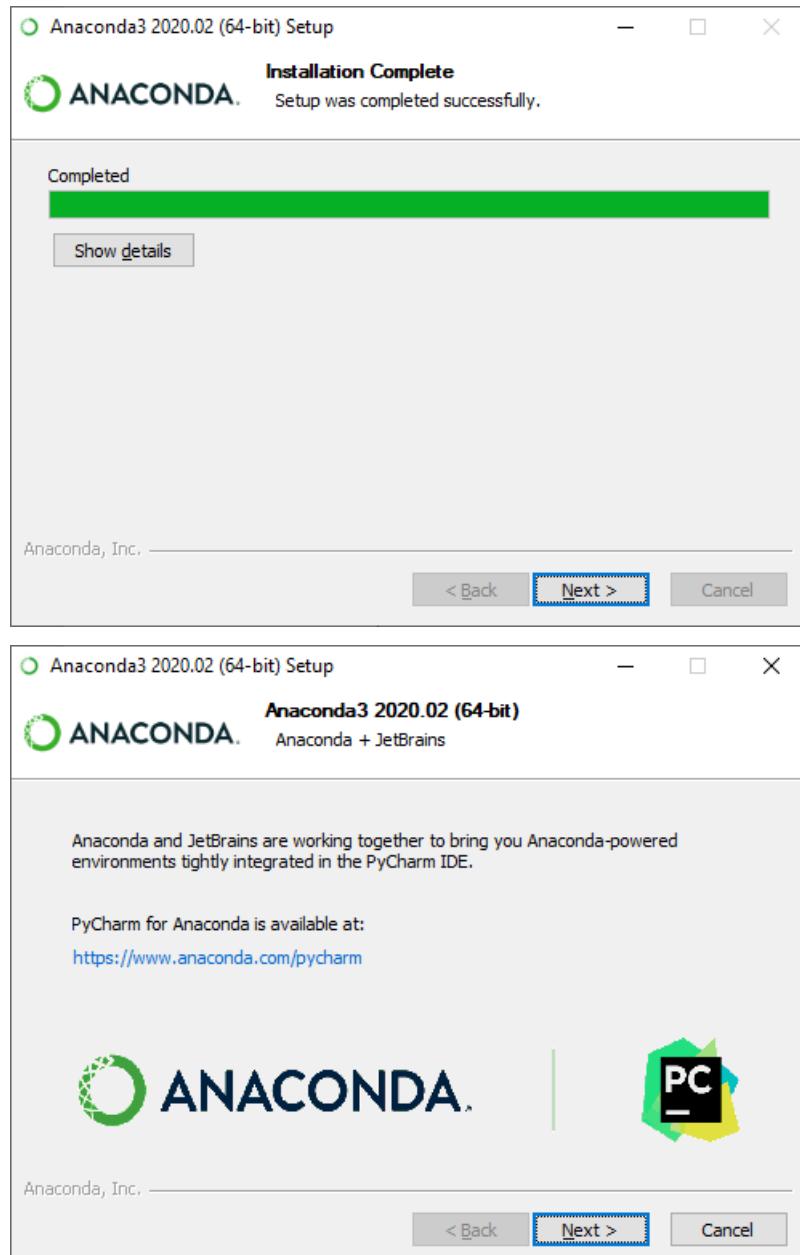
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



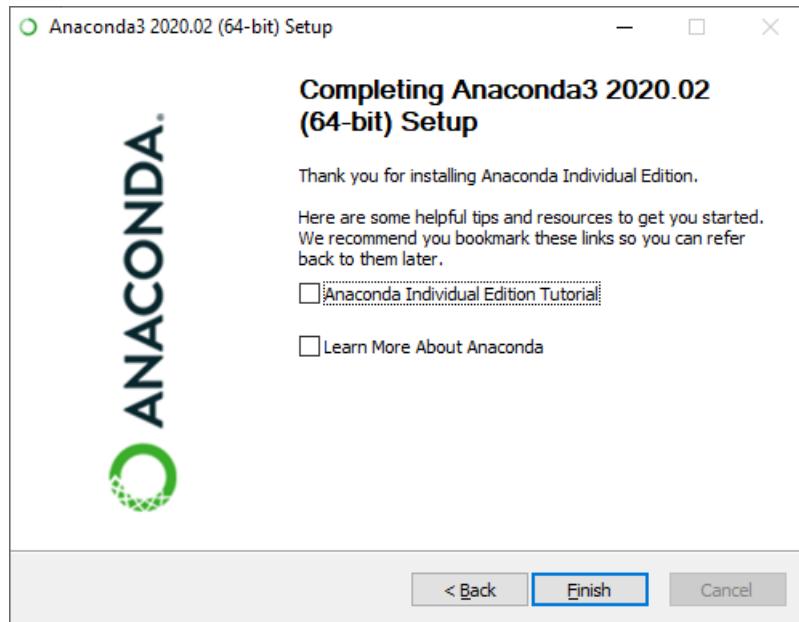
Rồi cứ thế bấm “Next”, “Install” và “Next” cho đến khi “Finish” là xong.



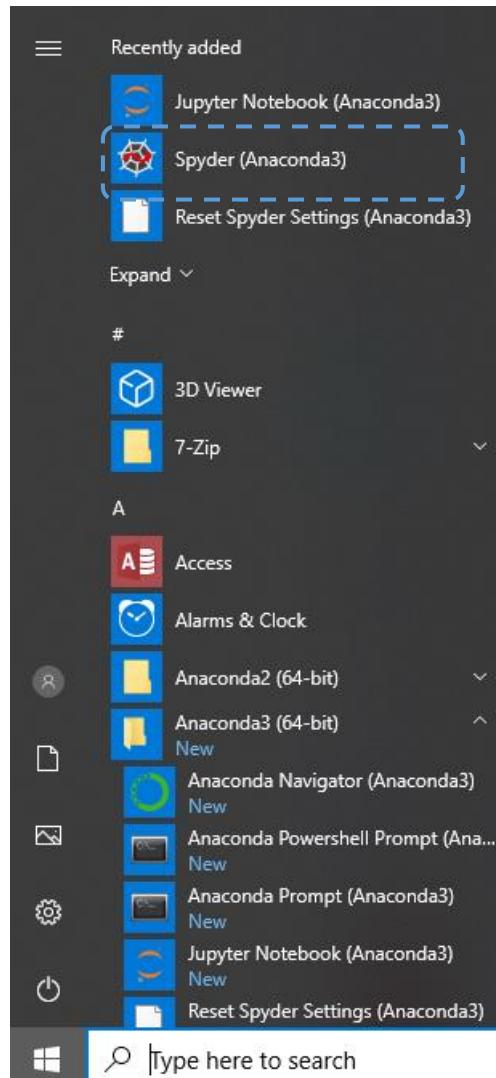
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



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



Sau khi cài xong, vào nút Start của Windows ở góc trái dưới màn hình hoặc bấm phím có hình cửa sổ hoặc (tùy bàn phím) bạn sẽ thấy biểu tượng chương trình Spyder (Anaconda 3) như sau:



Đến đây bạn đã biết cách tải và cài đặt Anaconda Python 3. Bạn cũng nên thử khởi động Spyder (Anaconda3) và thoát nó, tắt máy tính đi uống một ly café hoặc trà sữa tùy theo sở thích để tự thưởng cho mình.

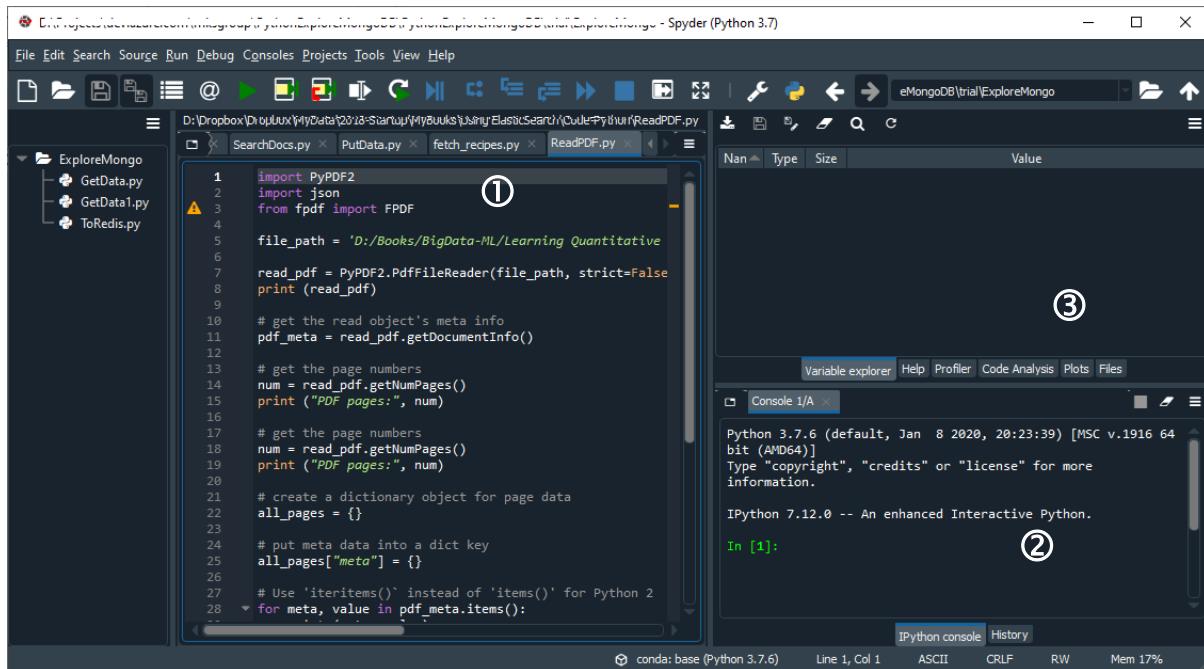
Ngôn ngữ lập trình Python

Bạn có thể bắt đầu làm quen với ngôn ngữ Python và thực hành với các gói phần mềm trong bộ Anaconda đã cài đặt trong phần trước.

Sử dụng Spyder

Sau khi cài đặt Anaconda Python, hãy khởi động chương trình Spyder sẽ có giao diện như sau:

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



Spyder là phần mềm để viết mã lệnh Python được thiết kế bởi các nhà khoa học (scientists), các kỹ sư công nghệ (engineers) và các nhà phân tích dữ liệu (data analysts).

① Phần cửa sổ bên trái giúp bạn viết lệnh Python. Các lệnh này sẽ được lưu vào một file tạm trên máy tính của bạn (ví dụ thư mục trên máy tôi là “C\Users\Le Ngoc Thach”). Tên file untitled0.py có nghĩa là file chưa được đặt tên (untitled) đầu tiên (có thứ tự bắt đầu là 0), phần mở rộng sau dấu chấm là “py” - viết tắt của chữ Python.

② Phần cửa sổ “Console” ở góc phải dưới là nơi trình bày kết quả của lệnh khi các lệnh được thực thi (execute).

③ Phần cửa sổ ở góc phải trên có nhiều tab, trong đó 2 tab “Variable explorer” và “Plots”. Variable explorer giúp bạn theo dõi các biến mà bạn đã khai báo (declare) trong cửa sổ lệnh bên trái khi các lệnh được thực thi. Plots giúp bạn xem kết quả về biểu đồ.

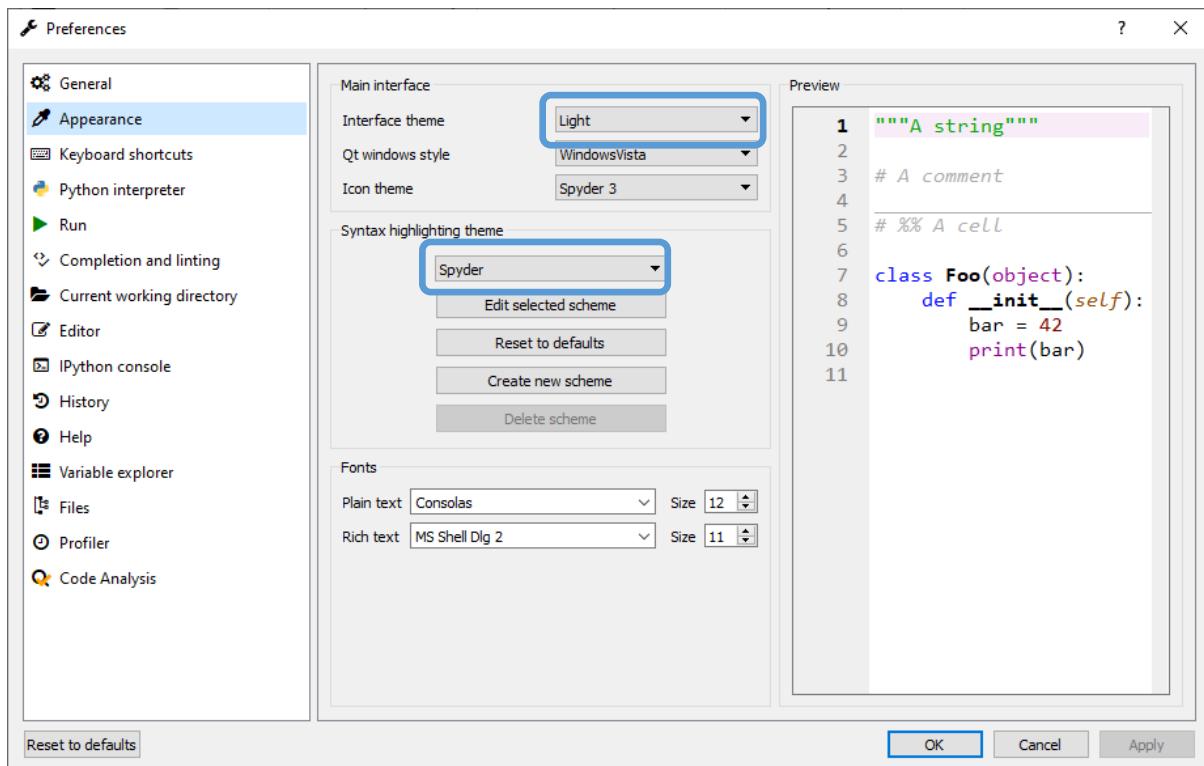
Đổi theme

Mặc định thì Spyder phiên bản 4.x có giao diện đen xì như trên. Nếu bạn không quen thì đổi sang giao diện sáng (light) bằng cách vào menu Tools > Preference, chọn lại:

Interface theme: **Light**

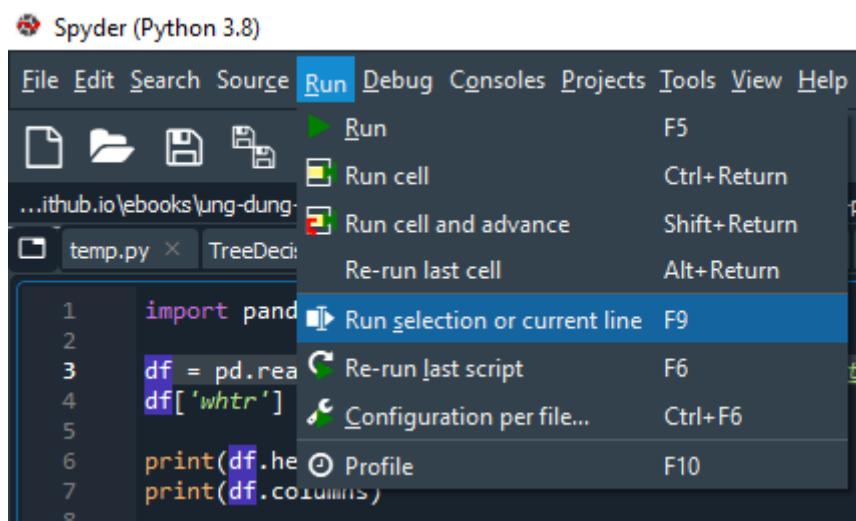
Syntax highlighting theme, mục đầu tiên: **Spyder**

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



Thực thi lệnh

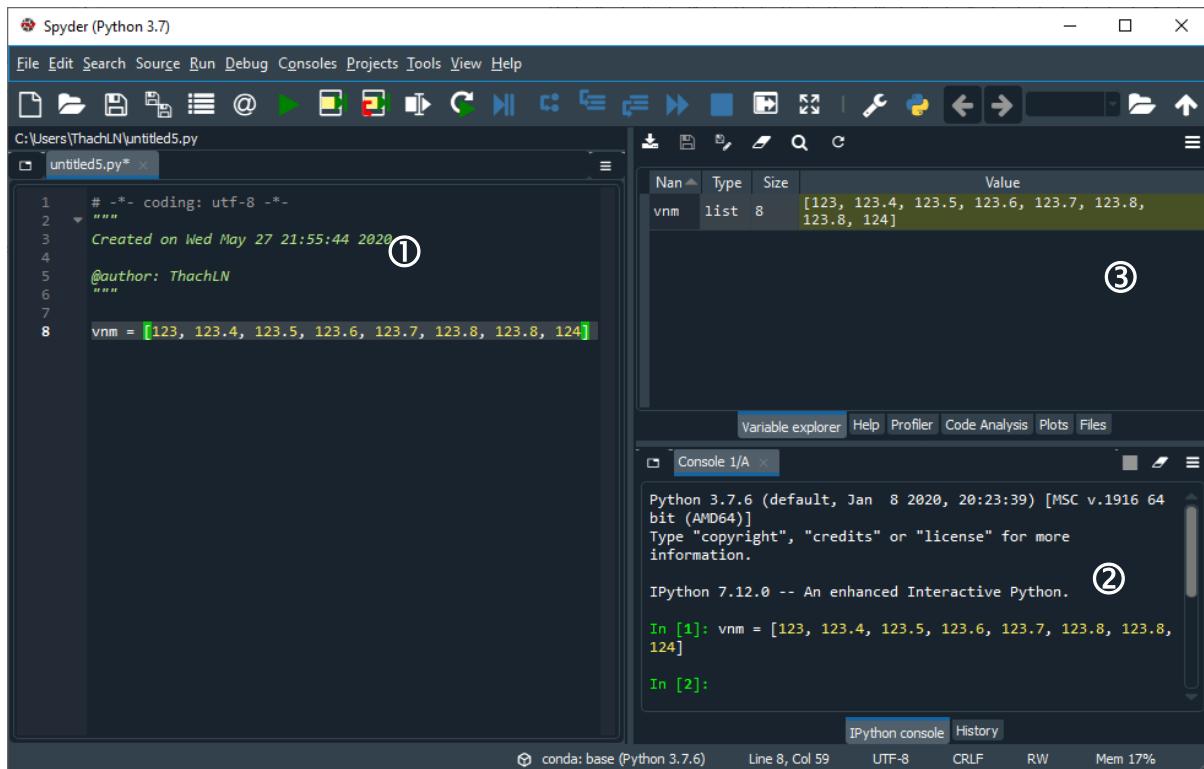
Chọn dòng lệnh cần thực thi, nhấn phím **F9**. Trường hợp không chọn dòng nào cả thì F9 sẽ thực thi dòng hiện tại có con nháy (cursor), sau đó con nháy sẽ nhảy đến dòng tiếp theo. Như vậy bạn có thể dùng F9 tại dòng đầu tiên của chương trình, vừa chạy từng lệnh vừa quan sát kết quả. Nếu bạn quên phím tắt thì có thể vào menu Run:



Ví dụ trong hình bên dưới khai báo một biến có tên **vnm** được gán (assign) bằng một mảng (array) gồm nhiều giá trị cách nhau bởi dấu phẩy. Cặp dấu móc vuông [] bao đóng array theo qui ước của Python.

```
vnm = [123, 123.4, 123.5, 123.6, 123.7, 123.8, 123.8, 124]
```

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



Trong cửa sổ bạn bôi dòng lệnh số 8 bằng các cách sau:

- 1) Dùng chuột bôi từ đầu đến cuối lệnh *bằng cách di chuyển con trỏ chuột đến trước biến vnm, bấm nút trái chuột giữ nguyên nút trái trong lúc di chuyển con chuột sang phải dòng lệnh – hướng di chuyển chuột theo hàng ngang đảm bảo con trỏ chuột lúc nào cũng nằm trên dòng lệnh. Khi con trỏ chuột đến cuối dòng lệnh bạn sẽ thấy dòng lệnh sẽ được bôi màu nền xanh như hình trên.*
- 2) Dùng phím Shift + Home: khi gõ lệnh xong thì con nháy đang ở cuối dòng lệnh. Bạn chỉ cần nhấn tổ hợp phím Shift + Home (tay trái nhấn và giữ nút Shift, sau đó tay phải nhấn phím Home rồi thả cả 2 tay ra khỏi bàn phím cùng lúc).
- 3) Dùng phím Shift + End: khi con nháy đang ở bất kỳ chỗ nào trên dòng lệnh, hãy gõ phím Home để đưa con nháy về vị trí đầu tiên. Sau đó nhấn tổ hợp phím Shift + End (tay trái nhấn và giữ nút Shift, sau đó tay phải nhấn phím End rồi thả cả 2 tay ra khỏi bàn phím cùng lúc).

Thực hành phép gán

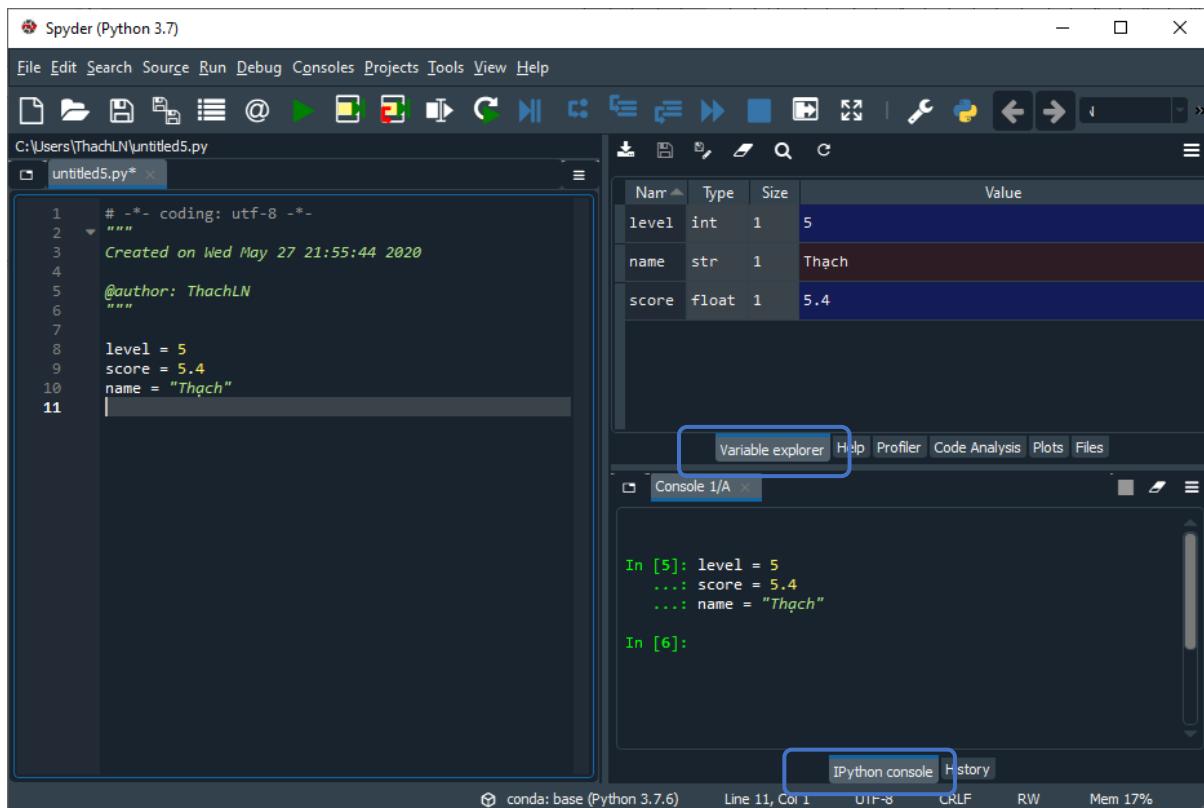
Hãy khởi động chương trình Spyder, mở file mới bằng cách nhấn **Ctrl + N**. Sau đó gõ 3 lệnh sau:

```
level = 5
```

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

```
score = 5.4  
name = "Thạch"
```

Thực thi 3 dòng lệnh bằng cách bôi cả 3 dòng rồi nhấn phím F9. Quan sát giá trị các biến trong thẻ “Variable explorer” và quan sát các lệnh được thực thi trong cửa sổ “Console” ở góc phải dưới.



Sử dụng các gói phần mềm

Python cung cấp rất nhiều gói thư viện. Phần mềm Anaconda đã cài sẵn nhiều gói thư viện cơ bản. Khi nào cần sử dụng cần gói thư viện thì dùng lệnh `import` như sau:

```
import <tên thư viện> as <tên viết tắt>
```

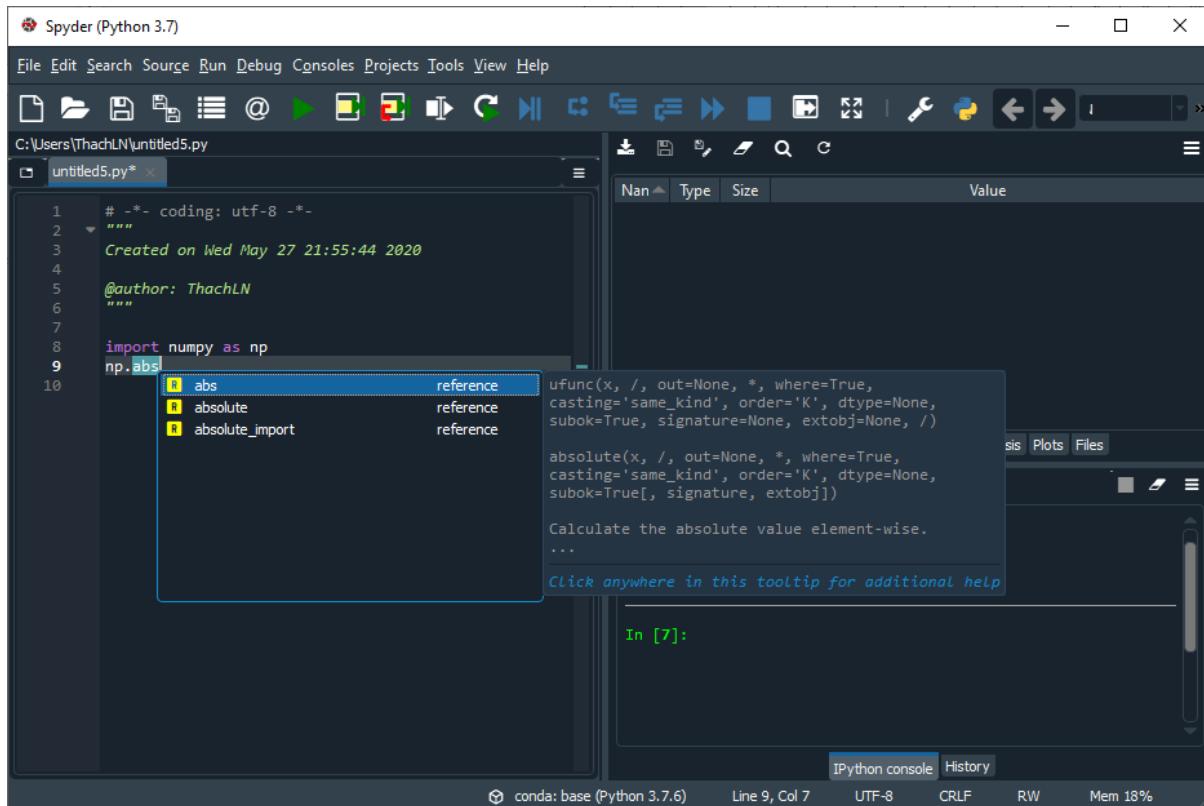
Ví dụ để sử dụng thư viện **numpy** thì sử dụng lệnh

```
import numpy as np
```

Tên viết tắt là do bạn quy định để thuận tiện khi viết lệnh. Dùng tên viết tắt này để cho mã nguồn gọn hơn.

Trong chương trình Spyder khi gõ lệnh **np**, sau đó nhấn Ctrl + Space thì bạn sẽ thấy các hàm của numpy hiển thị ra cho bạn dễ chọn hoặc dễ gõ tiếp.

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



Gọi hàm

Tương tự như minh họa gọi hàm trong R, phần này cũng sẽ giới thiệu lại ví dụ về điểm số để bạn làm quen trong Python.

Trong R thì các hàm để tính toán các khái niệm thống kê cơ bản đã có sẵn, bạn chỉ cần gõ lệnh là thực thi được.

Tuy nhiên, trong Python thì một số hàm được cung cấp trong thư viện **NumPy**. Vì vậy bạn cần phải thực thi lệnh `import` như sau để bắt đầu sử dụng NumPy:

```
import numpy as np
```

Dùng cú pháp `[]` để khai báo danh sách điểm. Sau đó gán danh sách cho biến `scores` như sau:

```
scores = [6, 7, 9, 4, 5, 7, 8, 6, 5, 7]
```

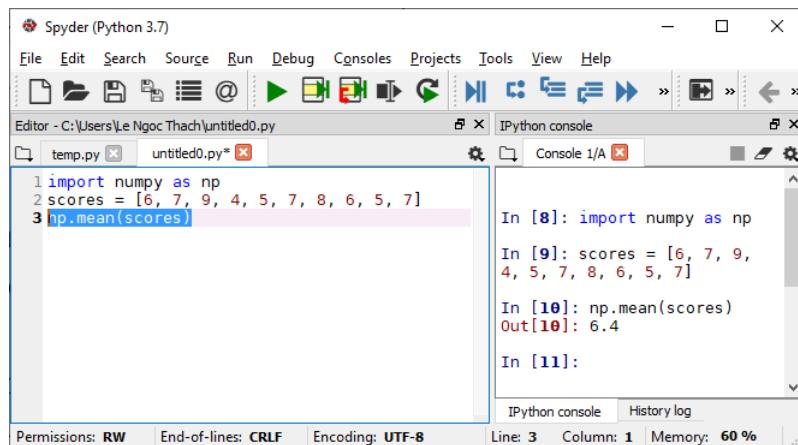
- Gọi hàm `mean` của thư viện `numpy` thông qua kí hiệu `np`:

```
np.mean(scores)
```

sẽ cho kết quả: 6.4

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

Cần nhắc lại một chút là trong lúc soạn thảo lệnh trong Spyder, để thực thi từng dòng lệnh thì bôi chọn từng dòng, nhấn Ctrl + Enter hoặc nhấn F9; trường hợp không bôi đoạn lệnh nào thì F9 sẽ thực thi dòng đang có con nháy. Sau đó theo dõi kết quả trong cửa sổ Console.



- Gọi hàm np.median(x):

```
np.median(scores)
```

sẽ cho kết quả: 6.5

- Gọi hàm np.sort(scores):

```
np.sort(scores)
```

sẽ cho kết quả: array([4, 5, 5, 6, 6, 7, 7, 7, 8, 9])

Phần mềm Spyder in ra kết quả có chữ array() và cặp dấu ngoặc [] để cho chúng ta biết đây là mảng.

- Gọi hàm np.var(x):

```
np.var(scores, ddof = 1)
```

sẽ cho kết quả: 2.2666666666666666

Bạn sẽ thắc mắc là trong Python để tính phương sai thì gọi hàm var của thư viện NumPy phải có tham số "ddof = 1". ddof viết tắt của Delta Degrees of Freedom. Kết quả Python cũng hiển thị số lượng kí số phần thập phân cũng khác với R. Delta Degrees of Freedom là gì thì tạm thời lúc này hãy quên nó đi nhé. Chúng ta đang tập làm quen với việc gọi hàm trong Python. Chúng ta sẽ quay lại khái niệm này sau.

- Gọi hàm np.std(x):

```
np.std(scores, ddof = 1)
```

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

sẽ cho kết quả: 1.505545305418162

- Gọi hàm np.quantile(a, q) để tính Bách phân vị:

```
np.quantile(scores, 0.5)  
np.quantile(scores, 0.25)  
np.quantile(scores, 0.75)
```

sẽ cho kết quả tương ứng của Bách phân vị 50%, 25%, 75%: 6.5, 5.25, 7.0

Bài tập thực hành

① *Làm quen với Biến và Phép gán.*

Trong bài 2 tôi có giới thiệu khái niệm Biến và Phép gán.

Đây là thời điểm để bạn mở Spyder thực hành các lệnh sau:

Python

```
import datetime
```

```
fullName = 'Lê Ngọc Thạch'
```

```
height = 165
```

```
weight = 72.5
```

```
sex = True
```

```
birthday = datetime.datetime.strptime('30/6/2019', '%d/%m/%Y')
```

```
favorNumbers = [1, 2, 5, 10, 20, 50]
```

```
favorSports = ['Bóng bàn', 'Bóng đá ', 'Quần vợt']
```

```
# Thử xem giá trị của vài biến
```

```
birthday
```

```
favorNumbers
```

```
# Xem phần tử đầu tiên của favorNumbers
```

```
favorNumbers[0]
```

```
# Đếm số phần tử của biến favorNumbers
```

```
len(favorNumbers)
```

```
# Lấy ra phần tử cuối cùng của biến favorNumbers
```

```
favorNumbers[len(favorNumbers) - 1]
```

Bạn nên copy từng lệnh hoặc tốt nhất là tự gõ vào Spyder để chạy và quan sát.

Sau mỗi lệnh bạn nên gõ lệnh type để biết thêm kiểu dữ liệu của biến:

```
type(<tên biến>)
```

Ví dụ:

```
type(fullName)
```

Cho kết quả là: str

str có nghĩa là String (chuỗi)

② Làm quen hàm thống kê

Khảo sát đoạn chương trình sau:

```
import pandas as pd
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
s_a = pd.Series(a)
s_a.describe()
```

Kết quả như sau:

```
count    15.000000
mean     8.000000
std      4.472136
min      1.000000
25%     4.500000
50%     8.000000
75%    11.500000
max     15.000000
dtype: float64
```

Điễn giải:

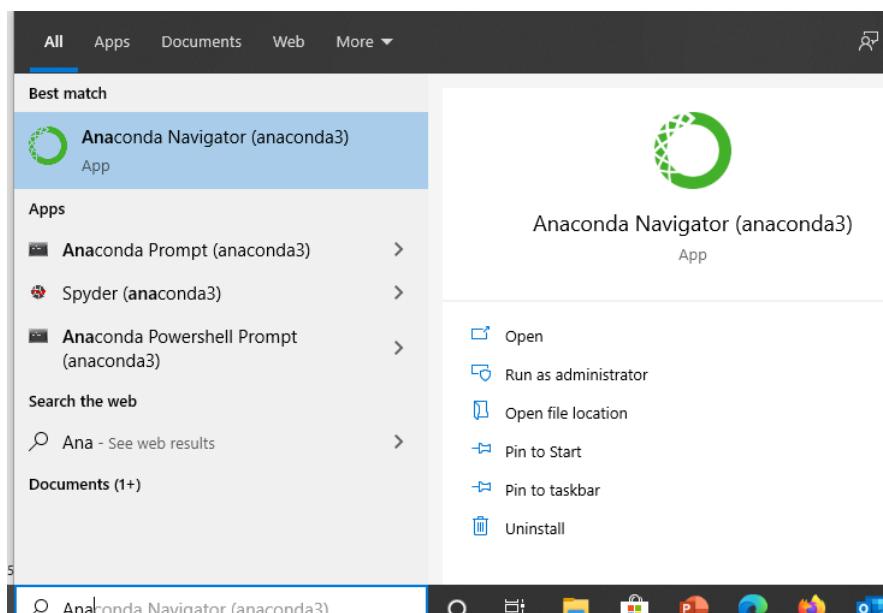
- Lệnh đầu tiên là khai báo sử dụng thư viện pandas với định danh là pd.
- Lệnh thứ hai khai báo một dãy số gồm 15 phần tử, mỗi phần tử có giá trị tương ứng từ 1 đến 15.
- Lệnh thu ba chuyển dãy a thành kiểu dữ liệu gọi là Series – là một cột dữ liệu trong bảng dữ liệu (gọi là Data Frame). Kết quả lưu vào biến s_a.
- Sử dụng hàm .describe() của cột dữ liệu (Series) s_a.

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

- Kết quả hàm `describe()` sẽ cung cấp vài thông tin thống kê để mô tả về biến `s_a`. Cụ thể gồm:
 - count: tổng số phần tử.
 - mean: giá trị trung bình của các phần tử.
 - std: Độ lệch chuẩn (Xem lại mô tả khái niệm [Độ lệch chuẩn](#))
 - min, max: Giá trị nhỏ nhất, Giá trị lớn nhất.
 - 25%: Giá trị bách phân vị 25%. Giá trị mà tại đó chia tập dữ liệu thành 2 phần $\frac{1}{4}$ và $\frac{3}{4}$.
 - 50%: Giá trị bách phân vị 50%. Giá trị mà tại đó chia tập dữ liệu thành 2 phần bằng nhau $\frac{1}{2}$ và $\frac{1}{2}$.
 - 75%: Giá trị bách phân vị 75%. Giá trị mà tại đó chia tập dữ liệu thành 2 phần $\frac{3}{4}$ (Phần các giá trị nhỏ) và $\frac{1}{4}$ (Phần các giá trị lớn..

Cài đặt thư viện

Một trong các lý do mà ngôn ngữ Python phổ biến nhất tại thời điểm eBook được viết trong lĩnh vực Machine Learning và AI là cộng đồng phát triển rất lớn. Trong đó có rất nhiều thư viện được cung cấp miễn phí. Trong Windows, để cài đặt thư viện Python thì mở cửa sổ của Anaconda Prompt hoặc Anaconda Powershell Prompt bằng cách bấm vào nút Windows Start, gõ chữ Ana thì ra màn hình bên dưới, sau đó bấm vào biểu tượng tương ứng (ví dụ Anaconda Prompt).



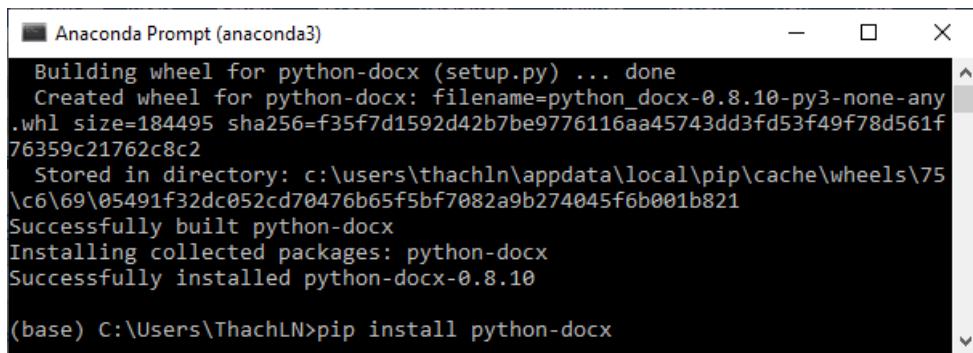
Trong cửa sổ Anaconda Prompt gõ lệnh:

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

```
pip install <tên thư viện>
```

Ví dụ cài thư viện python-docx để xử lý file .docx của Microsoft Word:

```
pip install python-docx
```



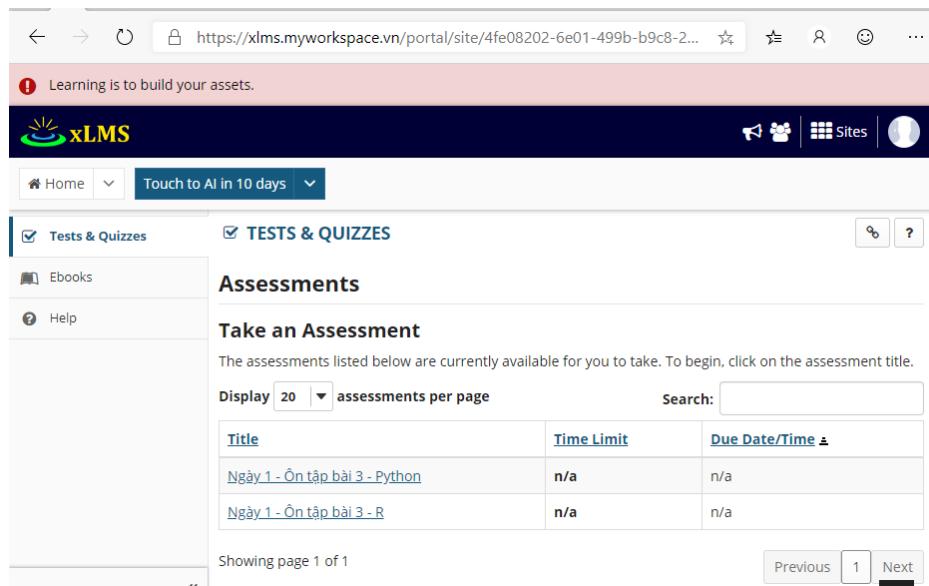
The screenshot shows the Anaconda Prompt window with the title "Anaconda Prompt (anaconda3)". The command "pip install python-docx" was run, and the output is displayed:

```
Building wheel for python-docx (setup.py) ... done
Created wheel for python-docx: filename=python_docx-0.8.10-py3-none-any
.whl size=184495 sha256=f35f7d1592d42b7be9776116aa45743dd3fd53f49f78d561f
76359c21762c8c2
Stored in directory: c:\users\thachln\appdata\local\pip\cache\wheels\75
\c6\69\05491f32dc052cd70476b65f5bf7082a9b274045f6b001b821
Successfully built python-docx
Installing collected packages: python-docx
Successfully installed python-docx-0.8.10
(base) C:\Users\ThachLN>pip install python-docx
```

Online Quizzes

Hãy sử dụng tài khoản được cấp truy cập vào website:

<https://xlms.myworkspace.vn/portal/site/touch-ai>



The screenshot shows a web browser window with the URL <https://xlms.myworkspace.vn/portal/site/4fe08202-6e01-499b-b9c8-2...>. At the top, there's a red banner with the text "Learning is to build your assets." Below it is the xlMS logo. The main content area has a sidebar on the left with "Tests & Quizzes" selected, showing "Ebooks" and "Help". The main panel is titled "TESTS & QUIZZES" with a sub-section "Assessments". It displays two assessments: "Ngày 1 - Ôn tập bài 3 - Python" and "Ngày 1 - Ôn tập bài 3 - R". Both assessments have "n/a" listed under "Time Limit" and "Due Date/Time". A search bar and a "Display 20 assessments per page" button are also present.

Bấm vào mục "Ngày 1 – Ôn tập bài 3 - Python" để thực hiện câu hỏi trắc nghiệm ôn bài.

Sử dụng chú thích

Gõ các lệnh sau vào Spyder để chạy thử. Các dòng có dấu # ở phía trước là các dòng chú thích. Spyder sẽ bỏ qua các dòng này khi thực thi lệnh.

```
# Khai báo biến tuổi
age = 40

# Khai báo 2 hằng số cho giới tính: 1 - Nam; 0 - Nữ
MALE = 1
FEMALE = 0

# Gán biến giới tính - Minh họa kiểu dữ liệu Danh mục
sex = MALE
name = 'Lê Ngọc Thạch'
```

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

The screenshot shows the Spyder IDE interface. In the top left, there's a file menu with options like File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help. Below the menu is a toolbar with icons for file operations, run, stop, and other tools. The main area has tabs for 'temp.py' and 'untitled0.py*'. The code editor contains a script with comments and assignments for variables like age, name, and sex. To the right of the editor is the 'Variable explorer' pane, which lists variables with their type, size, and current value. The 'IPython console' pane at the bottom shows the history of commands entered and their results. The status bar at the bottom provides information about permissions, encoding, and memory usage.

Chú ý là tôi cố tình có lúc dùng cặp nháy đôi, có lúc dùng cặp nháy đơn để bao đóng chuỗi để giúp bạn nhớ là dùng cái nào cũng được, ý nghĩa là như nhau trong Python.

Cập nhật phiên bản mới

Kiểm tra phiên bản mới đã phát hành (release) của Spyder tại website [“https://github.com/spyder-ide/spyder/releases”](https://github.com/spyder-ide/spyder/releases).

Trong cửa sổ Anaconda Powershell Prompt thực hiện lệnh:

```
conda install spyder=4.1.3
```

Nếu bạn kiểm tra phiên bản đã phát hành của Spyder lớn hơn 4.1.3 thì sửa lại lệnh trên cho phù hợp.

Sử dụng Python với CUDA

Cài đặt

The screenshot shows the NVIDIA Developer website with the title "CUDA Toolkit 11.0 Download". A green header bar at the top says "Select Target Platform". Below it, a message reads: "Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#)". The configuration options are listed as follows:

- Operating System:** Windows (selected), Linux
- Architecture:** x86_64 (selected)
- Version:** 10 (selected), Server 2019, Server 2016
- Installer Type:** exe (network) (selected), exe (local)

Download Installer for Windows 10 x86_64

The base installer is available for download below.

➤ Base Installer

Download (2.7 GB) 

Installation Instructions:

1. Double click cuda_11.0.2_451.48_win10.exe
2. Follow on-screen prompts

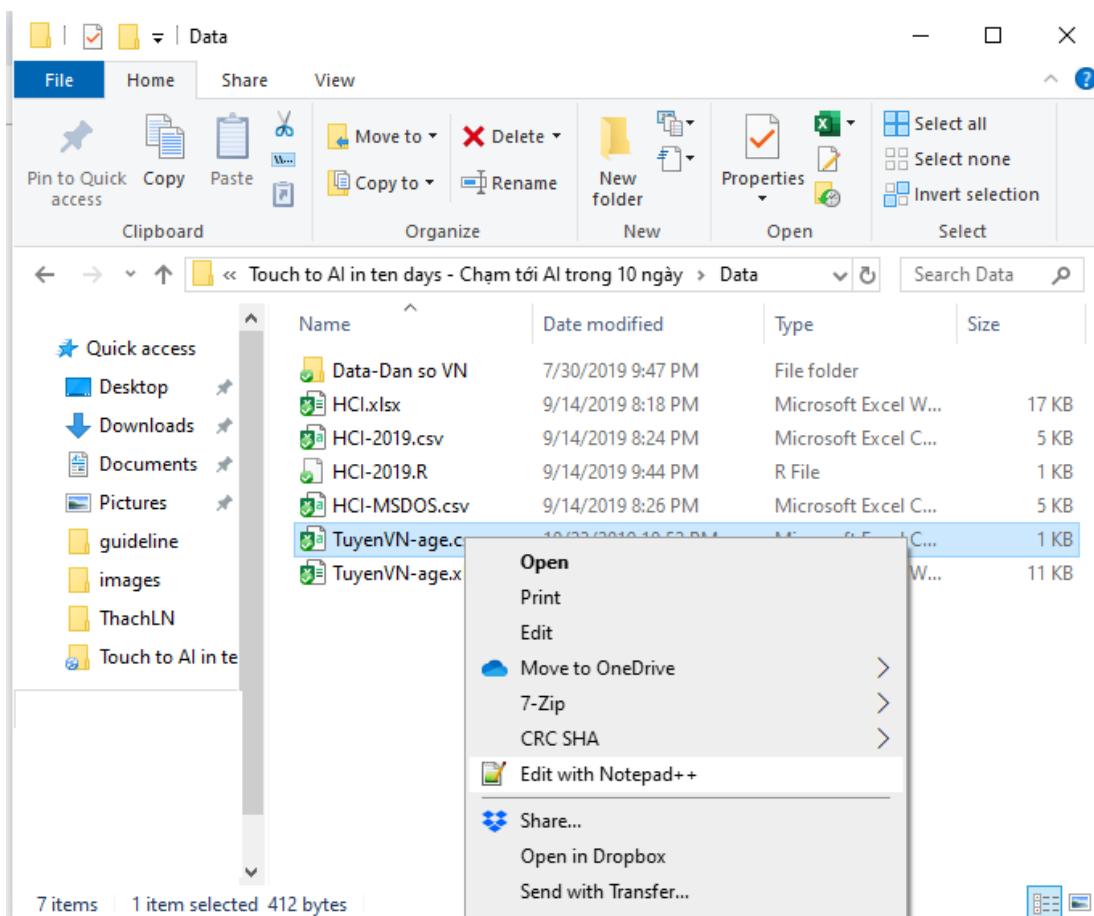
The checksums for the installer and patches can be found in [Installer Checksums](#). For further information, see the [Installation Guide for Microsoft Windows](#) and the [CUDA Quick Start Guide](#).

Bài 4: Cài đặt thêm phần mềm

Phần mềm Notepad++

Trong quá trình học và thực hành thì rất nhiều dữ liệu dùng để làm mẫu rất nhỏ, các lệnh được viết rất đơn giản để bạn nắm được vấn đề. Việc mở nhanh chóng các file dữ liệu và file mã nguồn (cả R và Python) sẽ giúp cho bạn rất nhiều. Tôi khuyên bạn nên cài phần mềm Notepad++ (Đọc là Notepad plus plus). Tải và cài đặt Notepad++ miễn phí tại trang chủ <https://notepad-plus-plus.org/downloads>. Chú ý là nên tải tại trang chủ này chứ không nên tải từ các trang khác để tránh nguy cơ phần mềm không chính chủ - có khả năng bị cài thêm các chức năng gián điệp, virus máy tính.

Sau khi cài đặt xong thì trong Phần mềm quản lý file (File Explorer) của Windows nếu bạn nhấp phải chuột vào file thì sẽ xuất hiện menu “Edit with Notepad++” để giúp bạn mở file nhanh chóng.



Phần mềm Visual Studio Code (VSC)

VSC là phần mềm miễn phí mà Microsoft cung cấp cho cộng đồng. Tải phần mềm tại <https://code.visualstudio.com>.

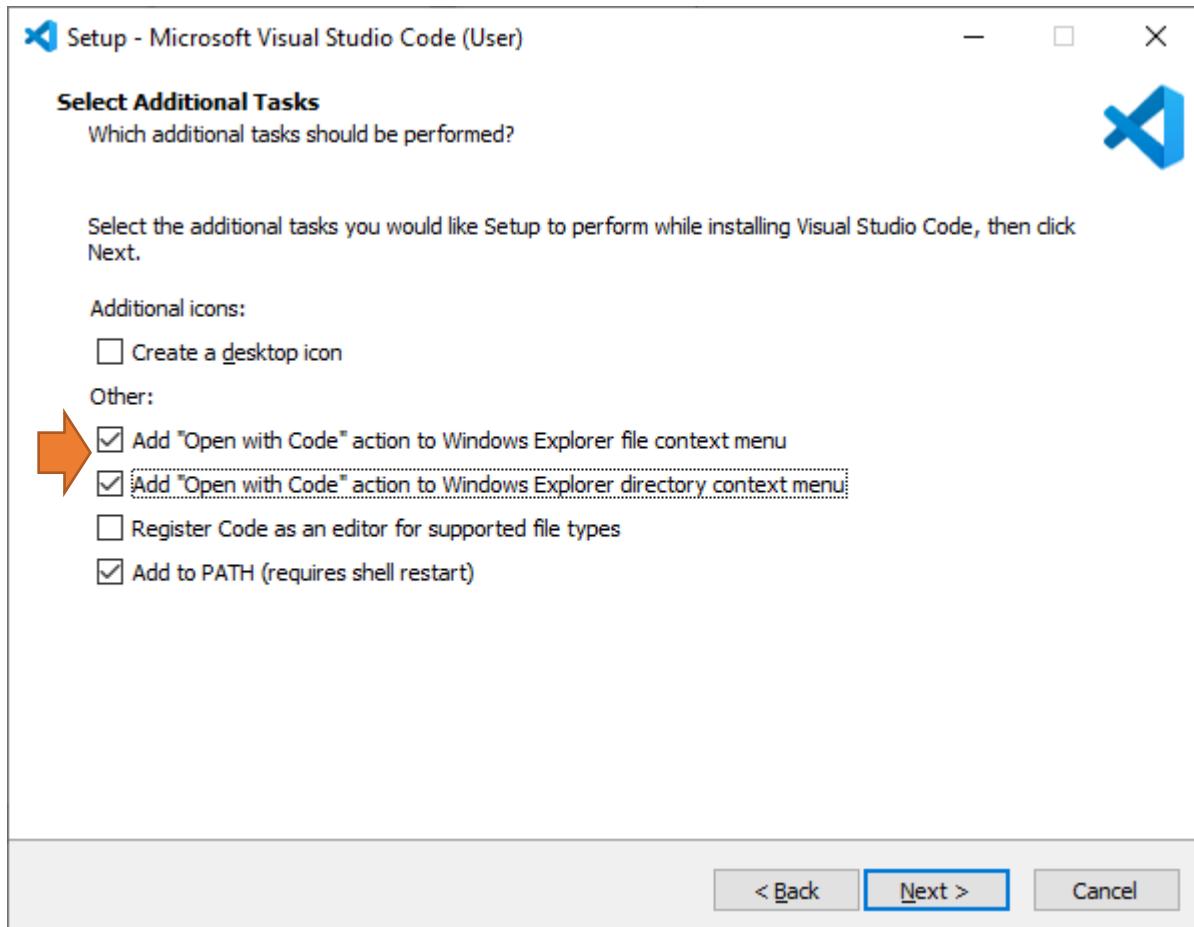
VSC có chức năng mở rộng tích hợp với Python.

Chú ý lúc cài đặt nhớ chọn 2 mục bên dưới:

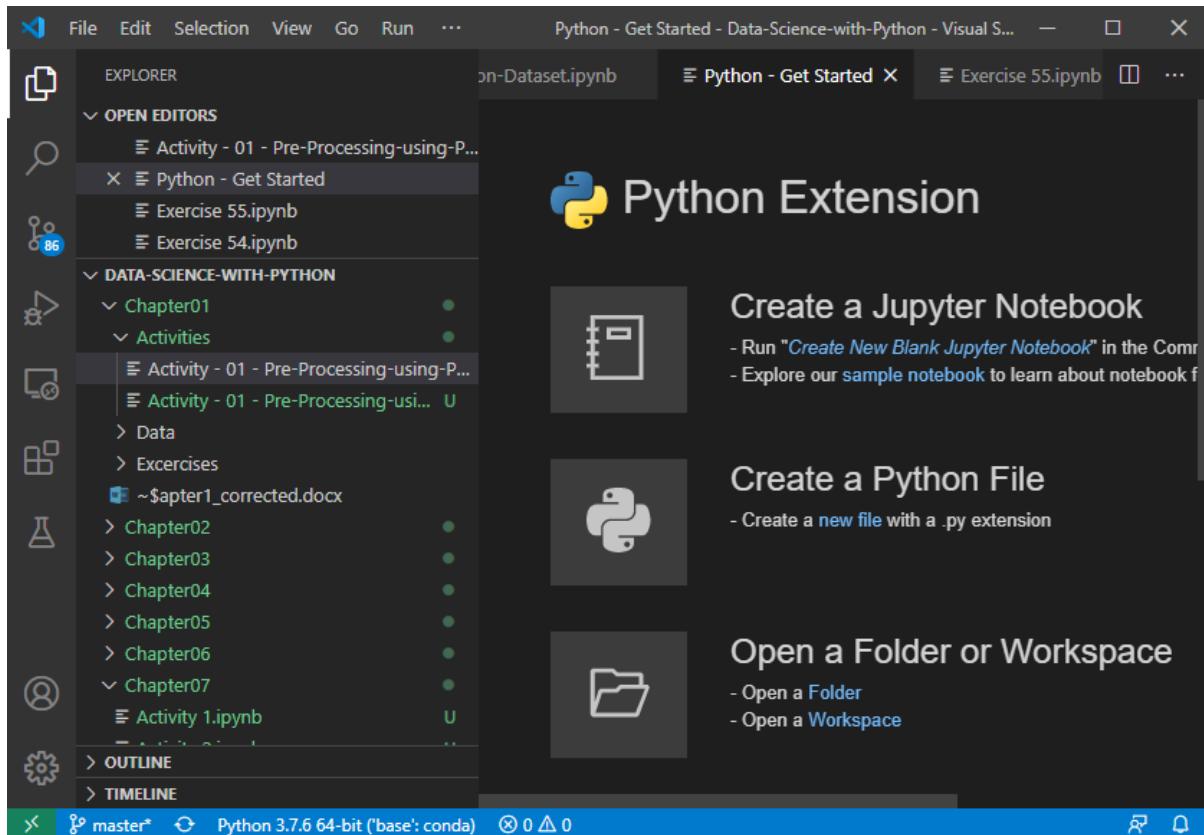
- Add “Open with Code” action to Windows Explorer file context menu

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

- Add “Open with Code” action to Windows Explorer directory context menu



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



Một trong các chức năng mà tôi rất thích VSC là có thể mở thư mục có chứa nhiều thư mục con, nhiều file mã nguồn để xem nhanh. Phím tắt để mở thư mục là Ctrl + K + O.

Phần mềm Visual Studio

Visual Studio là phiên bản thương mại của hãng Microsoft. Tính năng đương nhiên là đầy đủ và chuyên nghiệp hơn là phần mềm Visual Code đã giới thiệu ở phần trước. Dù là bản thương mại nhưng Microsoft cũng cung cấp một phiên bản miễn phí cho cộng đồng (Community). Tải phiên bản này tại:

<https://visualstudio.microsoft.com/vs/community>

Để làm quen với việc lập trình Python với Visual Studio thì bạn nên xem và thực hành theo trang này:

<https://docs.microsoft.com/en-us/visualstudio/python/tutorial-working-with-python-in-visual-studio-step-00-installation>

Phần mềm 7-zip

Đôi khi bạn nhận file nén từ đồng nghiệp hoặc tải file từ trên mạng, hoặc tự nén file để gửi cho người khác thì phần mềm miễn phí 7-zip rất phù hợp cho bạn.

Tải phần mềm tại: <https://www.7-zip.org>

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

Sử dụng PyCharm

PyCharm có thể được xem là một giải pháp phù hợp cho lập trình viên Python developers. IDE (Integrated Development Environment - Môi trường phát triển tích hợp) hỗ trợ nhiều extensions, môi trường ảo (Virtual Environment), nhiều tính năng thông minh giúp cho việc viết mã nguồn rất hiệu quả.

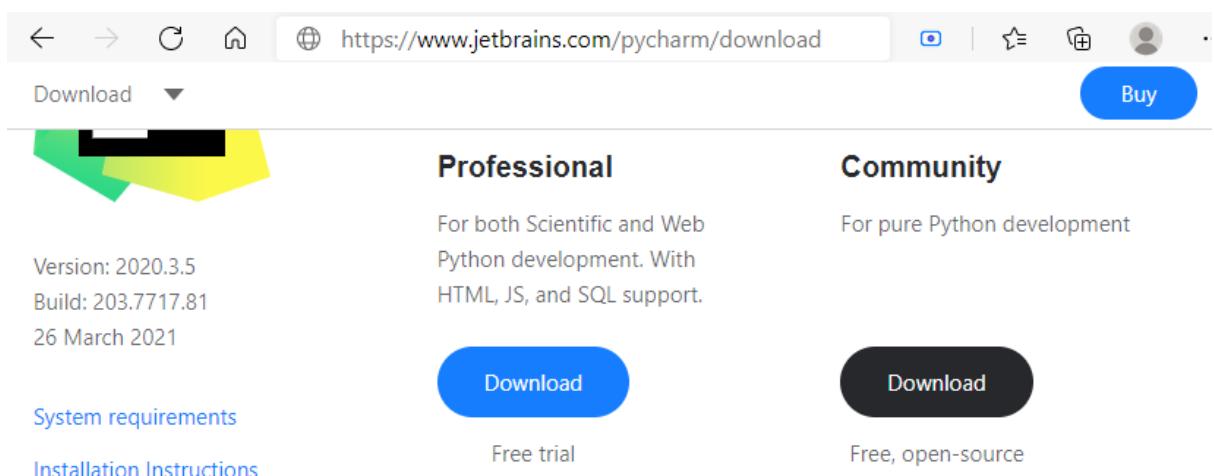
Để giúp cho các bạn không chỉ dừng lại việc học Python, học và ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo mà còn có thể tạo ra sản phẩm hiệu quả thì phần giúp bạn chuẩn bị môi trường lập trình (gọi chung là IDE) PyCharm.

Cài đặt

Vào trang web:

<https://www.jetbrains.com/pycharm/download>

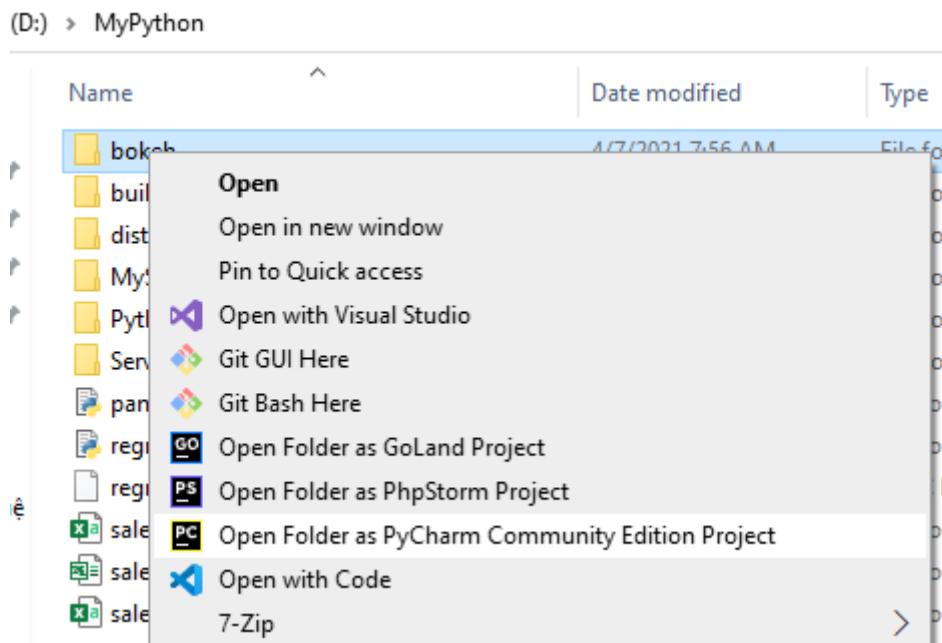
để tải bản Community (Free, open-source)



Sử dụng PyCharm

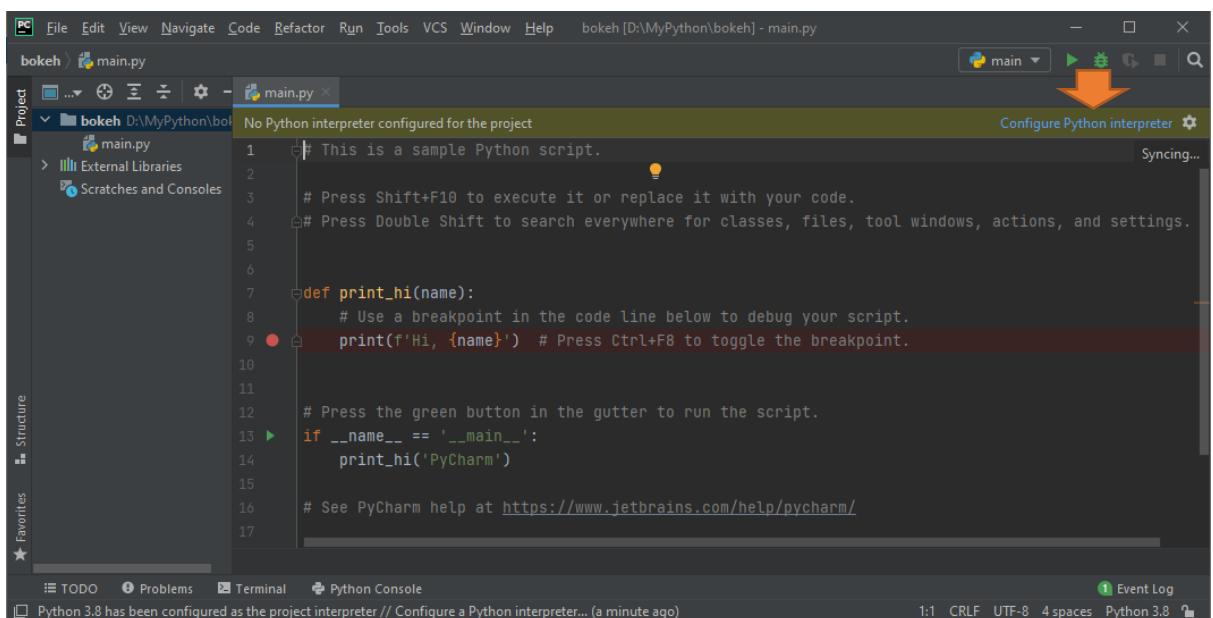
Một trong các cách thuận tiện mà tôi thích dùng là trong cửa sổ Windows Explorer, nhấp nút phải chuột vào thư mục chứa mã nguồn Python (Có thể là một thư mục bài tập gồm nhiều file .py độc lập, cũng có thể là một thư mục của một project), chọn menu “Open Folder as PyCharm Community Edition Project).

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



Nếu là thư mục trống thì PyCharm sẽ đoán rằng bạn muốn tạo ra một dự án Python. Vì vậy nó tạo ra cho bạn file mã nguồn main.py như hình bên dưới.

Nếu PyCharm không tự tìm thấy phần mềm Python thì bạn sẽ thấy mục Configure Python interpreter² (chỗ mũi tên)



² interpreter (thông dịch) phân biệt với compiler (biên dịch). Phần mềm biên dịch có chức năng là dịch toàn bộ mã nguồn thành mã máy tính (gọi là binary package). Sau đó binary package sẽ được chạy trong một hệ điều hành nào đó (nào đó là do thiết lập lúc chạy compiler). Phần mềm thông dịch có chức năng là khi chạy thì nó (interpreter) sẽ đọc file mã nguồn và dịch từng lệnh mã nguồn thành mã máy rồi chạy từng lệnh đó.

Bài 5: Nhập liệu, biên tập, lưu trữ dữ liệu với Python

Dữ liệu trong Python

Dữ liệu trong Python thường được tổ chức dưới dạng mảng đa chiều (multidimensional arrays); hoặc các dữ liệu có cấu trúc với nhiều kiểu dữ liệu khác nhau.

Trong Python, thư viện Pandas cung cấp rất nhiều hàm để thao tác với dữ liệu dạng DataFrame.

Phần này sẽ giúp bạn làm quen với việc xử lý dữ liệu cơ bản với Python thông qua thư viện Pandas.

Giới thiệu thư viện Pandas

Để sử dụng thư viện Pandas bạn dùng lệnh sau:

```
import pandas as pd
```

Để đọc dữ liệu từ file csv vào biến data dùng lệnh sau:

```
data = pd.read_csv('D:/Data/TuyenVN.csv', index_col = 0)
```

Bạn có thể thử thay thế đường dẫn file bằng đường dẫn trên Internet như:

<https://thachln.github.io/datasets/TuyenVN.csv>

Xem dữ liệu của biến data thì gõ tên biến rồi nhấn Enter:

```
data
```

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

The screenshot shows the Spyder Python 3.7 IDE interface. In the top menu bar, the tabs 'File', 'Edit', 'Search', 'Source', 'Run', 'Debug', 'Consoles', 'Projects', 'Tools', 'View', and 'Help' are visible. Below the menu is a toolbar with various icons for file operations like open, save, and run. The main area has tabs for 'temp.py*', 'untitled0.py*', and 'untitled1.py*'. The code in 'untitled1.py*' is:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sun Nov 10 23:28:48 2019
4
5 @author: Le Ngoc Thach
6 """
7
8 import pandas as pd
9 data = pd.read_csv('D:/Data/R/TuyenVN.csv', index_col = 0)
10 data
```

To the right, the IPython console displays the output of the command 'data':

	age	height
1	26	186
2	23	180
3	23	180
4	20	185
5	20	185
6	23	173
7	23	173
8	30	169
9	30	169
10	24	180
11	24	180
12	24	173
13	24	173
14	26	170
15	26	170
16	22	168
17	22	168
18	24	176
19	24	176
20	23	168
21	23	168
22	22	178
23	22	178
24	23	170
25	23	170
26	24	168

In [43]:

Để xem số dòng và cột của dữ liệu "data", dùng lệnh:

```
data.shape
```

Kết quả:

```
(26, 3)
```

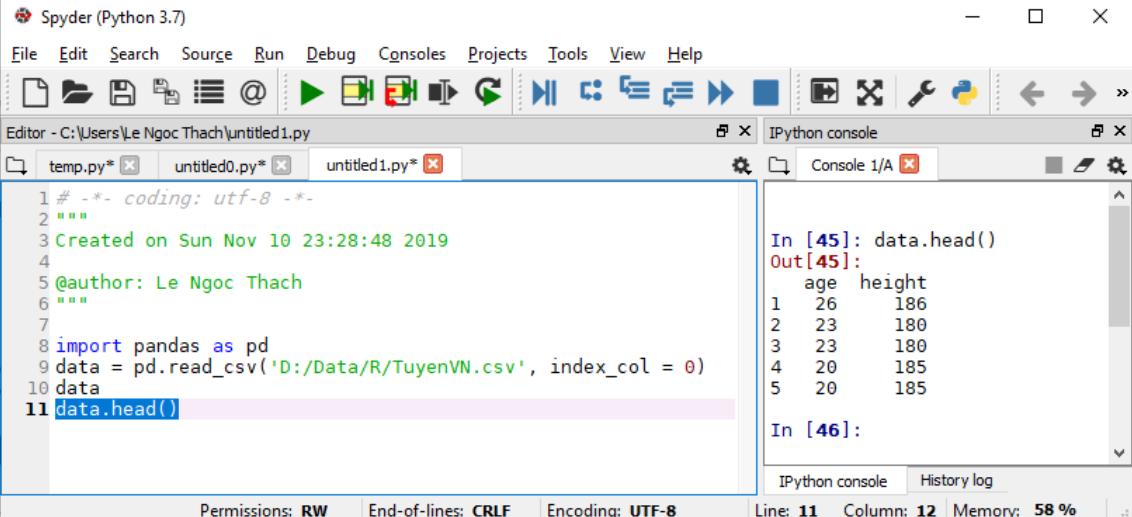
Để xem vài dòng dữ liệu **đầu** và **cuối** của data frame thì dùng hàm head() và tail().

Góc tiếng Anh:

☞ **head**: the part of the body on top of the neck containing the eyes, nose, mouth and brain

☞ **tail**: the part of sticks out and can be moved at the back of the body of a bird, an animal or a fish

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



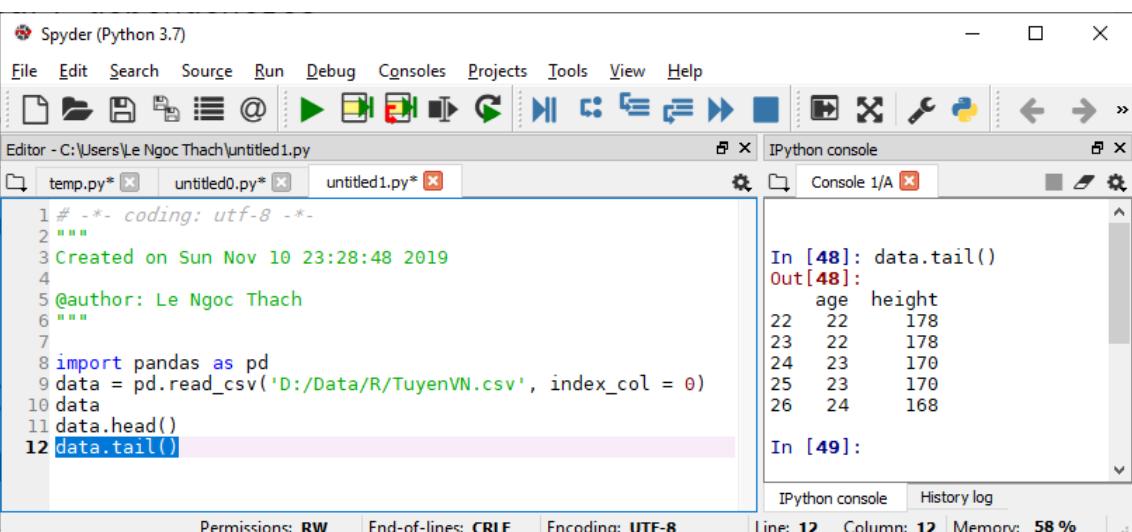
The screenshot shows the Spyder IDE interface with two panes. The left pane is the IPython console showing the following code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sun Nov 10 23:28:48 2019
4
5 @author: Le Ngoc Thach
6 """
7
8 import pandas as pd
9 data = pd.read_csv('D:/Data/R/TuyenVN.csv', index_col = 0)
10 data
11 data.head()
```

The right pane shows the output of the `data.head()` command:

```
In [45]: data.head()
Out[45]:
   age  height
1    26     186
2    23     180
3    23     180
4    20     185
5    20     185
```

Another screenshot below shows the execution of `data.tail()`:



```
In [48]: data.tail()
Out[48]:
   age  height
22   22     178
23   22     178
24   23     170
25   23     170
26   24     168
```

Để lấy ra cột dữ liệu `age` thì dùng cú pháp sau:

```
data["age"]
```

Để tính tuổi trung bình của các tuyển thủ thì dùng hàm `mean` như sau:

```
data["age"].mean()
```

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

The screenshot shows the Spyder IDE interface. In the code editor, a file named 'temp.py' contains the following code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sun Nov 10 23:28:48 2019
4
5 @author: Le Ngoc Thach
6 """
7
8 import pandas as pd
9 data = pd.read_csv('D:/Data/R/TuyenVN.csv', index_col = 0)
10 data
11 data.head()
12 data.tail()
13 data["age"]
14 data["age"].mean()
```

In the IPython console, the command `data["age"].mean()` is run, resulting in the output:

```
In [51]: data["age"].mean()
Out[51]: 23.76923076923077
```

Below the console, the status bar shows: Line: 14 Column: 19 Memory: 58 %.

Kết quả cho thấy tuổi trung bình của các tuyển thủ là xấp xỉ 23.8 tuổi.

Tương tự, có thể tính nhanh chiều cao trung bình bằng lệnh sau:

```
data["height"].mean()
```

Kết quả là: 174.3846153846154

Đơn vị ở đây là cm. Tức là chiều cao trung bình của các tuyển thủ xấp xỉ 1 mét 74.

Nếu gọi hàm `mean` cho data frame thì kết quả như sau:

```
data.mean()
```

```
age      23.769231
height   174.384615
dtype: float64
```

Thư viện Pandas sẽ tự tính trung bình các cột có kiểu số. Trong trường hợp này là `age` và `height`.

Hãy thử chạy các lệnh sau:

Lệnh

Ghi chú

`data.columns`

Xem tên các cột của data frame.

Khai thác thư viện Pandas

Phần trước đã giới thiệu để bạn làm quen với thư viện Pandas trong Python. Nếu bạn chỉ mới làm quen với Phân tích dữ liệu thì có thể bỏ qua phần này. Khi cần thì có thể quay lại sau. Nội dung phần này cố gắng liệt kê các hàm phổ biến trong Pandas để giúp bạn xử lý dữ liệu.

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

Nhãn (label) trong DataFrame là gì?

Trước khi đi vào xem các thao dữ liệu với DataFrame bằng thư viện pandas thì cũng cần biết qua khái niệm label.

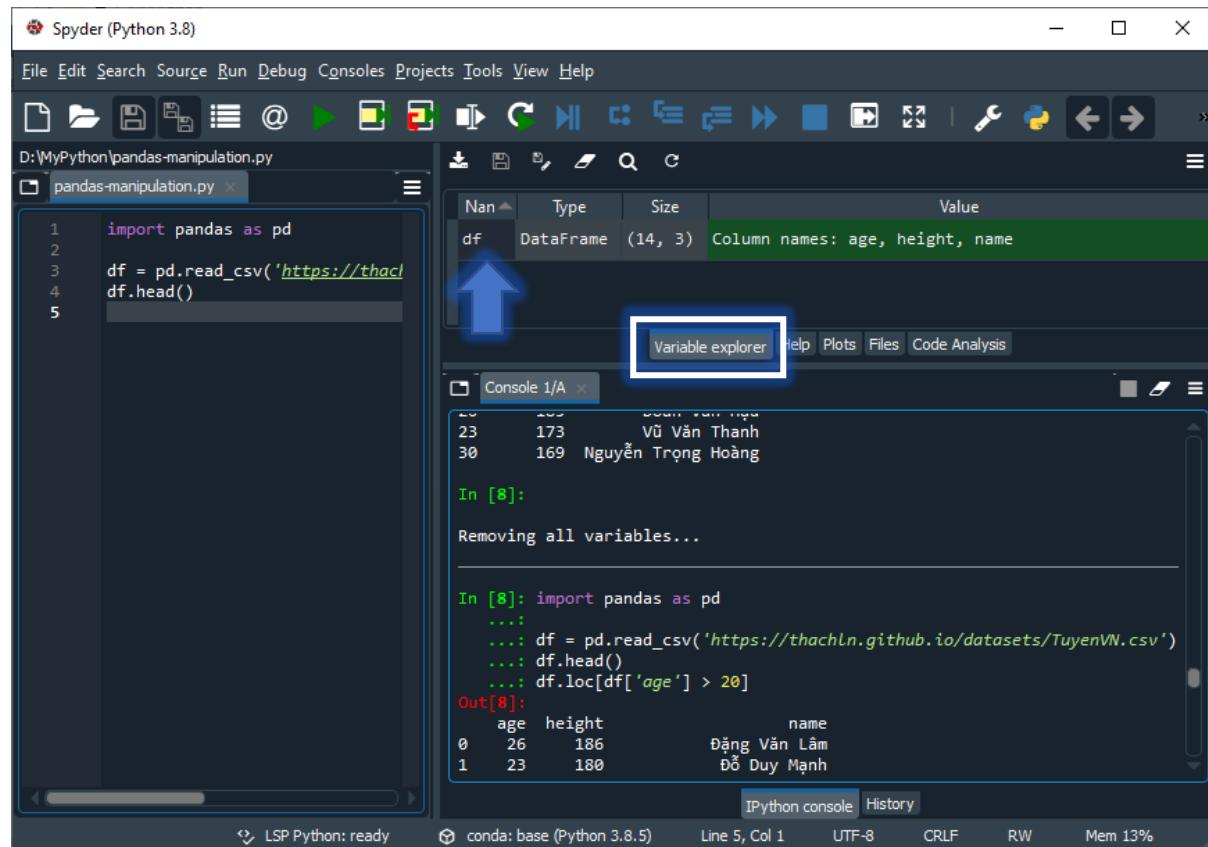
Hãy chạy 3 lệnh sau:

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

	age	height	name
0	26	186	Đặng Văn Lâm
1	23	180	Đỗ Duy Mạnh
2	20	185	Đoàn Văn Hậu
3	23	173	Vũ Văn Thành
4	30	169	Nguyễn Trọng Hoàng

Để ý cột đầu tiên trong phần kết quả có số thứ tự 0, 1, 2,... Đây chính là các giá trị **label** của DataFrame.

Hoặc trong phần mềm Anaconda Spyder, bạn xem biến bằng cách double-click vào tên biến trong khung “Variable explorer”, xem chỗ hình chữ nhật và mũi tên trong hình bên dưới:



Thì nội dung DataFrame `df` được hiển thị có cột Index như sau:

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

Index	age	height	name
0	26	186	Đặng Văn Lâm
1	23	180	Đỗ Duy Mạnh
2	20	185	Đoàn Văn Hậu
3	23	173	Vũ Văn Thành
4	30	169	Nguyễn Trọng Hoàng
5	24	180	Quế Ngọc Hải
6	24	173	Phạm Đức Huy
7	26	170	Đỗ Hùng Dũng
8	22	168	Nguyễn Quang Hải
9	24	176	Nguyễn Tuấn Anh
10	23	168	Nguyễn Phong Hồng Duy
11	22	178	Nguyễn Tiến Linh
12	23	170	Nguyễn Văn Toàn
13	24	168	Nguyễn Công Phượng

Cột Index chứa các **label** 0, 1, 2, ... cho mỗi dòng.

Bây giờ bạn thử thêm tham số `index_col=0` khi đọc file CSV như sau:

```
import pandas as pd  
df = pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv',  
index_col=0)
```

Xem giá trị DataFrame `df` thì kết quả như sau:

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

age	height	name
26	186	Đặng Văn Lâm
23	180	Đỗ Duy Mạnh
20	185	Đoàn Văn Hậu
23	173	Vũ Văn Thanh
30	169	Nguyễn Trọng Hoàng
24	180	Quế Ngọc Hải
24	173	Phạm Đức Huy
26	170	Đỗ Hùng Dũng
22	168	Nguyễn Quang Hải
24	176	Nguyễn Tuấn Anh
23	168	Nguyễn Phong Hồng Duy
22	178	Nguyễn Tiến Linh
23	170	Nguyễn Văn Toàn
24	168	Nguyễn Công Phương

Format Resize Background color Column min/max Save and Close Close

Lúc này cột age được dùng làm index, các giá trị 26, 23, ... gọi là label.

Nếu dùng lệnh head() thì bạn sẽ thấy cột age thấp hơn 1 dòng. Tức là age không còn là cột dữ liệu bình thường nữa (mà nó là cột làm Index).

```
df.head()
```

age	height	name
26	186	Đặng Văn Lâm
23	180	Đỗ Duy Mạnh
20	185	Đoàn Văn Hậu
23	173	Vũ Văn Thanh
30	169	Nguyễn Trọng Hoàng

Xem thêm các cột bằng cách xem thuộc tính columns bạn sẽ thấy chỉ có height và name như lệnh sau:

```
df.columns
```

```
Index(['height', 'name'], dtype='object')
```

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

Tóm lại label (nhãn) là một giá trị đặc biệt để định danh cho dòng dữ liệu. Nhãn không nhất thiết là duy nhất (như bạn thấy nhãn age trùng nhau trong ví dụ trên, xem thêm lệnh DataFrame.loc trong phần tiếp theo).

Chọn và lọc dữ liệu

Bảng dưới đây tóm tắt các nhu cầu thường dùng để thao tác dữ liệu với DataFrame (biến df).

Nhu cầu	Cú pháp	Kết quả
Chọn 1 cột dữ liệu	df[<i>tên cột</i>]	Series
Chọn nhiều cột dữ liệu	df[[<i>tên cột 1, tên cột 2,...</i>]]	DataFrame
Chọn các dòng dữ liệu	df.loc[<i>nhãn</i>]	Series hoặc DataFrame
Các trường hợp của nhãn.		
1) df.loc[<i>giá trị trong cột Index</i>]: trả lại các dòng dữ liệu có Index tương ứng.		
2) df.loc[<i>index1:index2</i>]: trả lại các dòng dữ liệu có Index từ index1 đến index2. Cách dùng này gọi là Slide rows.		
3) df.loc[<i>mảng giá trị true/false</i>]. <i>mảng giá trị true/false</i> thường dùng là biểu thức so sánh giá trị của cột. Vd: df.loc[df['tên cột'] > 25] Có thể dùng nhiều điều kiện so sánh như: df.loc[(df['tên cột 1'] > 25) & (df['tên cột 2'] <= 'abc')]		
Chọn dòng dữ liệu theo số thứ tự	df.loc[<i>số thứ tự tính từ 0</i>]	Series

Ví dụ minh họa:

Đoạn code sau lọc dữ liệu các cầu thủ có tuổi trên 25 và chiều cao bé hơn hoặc bằng 1m7

```
import pandas as pd  
df = pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')  
df.head()  
df.loc[(df['age'] > 25) & (df['height'] <= 170)]
```

Thêm / Xóa thuộc tính và dữ liệu quan sát (Observations)

Trong pandas gọi các dòng dữ liệu (rows) là observations (dữ liệu quan sát, hay dữ liệu thu thập được); các cột dữ liệu (columns) là attributes (thuộc tính). Các hàm liên quan thường dùng như sau:

- `df['tên cột mới'] = Giá trị | Biểu thức`

Ví dụ sau đây thêm 2 cột mới cho DataFrame. Cột sex được thêm với giá trị cứng. Cột “tuổi chẵn” (tôi dùng tiếng Việt có dấu và cả khoảng trắng để thử làm tên cột) được tính toán từ tuổi có chia hết cho 2 hay không?

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

df['sex'] = 'Nam'
df['tuổi chẵn'] = df['age'] % 2 == 0
df.head()
```

- Dùng hàm assign để thêm cột mới

Cú pháp chung:

```
df.assign(c1 = s1, c2 = s2,...)
```

Trong đó c1, c2 là tên cột (không có dấu nháy nhé); s1, s2,... là các biến có kiểu Series.

Ví dụ sau sẽ tính BMI cho cầu thủ bóng đá Việt Nam trong biến s_bmi. Sau đó dùng hàm assign để thêm cột vào DataFrame ban đầu:

```
import pandas as pd
df =
pd.read_csv('https://thachln.github.io/datasets/TuyenVN_2019.csv')
df.head()

s_met = df['Height'] / 100
s_kg = df['Weight']
s_bmi = s_kg / (s_met ** 2)

df = df.assign(bmi = s_bmi)
df.head()
```

• Thêm dòng dữ liệu

Quan sát ví dụ sau để biết cách thêm một dòng dữ liệu mới vào DataFrame:

```
import pandas as pd  
df = pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')  
df.head()  
  
df = df.append({'age': 31, 'height': 170, 'name': 'Lê Ngọc Thạch'},  
ignore_index=True)  
df
```

Chú ý giá trị tham số đầu tiên cho hàm df.append có dạng:

```
{'key1': value, 'key2': value}
```

gọi là dạng tự điển (Dictionary).

key1, key2,... ở đây nếu trùng với tên cột có sẵn trong DataFrame thì value được thêm vào cột tương ứng; nếu không có sẵn thì sẽ tạo cột mới.

Xóa cột với hàm drop

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

Kết quả hàm .drop (...) không làm thay đổi DataFrame. Nếu muốn DataFrame thay đổi sau khi xóa thì phải gán ngược kết quả lại cho biến df như sau:

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

Gom nhóm dữ liệu

Nếu bạn là lập trình viên thì ít nhiều có quen với khái niệm Group By. Trong Python cũng có hàm groupby để giúp chúng ta gom nhóm dữ liệu.

Dạng 1:

```
result = df.groupby(by='cột A')[ 'cột 1'].hàm()
```

Ý nghĩa: hàm .groupby sẽ duyệt các giá trị trong cột A của DataFrame df. Các giá trị trùng nhau được gom lại coi như là một giá trị. Cứ mỗi giá trị này thì hàm .groupby sẽ duyệt tất cả các giá trị tương ứng trong cột 1. Sau đó sẽ thực hiện hàm tính toán.

Ví dụ lệnh sau sẽ tổng hợp (hàm sum là tính tổng) doanh số (cột revenue) cho từng giá trị của sản phẩm (trong cột product) của tập dữ liệu trong biến df:

```
df.groupby(by='product')[ 'revenue'].sum()
```

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

Mở rộng một chút khi bạn cần tổng hợp nhiều cột dữ liệu thì thêm vào như sau:

```
df.groupby(by='cột A') ['cột 1', 'cột 2', ...].hàm()
```

Dạng 2:

```
df.groupby(['cột A', 'cột B', ...]).agg(  
    new_col=('cột 1', 'tên_hàm')  
)
```

Ý nghĩa: hàm groupby dạng này sẽ tạo ra cột mới (new_col) sẽ được thay bằng tên cột mà bạn muốn) bằng cách thực hiện tính toán **tên_hàm** trên **cột 1**. Giới hạn tính toán sẽ dựa vào các giá trị duy nhất của bộ các **cột A, cột B,...**

Ví dụ lệnh sau sẽ tạo thêm cột mới có tên là sum_rev cho DataFrame df. Cột sum_rev được tính bằng tổng của các giá trị trong cột revenue đối với các product cùng tên:

```
df.groupby(['product']).agg(sum_rev = ('revenue',  
    'sum'))
```

Bạn có thể gom nhóm nhiều cột theo dạng lệnh như sau:

```
df.groupby(['cột A', 'cột B', ...]).agg(  
    new_col_1=('cột 1', 'tên_hàm'),  
    new_col_2=('cột 2', 'tên_hàm')  
)
```

Hãy làm quen với việc tra cứu thêm tài liệu tại:

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.groupby.html>

Tinh huống xử lý dữ liệu thường gặp

Đối với các bạn mới làm quen với Python, Phân tích dữ liệu thì có thể tạm bỏ qua phần này. Nội dung này sẽ được sử dụng cho các vấn đề phức tạp hơn trong các ngày tiếp theo. Khi cần thiết thì tra cứu lại.

Đọc dữ liệu từ Internet và xem thông tin nhanh về dataframe

Ví dụ sau đọc file csv từ internet bằng thư viện pandas và gọi hàm info() để xem thông tin về dataframe. Trong R thì có hàm glimpse trong thư viện dplyr.

```
import pandas as pd  
iris = pd.read_csv('https://thachln.github.io/datasets/iris-data.csv')  
iris.info()  
  
<class 'pandas.core.frame.DataFrame'>
```

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

```
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype  
 --- 
  0   S.Length    150 non-null    float64
  1   S.Width     150 non-null    float64
  2   P.Length    150 non-null    float64
  3   P.Width     150 non-null    float64
  4   Species      150 non-null    object  
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Đọc dữ liệu từ file nén kiểu zip

Ví dụ sau đọc file .zip từ Internet, trong đó có file .csv với encoding (mã) của văn bản là latin-1.

```
import pandas as pd
fp = 'https://thachln.github.io/datasets/movie_reviews.zip'
df = pd.read_csv(fp, compression='zip', encoding='latin-1')
df.head()
```

Chuyển kiểu dữ liệu sau khi đọc từ file csv hoặc Excel

Khi dùng thư viện pandas đọc file csv hoặc Excel vào dataframe thì các cột dữ liệu số nguyên (int) mà có **dữ liệu trống** thì cột dữ liệu này sẽ bị tự động chuyển thành số thực (float). Để giữ đúng kiểu dữ liệu gốc ban đầu thì bạn phải kiểm tra lại cho chắc và tự ép kiểu.

Ví dụ file Excel hoặc CSV lưu vài đơn hàng như sau:

InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	84406B	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	6	12/1/2010 8:26	3.39		United Kingdom
536365	84029E	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	2	12/1/2010 8:26	7.65	17850	United Kingdom

Trong đó cột InvoiceNo và CustomerID là mã số của hóa đơn và mã khách hàng. Kiểu dữ liệu là số nguyên. Tuy nhiên khi dùng thư viện pandas để đọc vào dataframe thì cột CustomerID là số thực như sau:

```
import pandas as pd
file_path = 'https://thachln.github.io/datasets/Online_Retail_1.xlsx'
df = pd.read_excel(file_path)
df.head()
```

InvoiceNo	StockCode	Quantity	...	UnitPrice	CustomerID	Country
-----------	-----------	----------	-----	-----------	------------	---------

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

0	536365	85123A	6	...	2.55	17850.0	United Kingdom
1	536365	84406B	8	...	2.75	17850.0	United Kingdom
2	536365	84029G	6	...	3.39	NaN	United Kingdom
3	536365	84029E	6	...	3.39	17850.0	United Kingdom
4	536365	22752	2	...	7.65	17850.0	United Kingdom

[5 rows x 7 columns]

Kết quả cột CustomerID có số lẻ .0.

Để chuyển CustomerID trở về lại kiểu số nguyên (int) thì bạn phải xử lý giá trị trống rồi ép kiểu (type conversion) lại như sau:

```
df = df.dropna(subset=['CustomerID'])
df.CustomerID = df.CustomerID.astype(int)
# Hoặc
# df['CustomerID'] = df['CustomerID'].astype(int)
df.head()
```

Chú ý ở đây chỉ lấy ví dụ xóa dòng dữ liệu có CustomerID bị trống. Trong thực tế thì bạn phải quyết định xử lý bằng thuật toán hay xóa là tùy mục tiêu phân tích.

Chuyển kiểu dữ liệu Decimal128 khi đọc dữ liệu từ MongoDB

Trong trường hợp bạn dùng Python để đọc dữ liệu từ MongoDB (một phần mềm lưu trữ dữ liệu, MongoDB nằm ngoài phạm vi eBook này) thành Pandas DataFrame thì có thể một số cột dữ liệu có giá trị lớn sẽ được thiết lập kiểu Decimal128. Lúc đó các thao tác tính toán, + - sum,.. có thể gặp lỗi.

Một trong các cách giải quyết là chuyển kiểu dữ liệu sang dạng str, sau đó chuyển tiếp sang dạng float như sau:

```
df.col_name = df.col_name.astype(str).astype(float)
```

Lọc dữ liệu theo cột trong Pandas

Ví dụ đọc dữ liệu nghiên cứu các loài hoa trong dự án iris. Sau đó lọc dữ liệu theo cột cho biến X, và outcome y:

```
iris = pd.read_csv('https://thachln.github.io/datasets/iris-data.csv')
iris.head()
X = iris[['S.Length', 'S.Width', 'P.Length', 'P.Width']]
y = iris.Species
# Hoặc
y = iris['Species']
```

Ghi nhớ: Sử dụng cú pháp `dataframe[cols]` với `cols` là tên của cột hoặc array của các tên cột.

Lấy các cột dữ liệu trừ một cột

```
df.loc[:, df.columns != 'Tên cột']
```

Lọc dữ liệu theo dòng với điều kiện cho trước

Ví dụ: lọc các dòng dữ liệu trong dữ liệu Iris với điều kiện cột S.Length > 7.6.

```
df = pd.read_csv('https://thachln.github.io/datasets/iris-data.csv')
df.loc[df['S.Length'] > 7.6]
```

	S.Length	S.Width	P.Length	P.Width	Species
117	7.7	3.8	6.7	2.2	I.virginica
118	7.7	2.6	6.9	2.3	I.virginica
122	7.7	2.8	6.7	2.0	I.virginica
131	7.9	3.8	6.4	2.0	I.virginica
135	7.7	3.0	6.1	2.3	I.virginica

Lọc dữ liệu với điều kiện một cột khác rỗng

```
df[df['tên cột'].notnull()]
```

Lọc dữ liệu với nhiều điều kiện

```
df[df['tên cột1'].notnull() & (df['tên cột1'] > value)]
```

Chú ý: Phép toán luận lý và, hoặc (and, or) thì cú pháp trong Python là: &, |. Các bạn quen với ngôn ngữ lập trình C, Java hãy cẩn thận với thói quen &&, ||.

Lưu dataframe ra file csv

Sử dụng hàm `dataframe.to_csv(cvs path)`.

Ví dụ: `df.to_csv('out.csv')`

Lấy dòng dữ liệu cuối cùng của dataframe

```
last_row = df.tail(1)
```

Tiếp theo, lấy giá trị của một cột trong biến last_row ở trên thì dùng lệnh:

```
value = last_row['colum name'].values[0]
```

Tạo dataframe từ array hoặc list

Khi chỉ có 1 cột dữ liệu:

```
df = pd.DataFrame(array , columns =['col_name'])
```

Khi có hơn 1 cột dữ liệu:

Ví dụ bên dưới tạo data frame có 2 cột tên là col_1, col_2 có dữ liệu từ 2 biến mảng array1 và array2:

```
df = pd.DataFrame(list(zip(array1, array2)) , columns =['col_1', 'col_2'])
```

Thay đổi tên cột trong dataframe

```
df = df.rename(columns={'colname': 'newname'})
```

Tham khảo thêm tài liệu tại:

<http://pytotelearn.csd.auth.gr/b4-pandas/40/moddfcols.html>

Thay thế dữ liệu trong cột của dataframe

```
df['tên cột'] = df['tên cột'].replace('giá trị cũ', 'giá trị mới')
```

Thay đổi dữ liệu trong dataframe

```
df_pagination.loc[rowIndex, 'colname'] = newvalue
```

Chọn dữ liệu không trùng trong dataframe khác

Tình huống đặt ra là bạn có 2 dataframe df1 và df2 có cấu trúc giống nhau gồm 3 cột: c1, c2, c3.

Trong df2 có nhiều dòng dữ liệu đã xuất hiện trong df2. Bạn cần **lấy ra các dòng dữ liệu trong df2 mà các dòng này không xuất hiện trong df1** thì dùng đoạn code sau:

```
df3 = df2.merge(df1,  
                 on=['c1', 'c2', 'c3'],  
                 how='left',  
                 indicator=True).query('_merge ==  
"left_only"]').drop(columns='_merge')
```

Duyệt từng dòng dữ liệu của data frame

```
for index, item in df.iterrows():  
    cellVal = df.iloc[index]['ColName']  
    print(df['ColName'])
```

Duyệt từng phần tử của list / array với index

```
for index, item in enumerate(aList):  
    ...
```

Phát sinh dữ liệu mẫu

Khi cần tạo dữ liệu mẫu về tên người thì dùng thư viện sau:

```
pip install names
```

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

Đoạn code sau tạo dữ liệu mẫu cho data frame với 2 cột: fullname và numTask.
Dữ liệu gồm 10 dòng.

```
import names  
import random  
  
fullname = []  
numTask = []  
for i in range(0, 10):  
    fullname.append(names.get_full_name())  
    numTask.append(random.randint(0,20))
```

Sắp xếp dataframe

Tham khảo:

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort_values.html

Sắp xếp dataframe theo cột có kiểu là ngày thứ.

Ví bạn có DataFrame với cột dữ liệu day_of_week như hình bên dưới:

	day_of_week	total
0	thu	8623
1	mon	8514
2	wed	8134
3	tue	8090
4	fri	7827

Bạn cần sắp xếp lại DataFrame theo cột day_of_week theo đúng thứ tự của ngày từ Thứ Hai đến Chủ Nhật thì làm sao? Thậm chí bạn muốn thứ tự là Thứ Bảy, Thứ Hai,...Chủ Nhật thì làm sao?

Giải pháp được trình bày dưới đây.

Để minh họa thì bạn tổng hợp dữ liệu bằng cách đếm số quan sát theo thứ trong tuần (cột day_of_week):

```
import pandas as pd  
  
df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', sep=';')
```

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

```
s = df['day_of_week'].value_counts()  
print(s.reset_index())
```

index	day_of_week	
0	thu	8623
1	mon	8514
2	wed	8134
3	tue	8090
4	fri	7827

Bạn thấy cột index là thứ trong tuần nhưng không được xếp thứ tự.

Khai báo cats là mảng các ngày theo thứ tự bạn mong muốn. Sau đó áp dụng hàm `.reindex(cats)` sau khi tổng hợp dữ liệu.

```
cats = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun']  
df1 = df['day_of_week'].value_counts().reindex(cats)  
print(df1)
```

mon	8514.0
tue	8090.0
wed	8134.0
thu	8623.0
fri	7827.0
sat	NaN
sun	NaN

Bạn thử đổi lại thứ tự ngày bắt đầu là Chủ Nhật xem sao:

```
cats = ['sun', 'mon', 'tue', 'wed', 'thu', 'fri', 'sat']  
df1 = df['day_of_week'].value_counts().reindex(cats)  
print(df1)
```

sun	NaN
mon	8514.0
tue	8090.0
wed	8134.0
thu	8623.0
fri	7827.0
sat	NaN
Name: day_of_week, dtype: float64	

Truy cập mảng theo cú pháp x[a:b]

Khảo sát ví dụ sau:

```
a = [1, 2, 3, 4, 5]  
a[0:-1]  
a[1:]
```

`a[0:-1]` trả lại `[1, 2, 3, 4]`. Có nghĩa là không bao gồm phần tử cuối cùng.

`a[1:]` trả lại `[2, 3, 4, 5]`. Có nghĩa là không bao gồm phần tử đầu tiên.

Tìm các cột trong DataFrame có chứa từ nào đó

Ví dụ:

```
[x for x in df.columns if 'customer' in x]
```

Lệnh trên có ý nghĩa sau:

- df.columns lấy ra danh sách các header name của DataFrame df.
- Sau đó duyệt từng phần tử trong danh sách các header name bằng vòng lặp for, mỗi phần tử được gán vào biến x.
- Nếu chữ 'customer' (không bao gồm dấu nháy) xuất hiện trong biến x thì kết quả trả lại biến x
- Cú pháp dấu mốc ngoặc vuông [] ở đây có nghĩa là mảng các biến x.

Vì vậy lệnh trên sẽ trả lại array chứa các tên cột của DataFrame mà tên cột có chữ 'customer' trong đó.

Thêm tiền tố cho tên cột

```
df.add_prefix('abc_')
```

Mã hóa cột (encode) dữ liệu

Tình huống là DataFrame của bạn có cột status với giá trị 1 có nghĩa là enable; giá trị 2 có nghĩa là disable. Lúc đó bạn cần thêm 1 cột status_label thì dùng 2 lệnh như sau:

```
df.loc[df['status'] == 1, 'status_label'] = 'enable'  
df.loc[df['status'] == 2, 'status_label'] = 'disable'
```

Bài tập ngày 1

1) Thực thi code trong Spyder

Cho đoạn code định nghĩa hàm trong Python dưới đây:

```
# =====# Đây là hàm giới thiệu vài thông tin cá nhân. Trải nghiệm lệnh print.  
# =====def introduceMySelf():  
    name = 'Lê Ngọc Thạch'  
    print(name)  
  
    email = 'LNThach@gmail.com'  
    print('Email:', email)  
  
    phone = '09081234567'  
    print('Số {} là số điện thoại của tôi.'.format(phone))
```

Hãy lưu đoạn code vào một file RunCode.py. Dùng phần mềm Spyder để mở file. Hãy thực thi các lệnh để ra kết quả như bên dưới:

```
Lê Ngọc Thạch  
Email: LNThach@gmail.com  
Số 09081234567 là số điện thoại của tôi.
```

2) Sử dụng thư viện pandas

Download file tại link này về máy:

[“https://thachln.github.io/datasets/TuyenVN_2019.csv”](https://thachln.github.io/datasets/TuyenVN_2019.csv)

Thực hiện các việc sau:

- ① Sử dụng thư viện pandas để đọc file trên vào thành DataFrame có tên là **df**.
- ② Gọi lệnh phù hợp để tính giá trị mean của cân nặng trong biến Weight.
- ③ Sử dụng Spyder thì giá trị mean của Weight hiển thị là bao nhiêu?
(A) 68.91666666666666 (B) 68.9166666666667
(C) 69.91666666666666 (D) 69.91666666666667

3) Tra cứu tài liệu pandas

Hãy Google từ khóa “python pandas documentation” và mở link sau:

<https://pandas.pydata.org/docs/>

Tìm mục API Reference để mở link:

https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html

Hãy thực hiện các việc sau:

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

- ① Hãy giải thích ý nghĩa của tham số “**usecols**”.
- ② Hãy nâng cấp mã nguồn của bài 2 với yêu cầu: không đọc 2 cột “Club” và “BirthPlace” vào DataFrame df.

4) Tra cứu tài liệu hàm print

Hãy vào website sau để đọc tài liệu về string format trong Python.

https://www.w3schools.com/python/ref_string_format.asp

Hãy đó người bên cạnh một đến 2 cách dùng Placeholder và Formatting Types.

Thử thách cho bạn!

- 1) Viết chương trình bằng ngôn ngữ lập trình Python đọc file CSV từ thông số dòng lệnh, duyệt tất cả các cột dữ liệu tính toán các chỉ số sau và hiển thị ra màn hình các thông tin sau:

- Nếu cột dữ liệu là chuỗi: hiển thị ra các giá trị duy nhất của cột đó.
- Nếu cột dữ liệu là số: hiển thị ra các chỉ số min, max, mean.

Yêu cầu:

- Tuân thủ qui định mã nguồn ở đây:
<https://www.python.org/dev/peps/pep-0008>

- 2) Tự tạo ra dữ liệu trong file Excel mẫu gồm 2 cột: Số đơn hàng, Tổng số tiền. Viết chương trình bằng ngôn ngữ lập trình Python gồm 2 tham số dòng lệnh dạng như sau:

ExcelToChart.py <file Excel> <output folder>

ExcelToChart.py đọc file Excel từ tham số và xuất ra các file biểu đồ sau trong thư mục <output folder>

- chart_square.png: Biểu đồ gồm các điểm ảnh hình chữ nhật với trục x là “Số đơn hàng”, trục y là “Tổng số tiền” (tương ứng của Số đơn hàng).
- chart_line.png: Biểu đồ line nối các điểm (hình tam giác). Trong đó các điểm có x là “Số đơn hàng”, trục y là “Tổng số tiền” (tương ứng của Số đơn hàng)

1. Hiển thị giá trị “Tổng số tiền” bên trên mỗi điểm.

Yêu cầu:

- Tuân thủ qui định mã nguồn ở đây:
<https://www.python.org/dev/peps/pep-0008>
- Chọn một trong các tool trong bài viết sau để kiểm tra chất lượng code:
 - o <https://realpython.com/python-code-quality>
 - o <https://luminousmen.com/post/python-static-analysis-tools>

- 3) Lưu tất cả các ví dụ, bài tập (gọi chung là sản phẩm) mà bạn đã làm lên một trong các nền tảng Github, Gitlab, DevAzure. Sau đó viết bài thu hoạch tóm tắt các nội dung đã lưu.
- 4) Chọn ít nhất một người bạn học chung, hỏi bạn bài 3) ở đâu rồi tự lấy các sản phẩm của bạn về xem mã nguồn, tự chạy lại theo như lời giới thiệu trong các sản phẩm hoặc trong bài thu hoạch. Sau đó tự tổng hợp và viết bài tóm tắt, đánh giá về các sản phẩm của bạn.

Gợi ý: trả lời các câu hỏi sau:

- Bạn xem source của cả bạn học được cái gì mới?
- Các chức năng trong từng sản phẩm mà tác giả giới thiệu thì bạn có chạy được không?

Chọn ngẫu nhiên ít nhất 3 đoạn mà nguồn, hỏi tác giả chức năng liên quan đến 3 đoạn mã nguồn đó rồi chạy chương trình để kiểm tra các chức năng này có đúng không?

Ngày 2 - Chủ đề: Biểu đồ

Có thể nói phân tích dữ liệu (Data Analysis) hoặc khai phá dữ liệu (Data Mining) là một quá trình gồm nhiều bước. **Bước đầu tiên** là **xác định câu hỏi**. Tức là xác định rõ vấn đề cần giải quyết với các mục tiêu cụ thể dưới dạng các giả thuyết (hypothesis). **Bước thứ hai** là đi **tập hợp dữ liệu** (Selecting data, hoặc Collecting data). Có dữ liệu rồi thì chưa chắc dữ liệu có đầy đủ thông tin, cần phải thực hiện **bước thứ ba - tiền xử lý dữ liệu** (Preprocessing data). **Bước thứ tư** là **chuyển đổi dữ liệu** (Transforming data). Đôi khi dữ liệu có quá nhiều thuộc tính sẽ ảnh hưởng đến hiệu quả và sự phức tạp của các thuật toán. Cho nên việc giảm số lượng các thuộc tính để giúp cho quá trình phân tích hiệu quả hơn mà không làm mất thông tin là quan trọng. **Bước thứ năm** là **lưu trữ dữ liệu** (Storing data). Dữ liệu sau khi đã được chuyển đổi (transformed data) sẽ rất quý bởi vì đã tốn rất nhiều công sức và tiền của để có được dữ liệu "tốt" và "sạch sẽ". Vì thế lưu trữ để làm tài sản là đương nhiên. Đặc biệt là dạng thức (format) lưu trữ như thế nào để phục vụ tốt cho việc truy xuất, phân tích, và khai phá là rất quan trọng. **Bước thứ sáu** là **phân tích** (analysis) hoặc **khai phá** (mining) thông tin. Bước này là khâu để hiểu dữ liệu, đặc biệt là thấu hiểu dữ liệu thông qua các mối tương quan bằng nhiều phương pháp phân tích khác nhau, các phương pháp tham số (parametric), phi tham số (non-parametric) và các thuật toán máy học (machine-learning³). **Bước thứ bảy** là **đánh giá kết quả** (Evaluate results). Bước này đánh giá khả năng tiên lượng (predictive capability) hoặc dự báo (forecast) của mô hình trên cơ sở dữ liệu đã có. Đặc biệt là kiểm định lại giả thuyết đã đặt ra ở bước một, hoặc tìm câu trả lời cho câu hỏi đã xác định (dù là có, hoặc là không, hoặc một lý giải hợp lý). Đối với người làm về khoa học dữ liệu thì còn một bước nữa là **làm báo cáo** (Report). Trong giới nghiên cứu thì thông thường báo cáo là dạng bài báo khoa học (paper). Trong giới doanh nghiệp thì báo cáo thông thường là báo cáo kết quả cho cấp trên hoặc các bên liên quan.

Để bắt đầu cho **bước thứ Sáu** thì nhìn dữ liệu dưới dạng biểu đồ một cách trực quan sẽ giúp chúng ta hiểu được bức tranh tổng thể của dữ liệu, đôi khi có thể thấy ngay các thông tin ẩn (hidden) đằng sau "bức tranh" mà nếu chỉ nhìn con số không thôi thì rất khó phát hiện. Ngày thứ hai này chúng ta sẽ dạo qua các loại biểu đồ và làm quen với công việc phân tích biểu đồ.

Sau ngày thứ hai này chúng ta sẽ biết hoặc làm được các việc sau:

³ Bản thân tôi thì không thích dịch **machine-learning** là máy học hoặc học máy vì nó "tối nghĩa" sẽ dễ gây ảo tưởng (ít ra là đối với tôi). Tôi nghĩ dùng từ mô hình hóa bằng máy (nếu viết tiếng Anh là machine-model) thì có lẽ dễ liên tưởng đến bản chất hơn. Tức là machine-learning về bản chất là xây dựng mô hình (thông thường là dựa trên toán học, tức là thông qua các hàm số và các phép tính toán) để mô phỏng nguyên lý của dữ liệu từ đầu vào cho đến đầu ra. Tuy nhiên với công nghệ tính toán ngày càng mạnh và nhiều thiết bị điện tử có thể thu thập nhiều thông tin (như camera, sensor – cảm biến) để chuyển cho máy xử lý và có thể thay đổi mô hình (model) cho phù hợp với ngoại cảnh. Tức là các mô hình toán học không còn cố định nữa, mà có thể thay đổi để thích nghi với dữ liệu mới. Đây là "khả năng" tuyệt vời của máy theo hướng bắt chước con người chúng ta – đó là khả năng "học". Chính vì vậy dùng từ **máy học** thì không sai nhưng nếu hiểu là **mô hình hóa bằng máy** thì sẽ giúp chúng ta học chuyên sâu về phân tích dữ liệu và trí tuệ nhân tạo (Artificial Intelligent) nói chung sẽ ít nhầm lẫn và ảo tưởng hơn.

- ① Biết ý nghĩa và mục đích cơ bản của các loại biểu đồ. Cảm nhận được nếu dùng biểu đồ để trình bày thì giúp chuyển tải nhiều thông tin thế nào. Đặc biệt khám phá được nhiều thông tin ẩn đằng sau dữ liệu đang có.
- ② Sử dụng được các biểu đồ phổ biến trong Python.

Ngày thứ hai này sẽ gồm 5 bài:

Bài 6: Tóm tắt và giúp các bạn phân biệt được mục đích cơ bản của các loại biểu đồ.

Bài 7: Giúp bạn làm quen và cảm nhận với cách vẽ biểu đồ bằng R. Đặc biệt sử dụng thư viện ggplot2 để vẽ biểu đồ chất lượng cao.

Bài 8: Giúp bạn làm quen và cảm nhận với cách vẽ biểu đồ bằng Python. Trong bài này cũng giúp bạn sử dụng sử dụng ggplot2 (vốn là của R) trong Python.

Bài 9: Sưu tầm vài nguyên tắc soạn biểu đồ.

Bài 10: Giúp bạn làm quen với thư viện vẽ biểu đồ rất phổ biến trong Python.

Bài 11: Giúp bạn làm quen với thư viện vẽ biểu đồ hỗ trợ tương tác và làm ứng dụng trên Web.

Bài 6: Các loại biểu đồ

Trong bối cảnh ngày nay có quá nhiều dữ liệu thì việc cảm nhận nhanh hoặc nắm bắt bức tranh tổng thể của dữ liệu rất là quan trọng. Việc nhìn dữ liệu dưới dạng hình ảnh chắc chắn sẽ sinh động hơn nhiều. Ngoài ra hình ảnh của dữ liệu có thể cho chúng ta khám phá nhiều thông tin đằng sau mà nếu chỉ có con số không thôi thì ta không biết được. Như vậy nếu bạn biết sử dụng được các kỹ thuật để trực quan hóa dữ liệu là một lợi thế lớn trong công việc của mình.

Bài này sẽ giúp các bạn sử dụng biểu đồ nào phù hợp với mục đích của mình. Phần mã nguồn và cách sử dụng lệnh để vẽ biểu đồ bằng Python sẽ được trình bày tương ứng trong Bài 9 và Bài 10.

Chúng ta cùng ôn lại mục đích của các loại biểu đồ mà ít nhiều các bạn đã từng biết hoặc đã làm quen.

Bảng bên dưới trình bày mục đích hoặc chủ định của các bạn muốn làm gì ở cột **bên trái**. Tương ứng với chủ định thì cột **bên phải** sẽ cho biết nên dùng biểu đồ gì.

Mục đích	Biểu đồ
Cần nhìn thấy bức tranh tổng thể về phân số dữ liệu	Histogram Boxplot
Biểu diễn phân bố dữ liệu Tóm tắt thống kê: biểu diễn các 5 giá trị quan trọng của dữ liệu: min, max, lower quartile, up quartile, mean (nhỏ nhất, lớn nhất, bách phân vị 25%, bách phân vị 75%, trung bình)	
Biểu diễn dữ liệu theo thời gian	Time series
Cần so sánh	
So sánh (giá trị) của nhiều biến , hoặc cần thấy tầng số dữ liệu. So sánh giá trị một biến thay đổi theo thời gian .	Bar chart/Bar plot: biểu đồ cột (tên khác là Column chart) Line chart: biểu đồ đường kẻ.
Trường hợp mốc thời gian ít (dưới 10)	Vertical bar chart: biểu đồ thanh đứng
So sánh nhiều biến hoặc nhiều nhóm	Radar chart (tên khác là Spider chart): biểu đồ mạng nhện.
Cần nhận biết sự tương quan giữa hai hoặc nhiều biến	
Tương quan giữa 2 biến	Plot (dữ liệu của 2 biến lên 2 trục của biểu đồ).

Tương quan giữa 2 biến trong đó có chia theo nhóm

Scatterplot.

Scatterplot có chia nhóm.

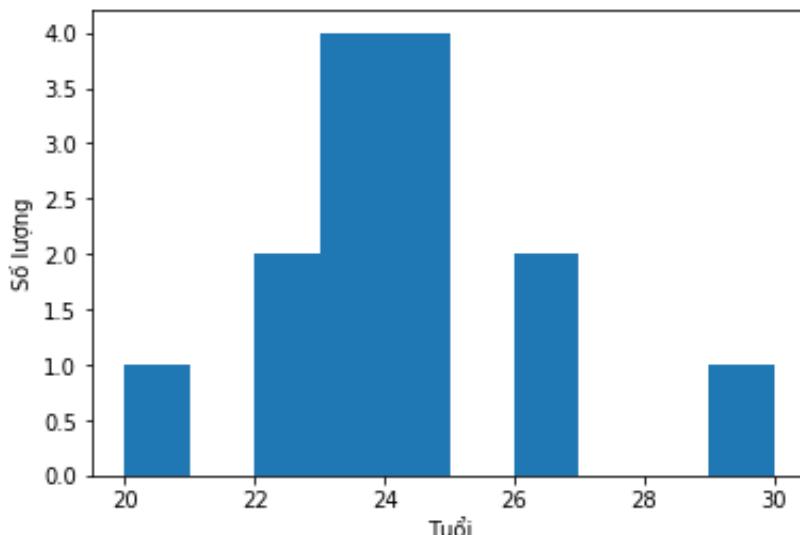
Tương quan giữa nhiều biến

Scatterplot nhiều biến.

Biểu đồ phân bố dữ liệu (histogram)

Khi bạn muốn nhìn các giá trị của một biến phân bố như thế nào thì chúng ta cần đến biểu đồ histogram. Ví dụ phân bố tuổi của đội tuyển bóng đá Nam của Việt Nam như sau:

Biểu đồ phân bố độ tuổi của tuyển bóng đá nam Việt Nam.



Biểu đồ này cho thấy có khoảng 3 tuyển thủ tuổi từ 20 đến 22, 8 tuyển thủ tuổi từ 22 đến 24 và vài tuyển (từ 1 đến 3) thủ trên 24. Như vậy có thể mường tượng tuổi của các tuyển thủ còn rất trẻ, phần lớn là từ 22 đến 24.

Biểu đồ so sánh (Comparison Plots)

Khi có nhu cầu so sánh nhiều biến, hoặc so sánh giá trị của biến theo thời gian thì dùng các biểu đồ so sánh. Biểu đồ thông dụng là biểu đồ thanh (**Bar chart**) hay còn gọi biểu đồ cột (column chart). Để nhìn dữ liệu theo thời gian thì biểu đồ **Line** là phù hợp. Trong trường hợp giá trị theo thời gian ít (dưới 10 cột mốc) thì có thể dùng biểu đồ thanh đứng (vertical bar chart). Trong trường hợp nhiều biến hoặc nhiều nhóm thì biểu đồ mạng nhện (**Radar chart**, **Spider chart**) được sử dụng.

Line Chart

Các line chart thường được dùng để hiển thị các giá trị định lượng (quantitative values) trong khoảng thời gian liên tục.

Trục x (x-axis) biểu diễn thời gian, trục y (y-axis) biểu diễn giá trị của biến cần quan sát.

Cách dùng:

- ✓ Line chart phù hợp để so sánh giá trị của nhiều biến và trực quan hóa (visualizing) các xu hướng cho cả hai trường hợp có một biến hoặc nhiều biến.
- ✓ Đối với các chu kỳ thời gian nhỏ (cỡ 10 trở lại) thì các biểu đồ thanh đứng (vertical bar chart) có thể được sử dụng.

Bar Chart – biểu đồ thanh

Mỗi thanh trong biểu đồ thể hiện tương ứng với một giá trị. Có 2 dạng biểu đồ thanh: dạng thanh đứng (vertical bar) và dạng thanh ngang (horizontal bar).

Cách dùng:

- So sánh các biến trong các danh mục. Đôi khi biểu đồ thanh đứng được dùng để thể hiện giá trị của một biến thay đổi theo thời gian.

Radar chart

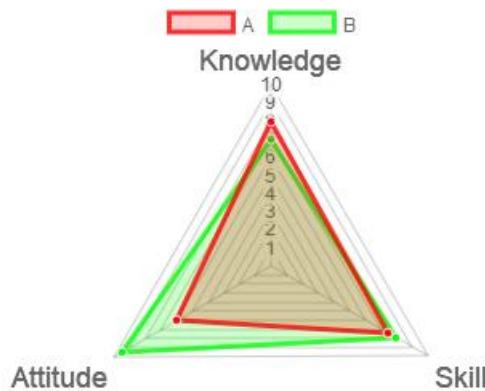
Radar chart có nhiều tên khác như spider chart (biểu đồ mạng nhện – do nó giống cái mạng nhện), hoặc web chart. Radar chart thể hiện nhiều biến trên một đa giác. Mỗi giá trị của biến tương ứng với mỗi đỉnh của đa giác.

Sử dụng:

- Radar chart dùng để so sánh nhiều giá trị biến định lượng trong một hoặc nhiều nhóm.
- Radar chart cũng hữu ích khi cần hiển thị giá trị cao/thấp trong tập dữ liệu.

Ví dụ:

Để hiển thị điểm các kỹ năng của sinh viên hoặc ứng viên thì có thể dùng radar chart. Một ví dụ đơn giản là cần so sánh điểm ASK của hai ứng viên A và B thì có cái hình như sau:



Ghi chú một chút ASK là viết tắt của Thái độ (Attitude), Kỹ năng (Skill) và Kiến thức (Knowledge). ASK thường được các nhà tuyển dụng đánh giá ứng viên.

Biểu đồ tương quan (Relation Plots)

Khi bạn có nhu cầu thể hiện mối liên quan (relationships) giữa các biến thì các loại biểu đồ sau đây là hữu ích.

Scatter plot đơn giản

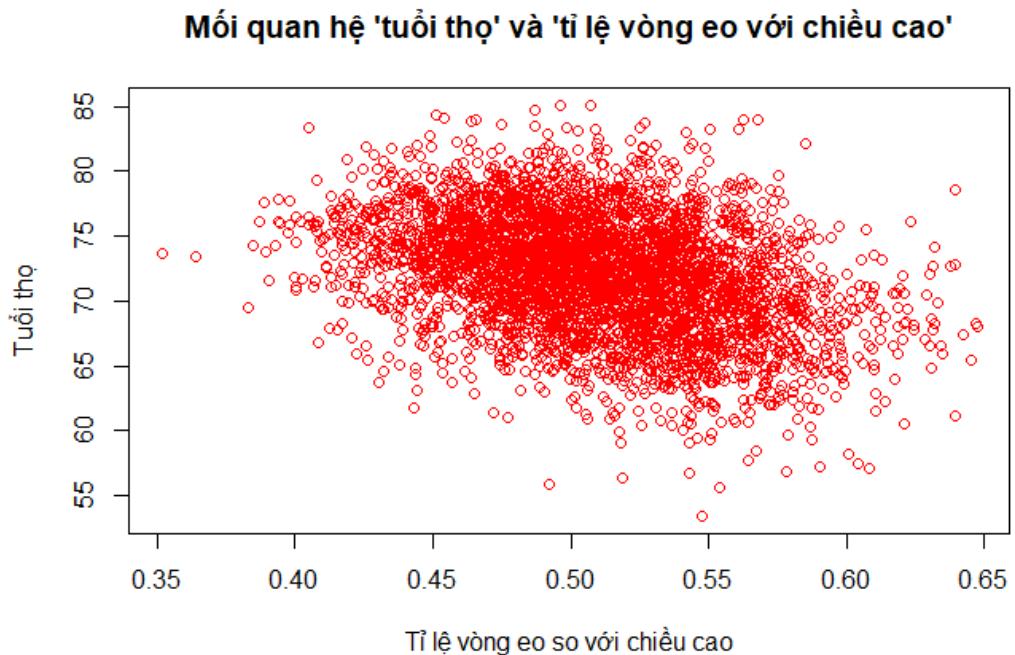
Scatter plot thể hiện các điểm (point) của hai biến số, hoặc nói cách khác là giá trị của một biến được thể hiện trong mối tương quan của hai trục x và y. Biểu đồ này cũng rất hữu ích khi cần quan sát mối tương quan giữa hay nhiều biến số trong nhiều nhóm.

Để dễ hình dung thì tôi lấy câu “Vòng bụng càng lớn thì vòng đori càng ngắn” của người Việt chúng ta minh họa biểu đồ.

Hình bên dưới là kết quả nghiên cứu của nhóm DataInDream⁴ theo dõi 3960 người có độ tuổi từ 18 đến 69 tuổi cho đến cuối đời.

Để khách quan giữa người có chiều cao khác nhau thì cách so bụng có lớn hay không thì tính bằng cách lấy số đo vòng bụng (còn gọi là vòng eo – waist size) chia cho chiều cao được tính cùng đơn vị. Tỉ số này gọi là WhtR (Waist and height ratio).

⁴ Đây là nhóm tôi tự đặt với ý là sẽ tự phịa ra dữ liệu để minh họa cho các bạn dễ nắm ý tưởng cần trình bày. Không phải tôi phịa lung tung mà sẽ cố gắng bám vào các nghiên cứu thực tế nhưng việc xin dữ liệu thật không mấy dễ dàng. Vì vậy khi các bạn thấy chỗ nào tôi ghi nguồn dữ liệu từ nhóm này thì biết rồi đấy! Chỉ nên tập trung vào ý tưởng và kỹ thuật đang trình bày chứ không nên để ý tính chính xác của dữ liệu.



Khái niệm tương quan

Hệ số tương quan

Hệ số tương quan R^2 , tiếng Anh là R squared: là một chỉ số thống kê để đo mối tương quan của hai đối tượng (cá thể - nói theo ngôn ngữ thống kê). R squared dao động từ 0.0 đến 1.0.

0: có nghĩa là không có liên quan

Giá trị càng cao cho thấy mức độ liên quan càng lớn.

1: Giá trị liên quan cao nhất.

Ví dụ tính toán hệ số tương quan của 2 dãy số trong Python:

```
import numpy as np
x_values = [1,2,3]
y_values = [4,5,7]

correlation_matrix = np.corrcoef(x_values, y_values)
correlation_xy = correlation_matrix[0,1]
r_squared = correlation_xy**2

print(r_squared)
```

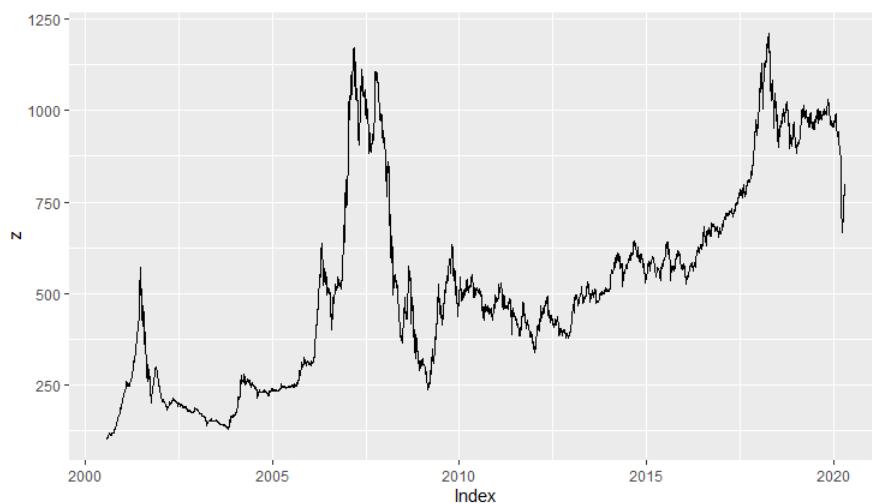
```
0.9642857142857141
```

Trong các bài phân tích về hồi qui tuyến tính, hồi qui logistic chúng ta sẽ gặp lại khái niệm này và sẽ có cơ hội phân tích sâu hơn một chút.

Biểu đồ dữ liệu theo thời gian

Một loại dữ liệu đặc biệt là dữ liệu theo thời gian (time series).

Ví dụ nếu bạn quan tâm đến chỉ số chứng khoán của Việt Nam (VNIndex) trong vài năm thì cần hình dung được bức tranh chung của nó như thế nào. Hình bên dưới là chỉ số VNIndex từ năm 2000 đến tháng 4/2020.

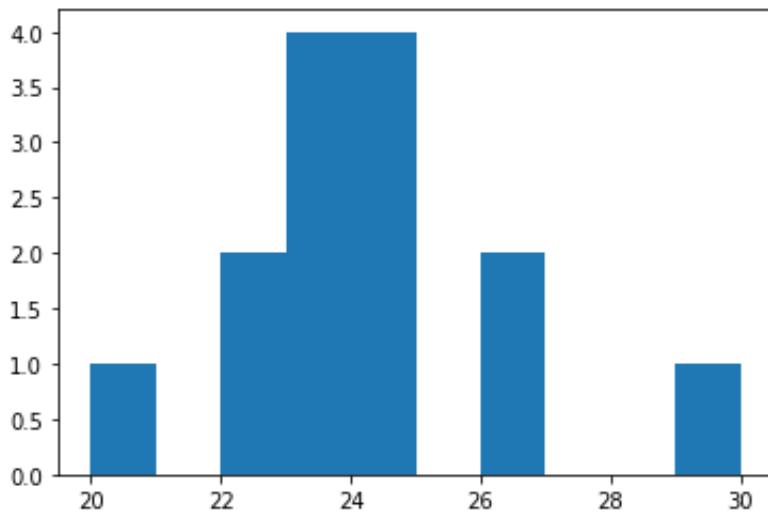


Bài 7: Vẽ biểu đồ trong Python

Tương tự Bài 9, bài này giúp các bạn yêu thích Python có thể vẽ nhanh được các loại biểu đồ được giới thiệu trong Bài 8. Mục đích của bài này là giúp bạn làm quen với kỹ thuật vẽ biểu đồ với Python thôi chứ không đi sâu vào phân tích và giải thích.

Biểu đồ phân bố dữ liệu - Histogram

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')  
plt.hist(df["age"])
```

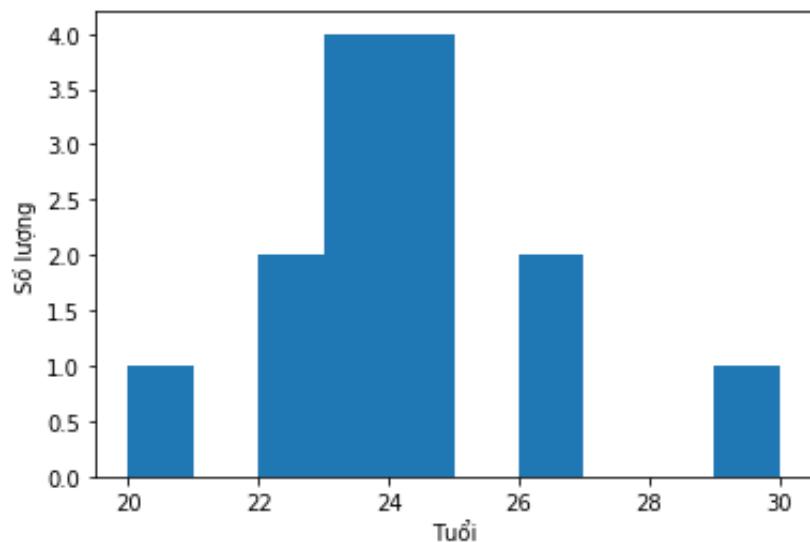


Ghi chú:

- Trong Python, cú pháp để truy xuất cột dữ liệu trong data frame là dùng cặp dấu ngoặc vuông [], bên trong cặp [] là tên cột bao đóng bởi cặp dấu nháy (nháy đơn hoặc đôi đều được). Ví dụ data: `df["age"]`

Trang trí thêm trực x và y:

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read_csv('https://thachln.github.io/datasets/TuyenVN.csv')  
plt.xlabel('Tuổi')  
plt.ylabel('Số lượng')  
plt.hist(df['age'])
```



Để trang trí thêm tiêu đề cho biểu đồ thì trong thư viện **matplotlib** cung cấp đối tượng **Figure**.

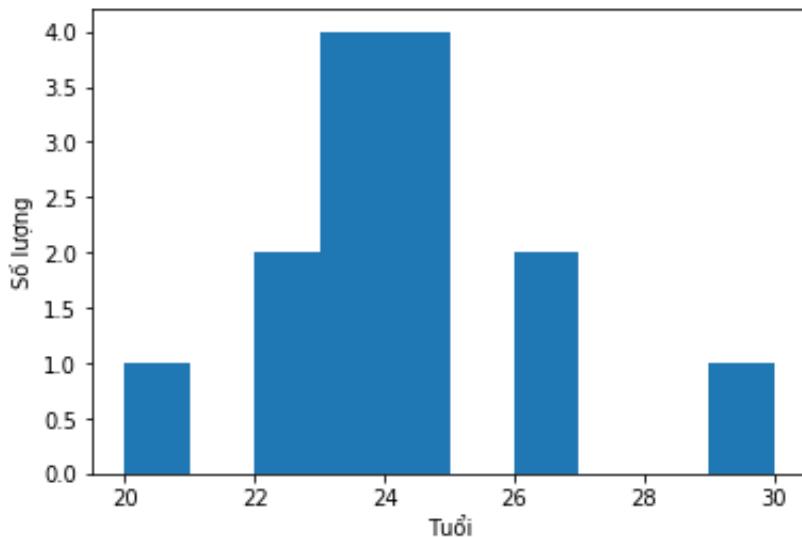
```
import pandas as pd
import matplotlib.pyplot as plt

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

fig = plt.figure()
fig.suptitle('Biểu đồ phân bố độ tuổi của tuyển bóng đá nam Việt Nam.')
plt.xlabel('Tuổi')
plt.ylabel('Số lượng')
plt.hist(df['age'])
```

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

Biểu đồ phân bố độ tuổi của tuyển bóng đá nam Việt Nam.



Biểu đồ phân bố dữ liệu – Boxplot

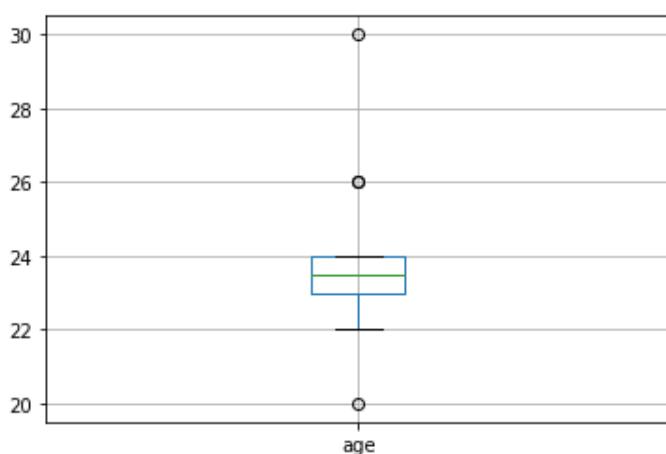
Thư viện pandas ngoài việc cung cấp chức năng đọc dữ liệu từ file vào data frame nó có luôn hàm xem dữ liệu dạng boxplot. Dưới đây là ví dụ:

```
import pandas as pd

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

Bạn thấy với Python thì hàm `boxplot` được gọi trực tiếp từ đối tượng chứa dữ liệu (biến `df`) luôn chứ không cần phải qua hàm vẽ nào hết.

Về logic thì rất dễ hiểu: lệnh `df.boxplot(column = ['age'])` ý nói là cho tôi thấy dữ liệu dạng boxplot với lựa chọn là chỉ vẽ cột `age` thôi.



Một cách khác là sử dụng thư viện Seaborn với code như sau:

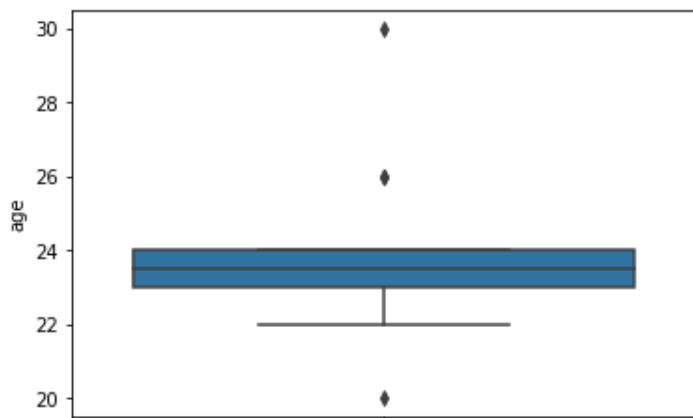
```
import pandas as pd
```

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

```
import seaborn as sns

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

sns.boxplot(y=df["age"])
```



Biểu đồ so sánh - Line Chart – 1 biến

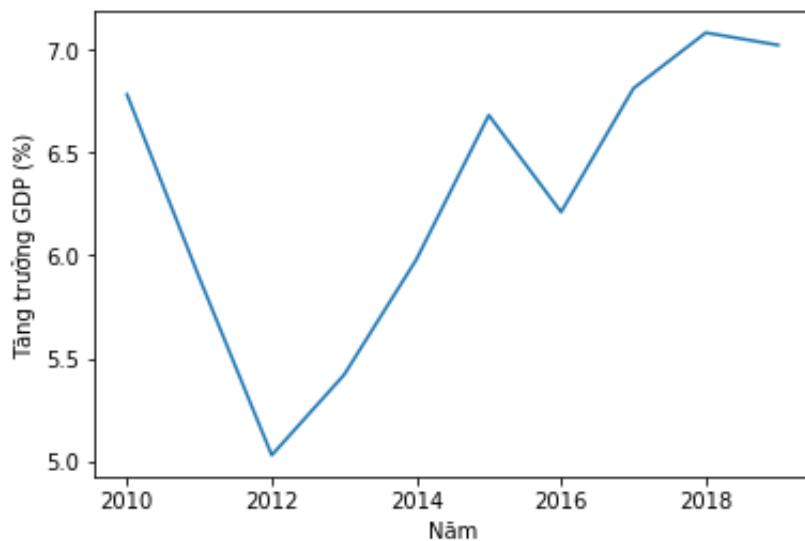
```
import pandas as pd
import matplotlib.pyplot as plt

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.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'])
```

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



Biểu đồ so sánh tốc độ tăng trưởng GDP theo năm dùng biểu đồ thanh BarPlot

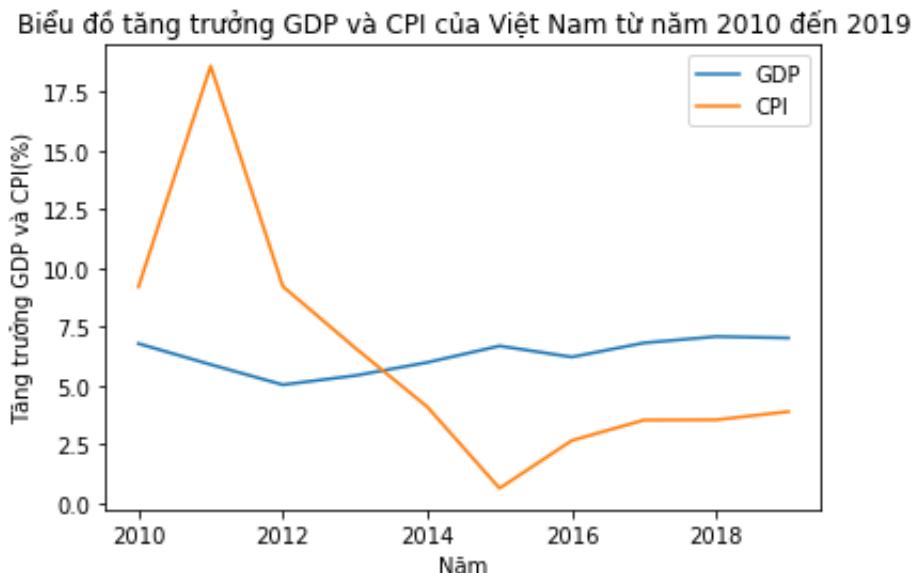
Biểu đồ so sánh - Line Chart – 2 biến

```
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, 6.6, 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.title('Biểu đồ tăng trưởng GDP và CPI của Việt Nam từ năm 2010 đến 2019')
plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP và CPI (%) ')
plt.plot(df['year'], df['gdp'], label = 'GDP')
plt.plot(df['year'], df['cpi'], label = 'CPI')
plt.legend()
```



Biểu đồ so sánh - Bar Chart

Trong Python, để vẽ biểu đồ BarPlot thì dùng thư viện Seaborn. Seaborn không phải là một thư viện độc lập mà nó được phát triển trên nền của Matplotlib. Seaborn có thể dùng kết hợp với Matplotlib để vẽ nhiều biểu đồ hơn, nhiều lệnh để trang trí biểu đồ đẹp hơn.

Chúng ta có thể kết hợp hai thư viện Matplotlib và Seaborn để vẽ và trang trí biểu đồ như bên dưới:

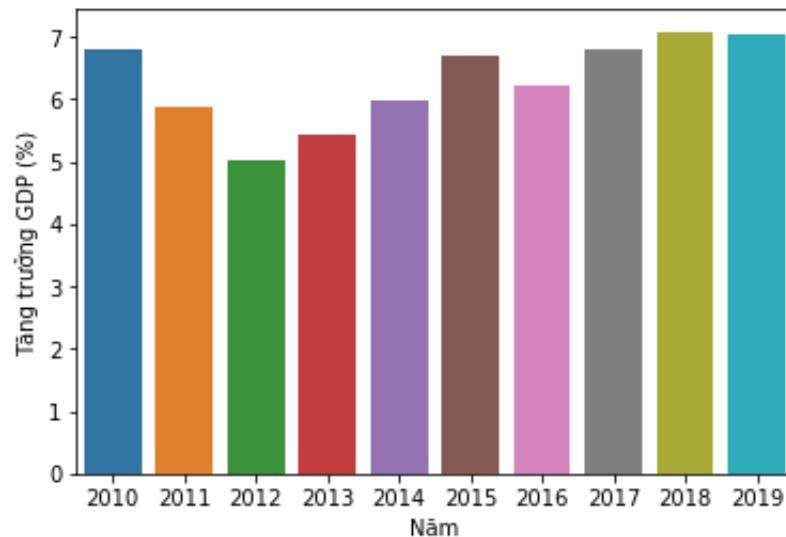
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

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.title('Biểu đồ tăng trưởng GDP của Việt Nam từ năm 2010 đến 2019')
fig = sns.barplot(df['year'], df['gdp'])
plt.xlabel('Năm')
plt.ylabel('Tăng trưởng GDP (%)')
```

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



Biểu đồ so sánh - Bar Chart có phân nhóm

Có thể gọi hàm `.plot.bar(...)` trực tiếp từ DataFrame. Ví dụ sau tổng hợp dữ liệu bằng hàm `.groupby()` []. Kết quả lưu vào DataFrame `dfg` gồm 2 cột dữ liệu `duration` và `age` tương ứng là thời gian cuộc gọi trung bình và tuổi trung bình; và Index của DataFrame là `education`. Sau đó vẽ biểu đồ trên kết quả `groupby`.

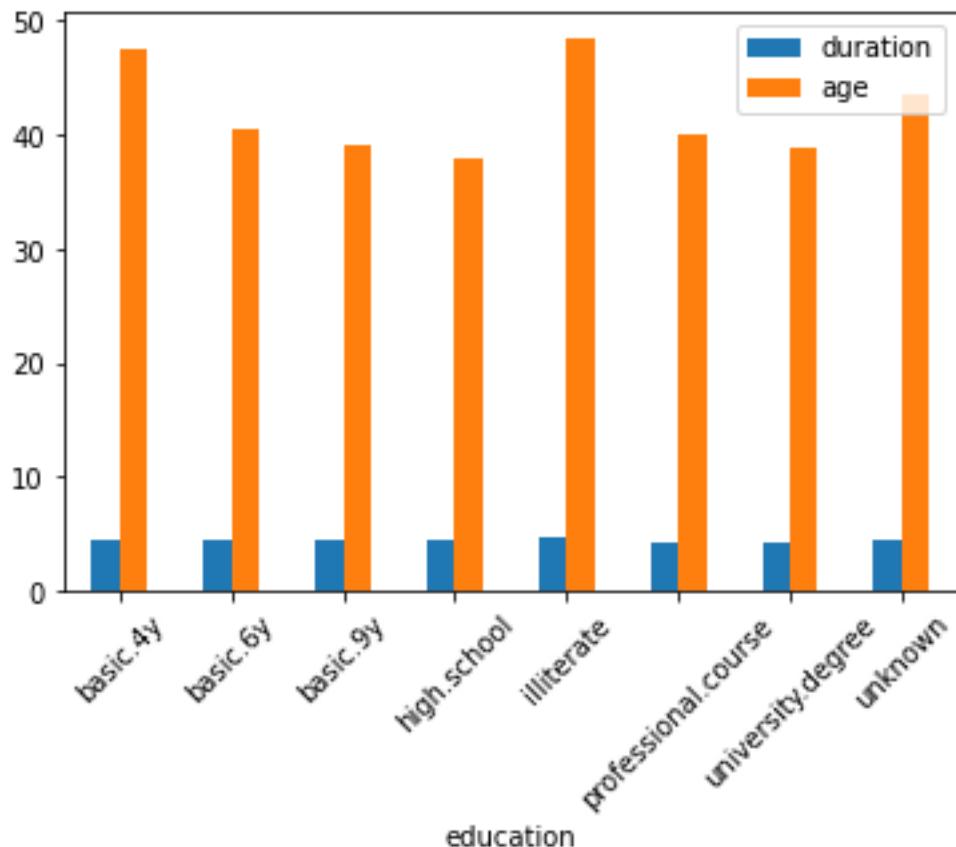
```
import pandas as pd
df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', sep=';')

# Tính thời lượng cuộc gọi bằng Phút
df['duration'] = df['duration'] / 60
print(df.head())
print(df.columns)

dfg = df.groupby('education')['duration', 'age'].mean()

ax = dfg.plot.bar(rot = 45)
```

education	duration	age
basic.4y	4.413797	47.596504
basic.6y	4.406908	40.448953
basic.9y	4.354864	39.061208
high.school	4.348114	37.998213
illiterate	4.612963	48.500000



Biểu đồ so sánh - Radar Chart

Phần code này mang tính giới thiệu cho các bạn cảm nhận được cách vẽ biểu đồ radar. Tôi tạm bỏ qua phần giải thích vì nó mang tính kỹ thuật hơi nhiều. Tạm thời bạn hãy bỏ qua sự phức tạp kỹ thuật này.

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

df = pd.DataFrame({
    'Employee': ['A', 'B'],
    'Knowledge': [8, 7],
    'Skill': [7.5, 8],
    'Attitude': [6, 9.5]
})

attributes = list(df.columns[1:])
values = list(df.values[:, 1:])
employees = list(df.values[:, 0])
```

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

```
angles = [n / float(len(attributes)) * 2 * np.pi for n in
range(len(attributes))]

# Close the plot
angles += angles[:1]

values = np.asarray(values)
values = np.concatenate([values, values[:, 0:1]], axis=1)

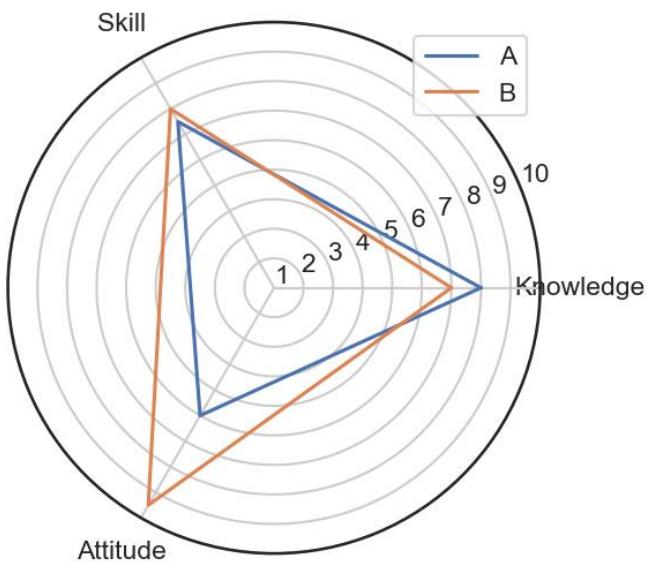
# Create figure
plt.figure(figsize=(4, 4), dpi=150).suptitle('Biểu đồ so sánh tam giác
ASK của 2 người.')
# Create subplots
for i in range(2):

    ax = plt.subplot(1, 1, 1, polar=True)
    ax.plot(angles, values[i], label = employees[i])
    ax.set_yticks([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

    ax.set_xticks(angles)
    ax.set_xticklabels(attributes)

plt.legend()
# Set tight layout
plt.tight_layout()
# Show plot
plt.show()
```

Biểu đồ so sánh tam giác ASK của 2 người.

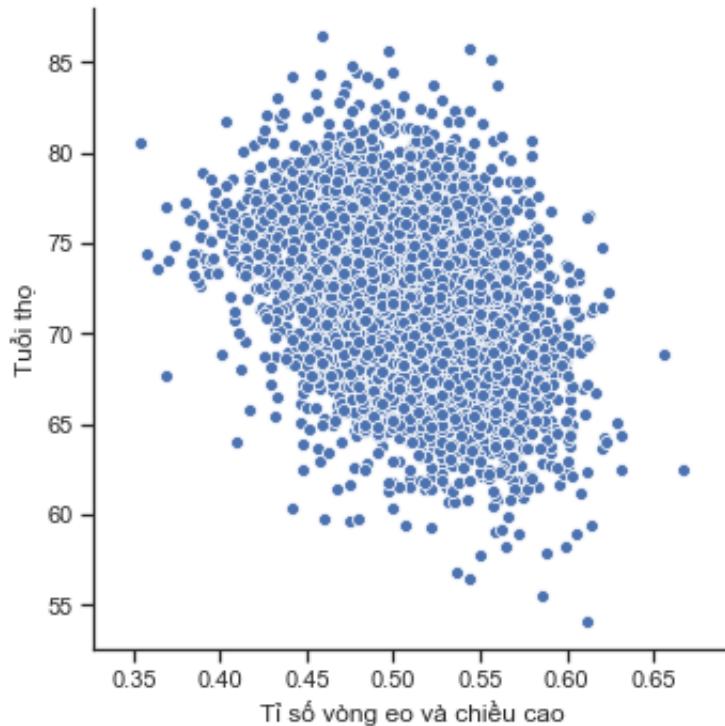


Biểu đồ tương quan đơn giản

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

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

df['whtr'] = df['waist'] / df['height']
plt.figure().suptitle("Mối quan hệ 'tuổi thọ' và 'tỉ lệ vòng eo với chiều cao'")
sns.set(style='ticks')
sns.relplot(x='whtr', y='life', data=df)
plt.xlabel('Tỉ số vòng eo và chiều cao')
plt.ylabel('Tuổi thọ')
```

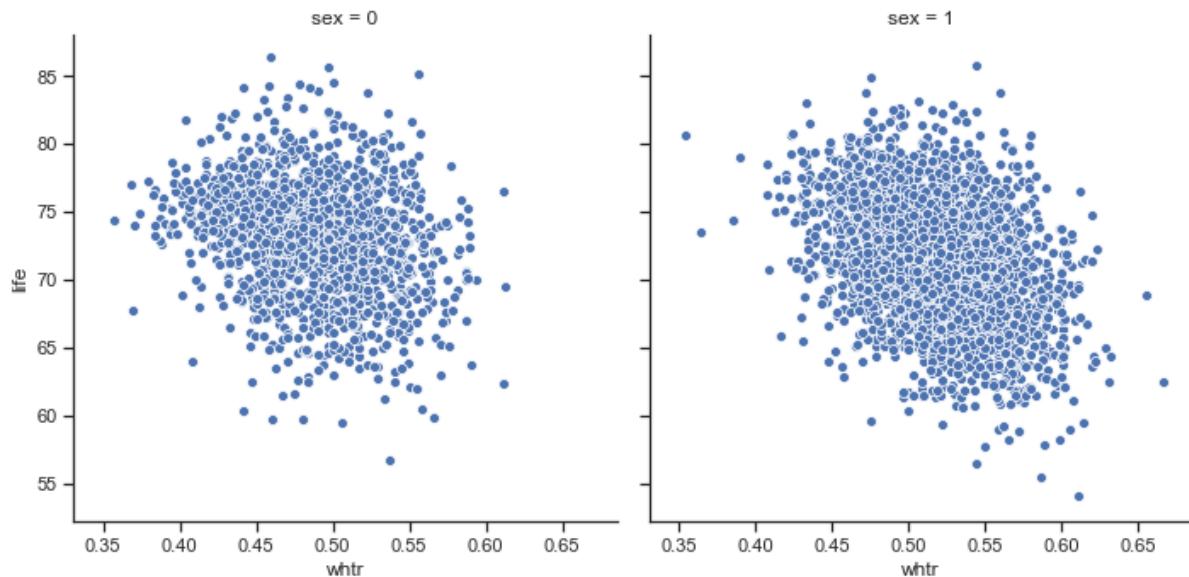


Biểu đồ tương quan có phân nhóm

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

df =
pd.read_csv('https://thachln.github.io/datasets/sample_health_vn.csv')
df.head()
df['whtr'] = df['waist'] / df['height']
plt.figure().suptitle("Mối quan hệ 'tuổi thọ' và 'tỉ lệ vòng eo với chiều cao'")
sns.set(style="ticks")
sns.relplot(x="whtr", y="life", col="sex", data=df)
plt.xlabel('Tỉ số vòng eo và chiều cao')
plt.ylabel('Tuổi thọ')
```

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

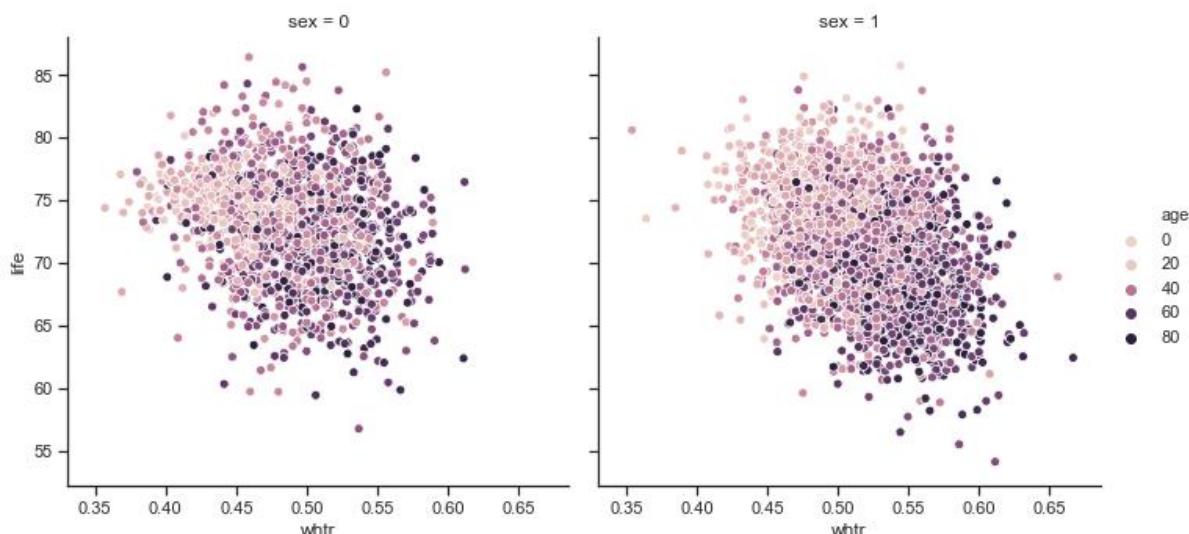


Trang trí thêm một chút bằng cách vẽ các điểm ảnh có tông màu khác nhau theo tuổi (age). Tuổi này là tuổi tại thời điểm bắt đầu nghiên cứu, khác với biến life là tuổi thọ.

Lệnh bên dưới dùng thêm tham số hue = 'age':

```
import seaborn as sns
import pandas as pd

df =
pd.read_csv('https://thachln.github.io/datasets/sample_health_vn.csv')
df.head()
df['whtr'] = df['waist'] / df['height']
sns.set(style="ticks")
sns.relplot(x="whtr", y="life", col="sex", hue='age', data=df)
```



Một vài điểm quan sát từ dữ liệu trên:

- Các chấm đen có thiên hướng lệch dần về bên phải. Tức là càng nhiều tuổi thì tỷ lệ whtr các lớn. Nói nôm na càng lớn tuổi thì bụng càng phệ.
- Xu hướng “Vòng eo càng lớn thì vòng đùi càng ngắn” trong nam giới rõ ràng hơn là nữ giới.

Biểu đồ tương quan đa biến

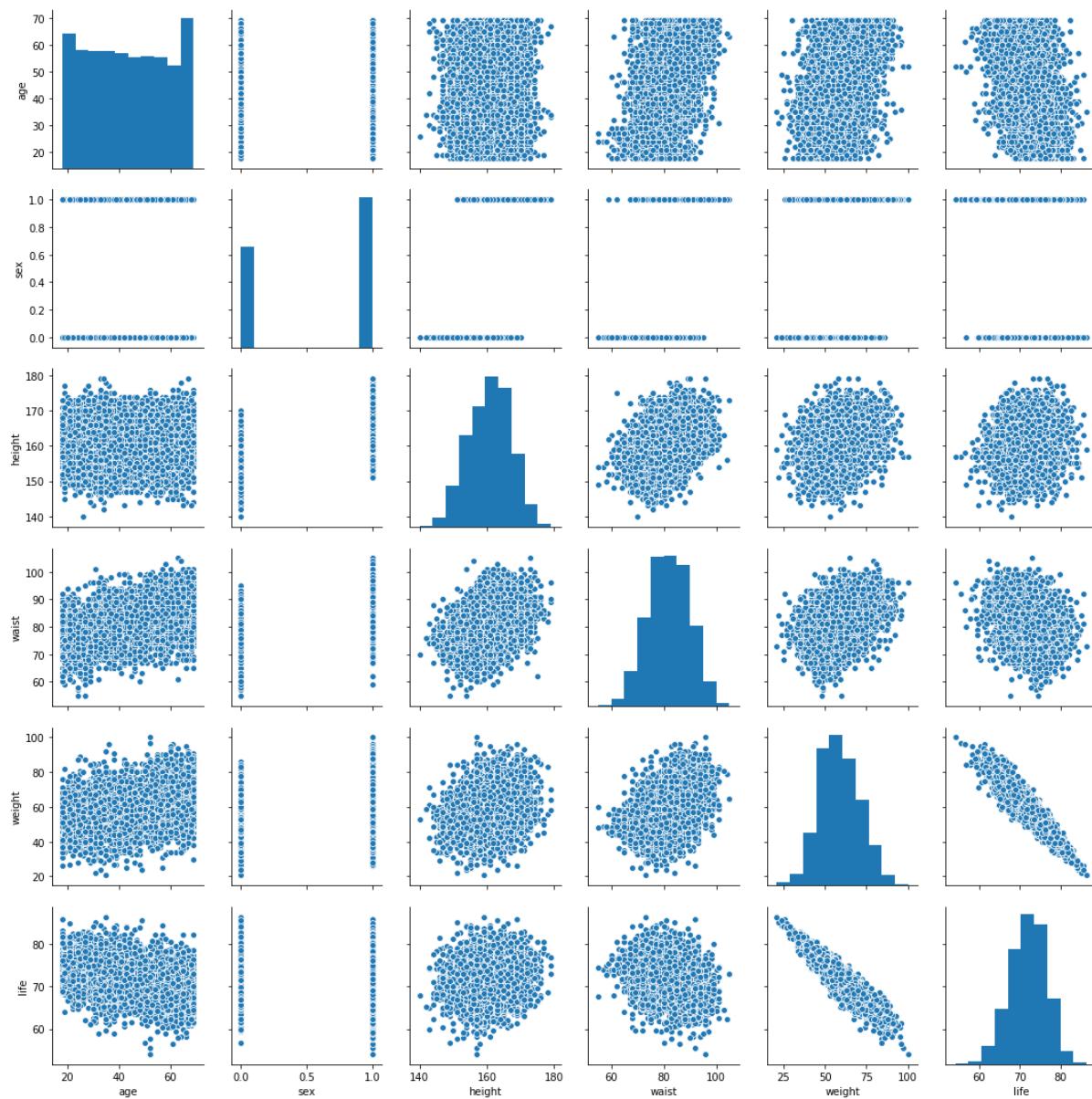
Để vẽ biểu đồ trình bày mối tương quan của các biến trong data frame thì thư viện Seaborn cung cấp hàm **pairplot**.

Đoạn code bên dưới xóa bỏ các cột id, risk, hit sau khi đọc từ dữ liệu mẫu.

```
import pandas as pd
import seaborn as sns

df =
pd.read_csv('https://thachln.github.io/datasets/sample_health_vn.csv')
# Lệnh bên dưới loại bỏ cột id
df = df.drop( axis = '1', columns = ['id', 'risk', 'hit'])
sns.pairplot(df)
```

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

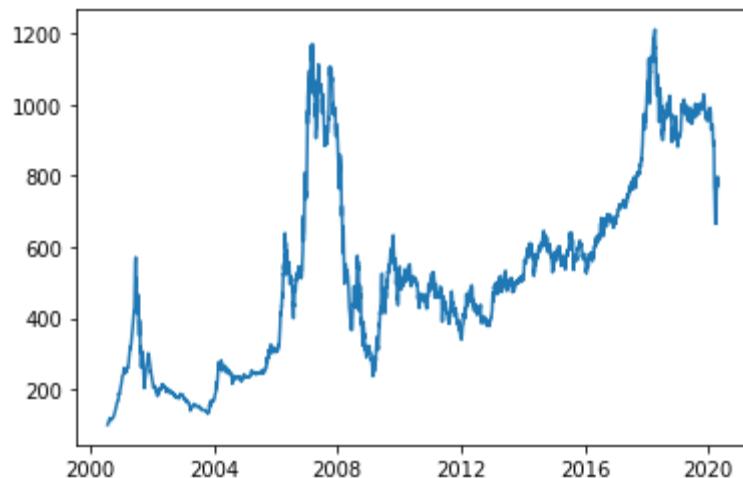


Biểu đồ dữ liệu theo thời gian

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

df =
pd.read_csv('https://thachln.github.io/datasets/vnindex_20200424.txt')
df.head()

pd.to_datetime('13000101', format='%YYYY%mm%dd', errors='coerce')
df['date'] = pd.to_datetime(df['<DTYYYYMMDD>'], format='%Y%m%d')
plt.plot(df['date'], df['<High>'])
```



Vẽ biểu đồ với ggplot2

Trong R thì ggplot2 rất ưa được sử dụng vì triết lý của nó rất hay “Xem biểu đồ như là một bức tranh hoàn thiện” nên có nhiều lệnh giúp người chủ động trang trí bức tranh. Còn trong Python thì cũng có nhiều thư viện để vẽ biểu đồ như tôi đã giới thiệu cho bạn làm quen ở trên. Có một câu hỏi đặt ra trong phần này là “Nếu tôi đã quen với ggplot2 trong R thì có cách nào để tôi vận dụng vào Python hay không?”.

Đầu tiên là cần cài đặt thư viện rpy2 vào môi trường Python bằng các copy & paste lệnh sau vào dấu nhắc của Anaconda:

```
conda install -c r rpy2
```

Để chuẩn bị cho ví dụ thì cài đặt tiếp module tzlocal

```
pip install tzlocal
```

```
(base) C:\Users\ThachLN>conda install -c r rpy2
```

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

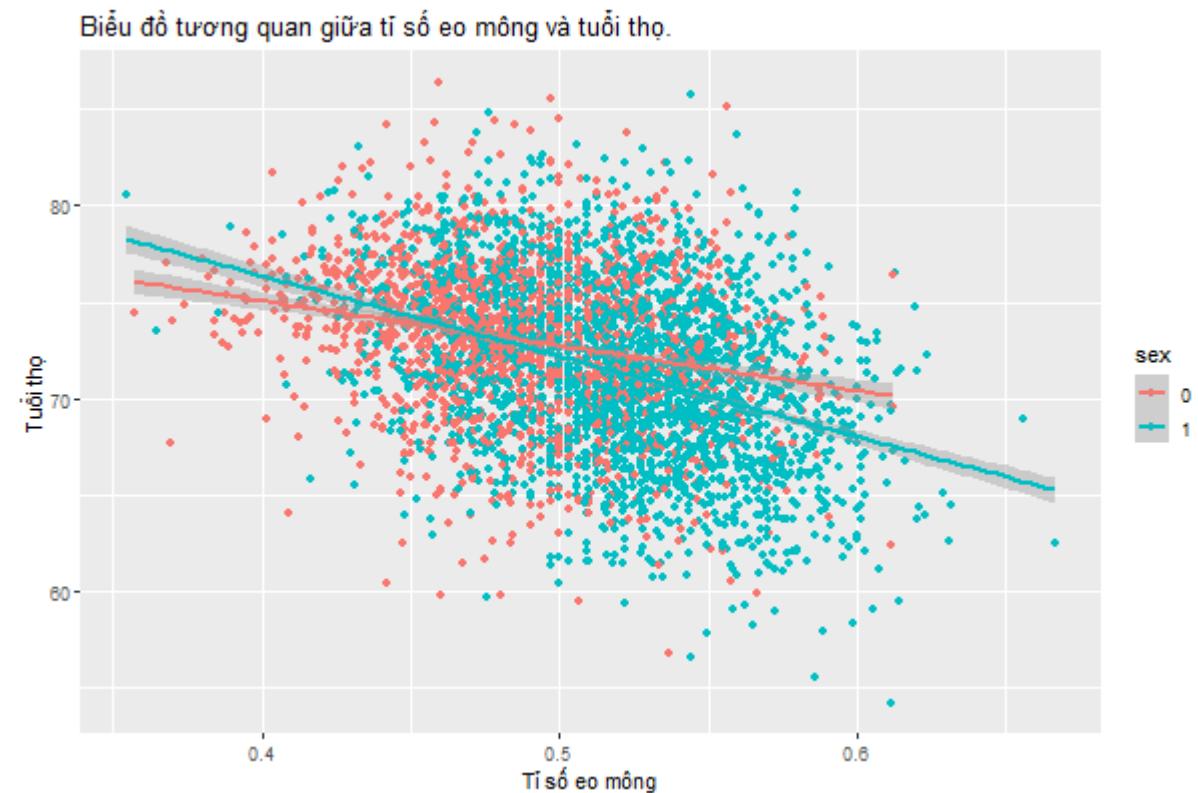
Tiếp theo cài đặt gói ggplot2:

```
from rpy2.robjects.packages import importr  
utils = importr('utils')  
utils.install_packages('ggplot2')
```

```
from rpy2.robjects.packages import importr  
  
import pandas as pd  
from rpy2.robjects import pandas2ri  
import rpy2.robjects.lib.ggplot2 as ggplot2  
  
import uuid  
from IPython.core.display import Image, display  
  
grdevices = importr('grDevices')  
  
df =  
pd.read_csv('https://thachln.github.io/datasets/sample_health_vn.csv')  
df['whtr'] = df['waist'] / df['height']  
df['sex'] = df['sex'].astype('category')  
df.head()  
  
pandas2ri.activate()  
r_dataframe = pandas2ri.py2ri(df)  
  
pp = ggplot2.ggplot(r_dataframe) + \  
    ggplot2.aes_string(x='whtr', y='life', col = 'sex') + \  
    ggplot2.geom_point() + \  
    ggplot2.geom_smooth(ggplot2.aes_string(group = 'sex'), method =  
'lm') + \  
    ggplot2.labs(x='Tỉ số eo mông', y='Tuổi thọ', title='Biểu đồ  
tương quan giữa tỉ số eo mông và tuổi thọ.')  
  
fn = '{uuid}.png'.format(uuid = uuid.uuid4())  
grdevices.png(fn, width = 600, height = 400)  
pp.plot()  
grdevices.dev_off()
```

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

```
image = Image(filename=fn)  
display(image)
```



Bài 8: Nguyên tắc soạn biểu đồ

Từ Bài 8 đến Bài 11 bạn đã làm quen với các loại biểu đồ phổ biến. Trong đó đã giúp các bạn sử dụng các lệnh R và cả Python để trình bày được các biểu đồ. Nói chung các vấn đề kỹ thuật thì chắc các bạn cũng không gặp khó khăn gì. Bài này sẽ chia sẻ với các bạn vài hướng dẫn để giúp các bạn soạn biểu đồ có chất lượng cao.

Phản ảnh trung thực dữ liệu

Với góc nhìn của người phân tích dữ liệu thì chúng ta đi tìm thêm thông tin từ dữ liệu, thông thường là các con số. Nếu hình ảnh hóa tối đa thì có cơ may phát hiện thêm nhiều thông tin quý giá như:

- Hình dung được bức tranh tổng thể của dữ liệu. Ví dụ sử dụng các biểu đồ histogram có thể mường tượng nhanh phân bố của dữ liệu. Plot hoặc boxplot có thể phát hiện nhanh các dữ liệu outlier (ngoại vi hay ngoại biên)
- Phát hiện được qui luật của dữ liệu.
- Phát hiện được mối tương quan của dữ liệu.

Vì thế cần phải trình bày biểu đồ sao cho rõ ràng, áp dụng đúng loại biểu đồ để phản ánh dữ liệu.

Tiết kiệm mực in

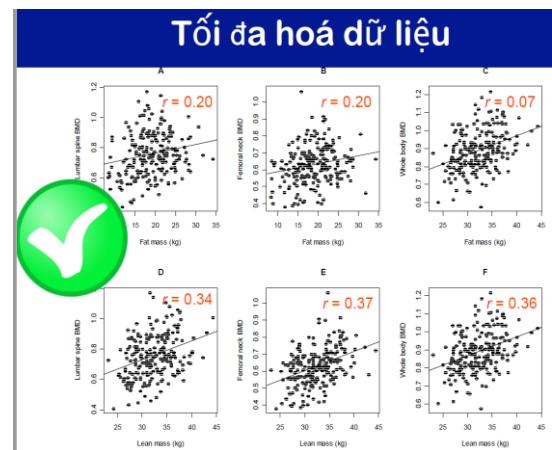
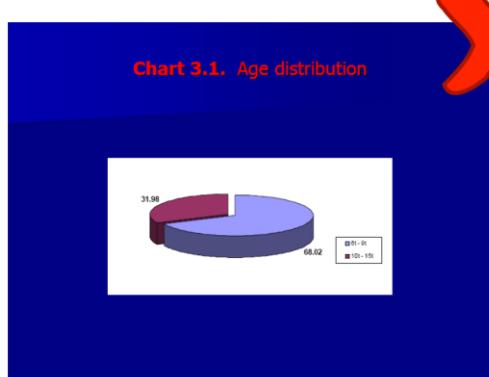
Có thể đây đơn thuần là bài toán kinh tế. Tức là trên một trang giấy nếu bạn dùng ít mực in mà cung cấp được nhiều thông tin nhất thì sẽ có lợi nhất. Vì vậy chúng ta cần chọn biểu đồ, màu sắc sao cho cung cấp cho người đọc nhiều thông tin nhất có thể. Nếu in ra giấy thì tiết kiệm mực nhất có thể.

Bạn thử hình dung trong công ty, sếp của bạn yêu cầu bạn (Data Analyst hoặc Data Scientist) in tài liệu mà bạn đã phân tích về tình hình kinh doanh của công ty cho các nhà đầu tư xem trong lúc sếp trình chiếu. Nếu nhà đầu tư cầm bảng tài liệu như các hình minh họa bên trái dưới đây thì họ nghĩ gì? Có thể câu đầu tiên họ hỏi là sao các bạn không biết tiết kiệm mực in vậy? Câu này tôi tưởng tượng ra thôi, có thể lầm chứ? Nếu họ không hỏi như vậy thì chắc họ cũng đánh giá tài liệu không chuyên nghiệp?

Một số minh họa NÊN và KHÔNG NÊN

KHÔNG NÊN

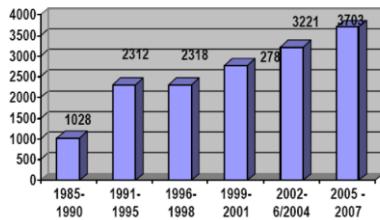
NÊN



III. Results and discussion

1. Epidemiology (1/1985 - 12/2007)

1.1. Number of admitted patients



Bài 9: Giới thiệu Matplotlib

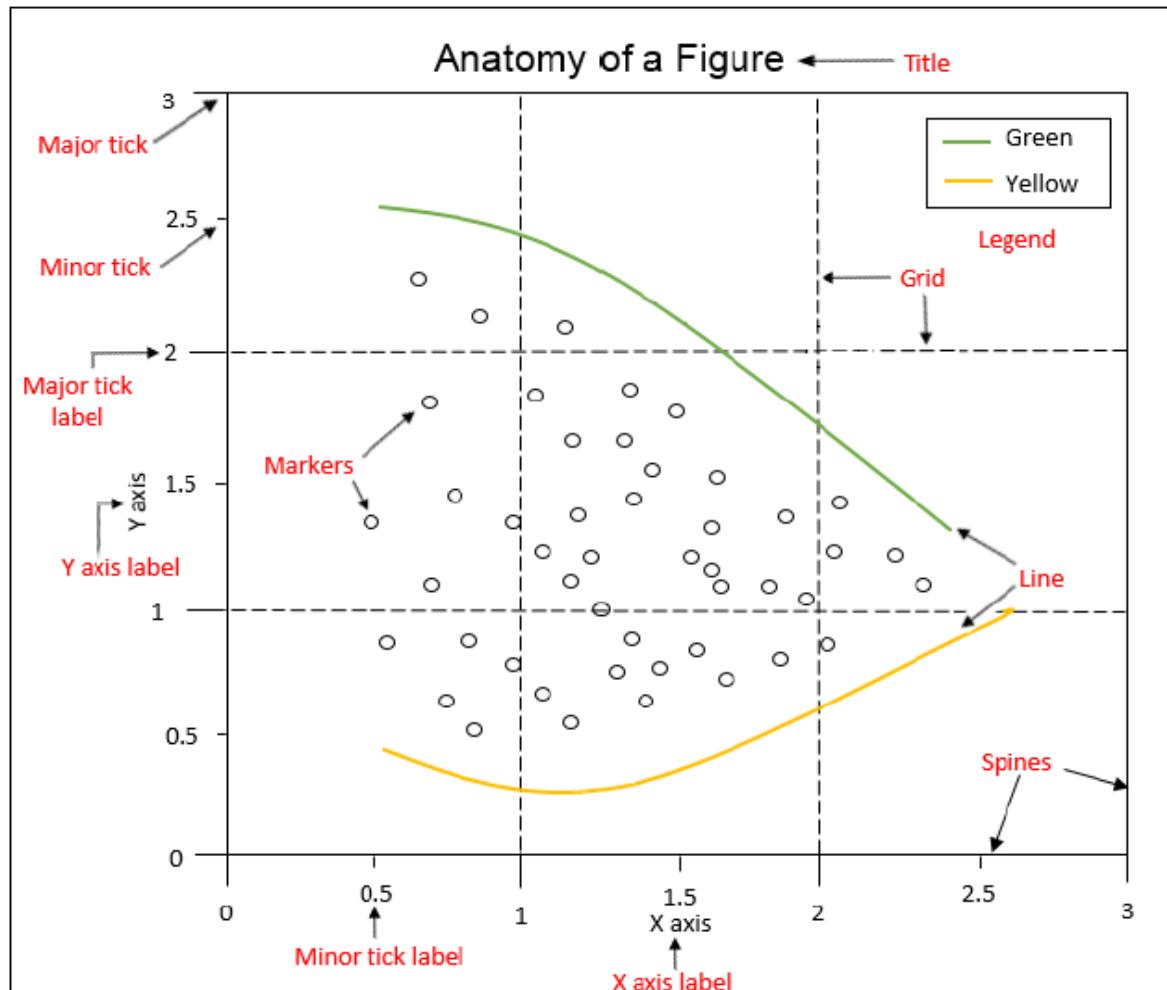
Trong ngày đầu tiên bạn đã làm quen với các loại biểu đồ và đã biết thư viện matplotlib. Tuy nhiên bài này sẽ dành riêng để tìm hiểu về thư viện vẽ biểu đồ phổ biến trong Python này. Matplotlib rất phổ biến trong giới data science (khoa học dữ liệu) và machine learning (máy học). Matplotlib được John 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 tick), **nhãn giá trị đơn vị chính** (Major tick label); **kí hiệu chia đơn vị phụ** (Minor tick), **nhãn giá trị đơn vị phụ** (Minor tick label)



Sub module pyplot của Matplotlib

Module `pyplot` sẽ giúp chúng ta vẽ các biểu đồ mà không cần tốn nhiều thời gian 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))

# Thiếp 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 tấm 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))

# Thiếp 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"	+ (filled)	plus (filled)
"*"	★	star
"h"	⬡	hexagon1
"H"	⬢	hexagon2
"+"	+	plus
"x"	✗	x
"X"	✗ (filled)	x (filled)
"D"	◆ (filled)	diamond
"d"	◆ (thin)	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 stylesline. 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:

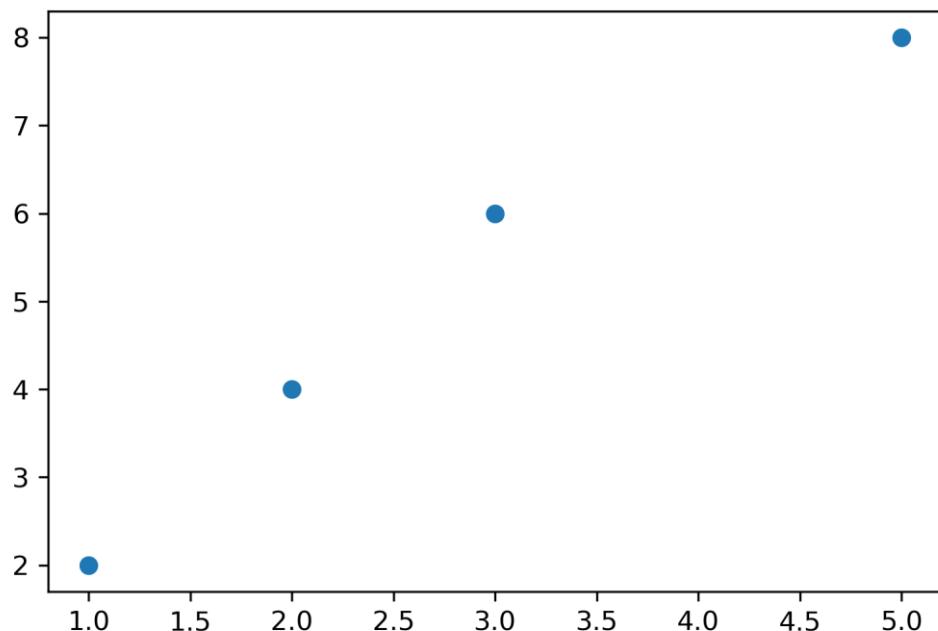
```
plt.plot(x, y, color='blue', marker='o', linestyle='dashed')
```

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
```

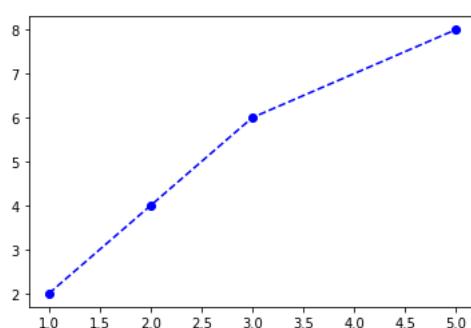
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
# 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')
```



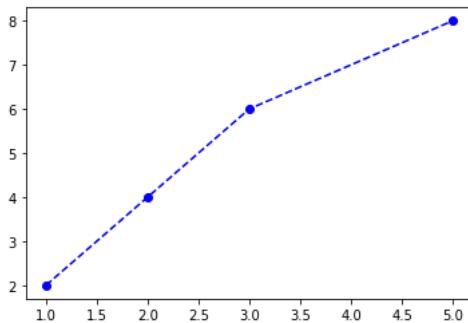
Thử plot với format khác nhau để tự khám phá ý nghĩa các kí hiệu:

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

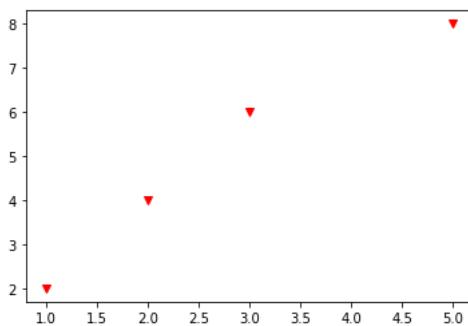


```
plt.plot(x, y, color='blue', marker='o', linestyle='dashed')
```

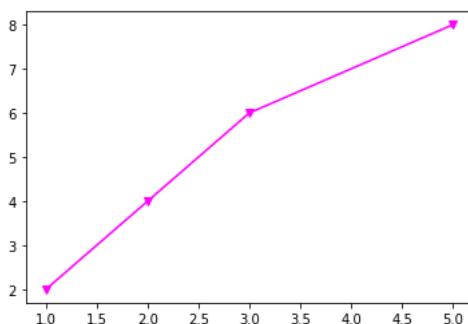
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



```
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

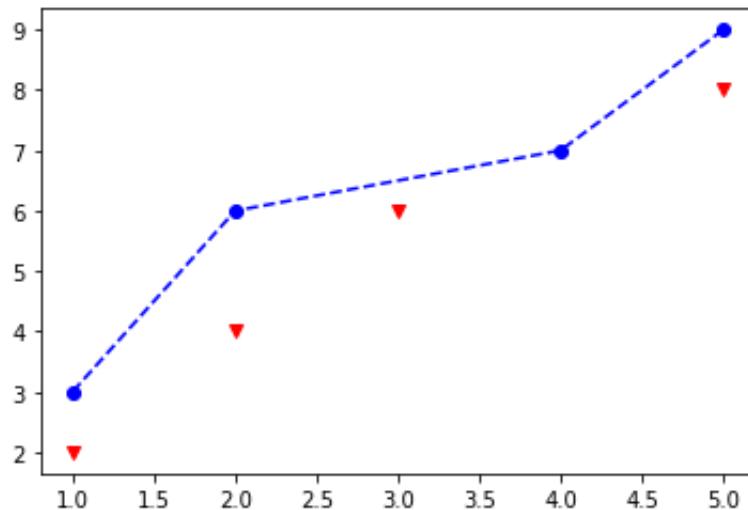
Có thể mở rộng gồm 2 bộ tham số:

```
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--')
```

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



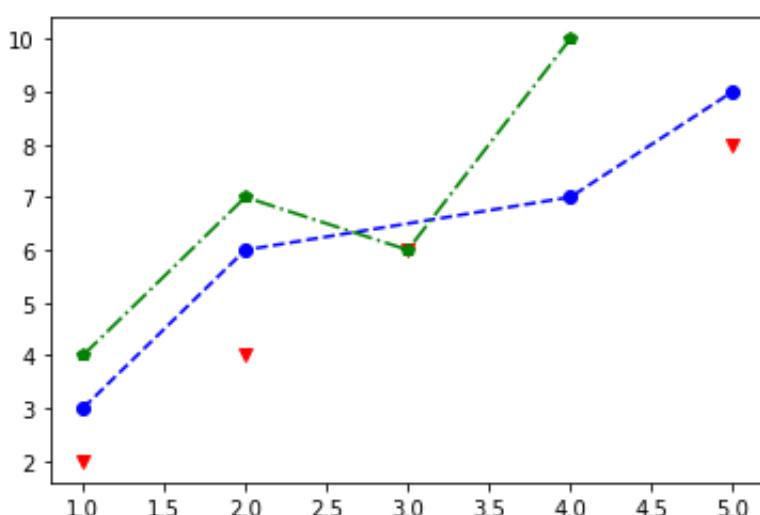
Với 3 bộ tham số:

```
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:

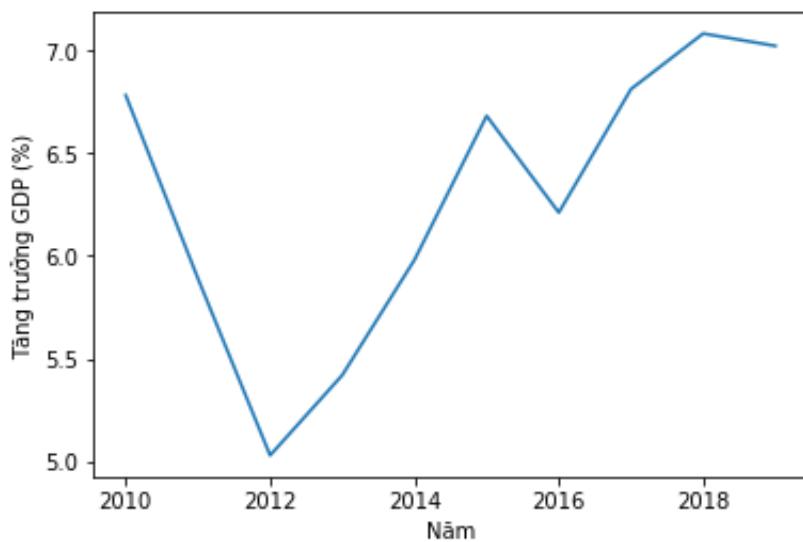
```
import pandas as pd
import matplotlib.pyplot as plt

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 dùng: vẽ với trực x là giá trị cột tên '**year**', trực 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

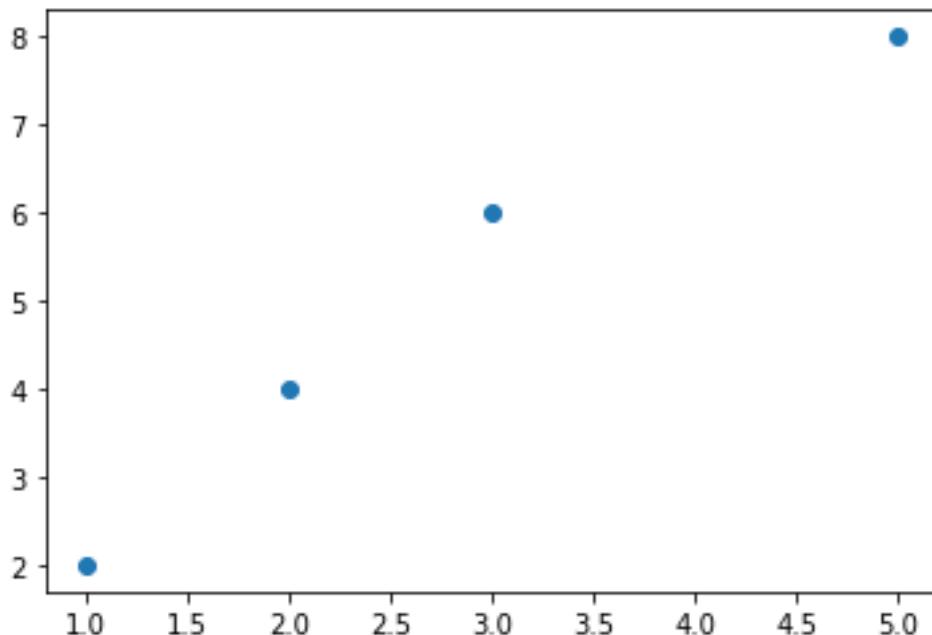
Lệnh Python để gọi tạo Figure và Axe từ hàm `.subplots()` như sau:

```
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')
```

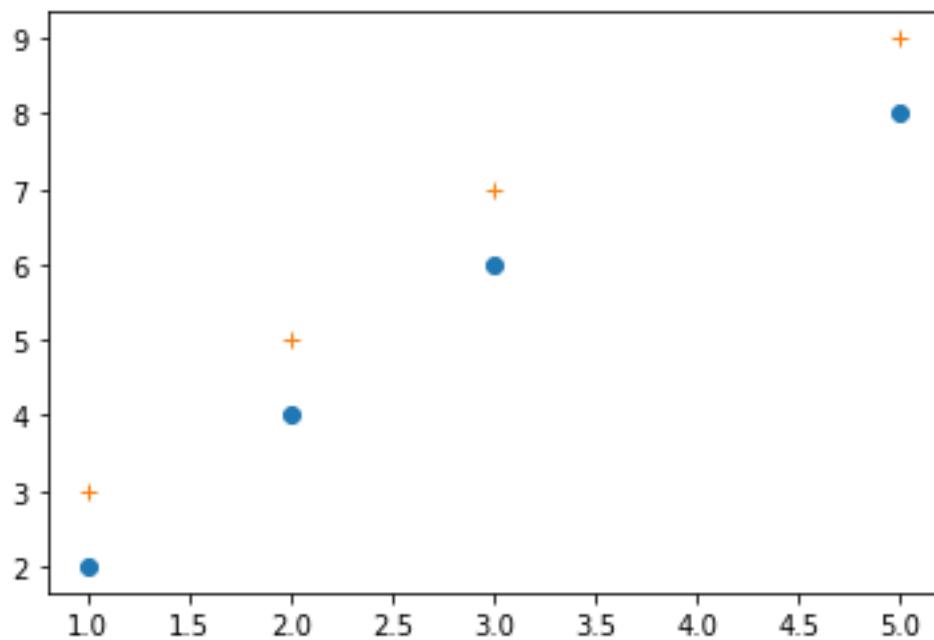


Dùng Axe để vẽ nhiều biểu đồ

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 (x_1, y_1). 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')
```

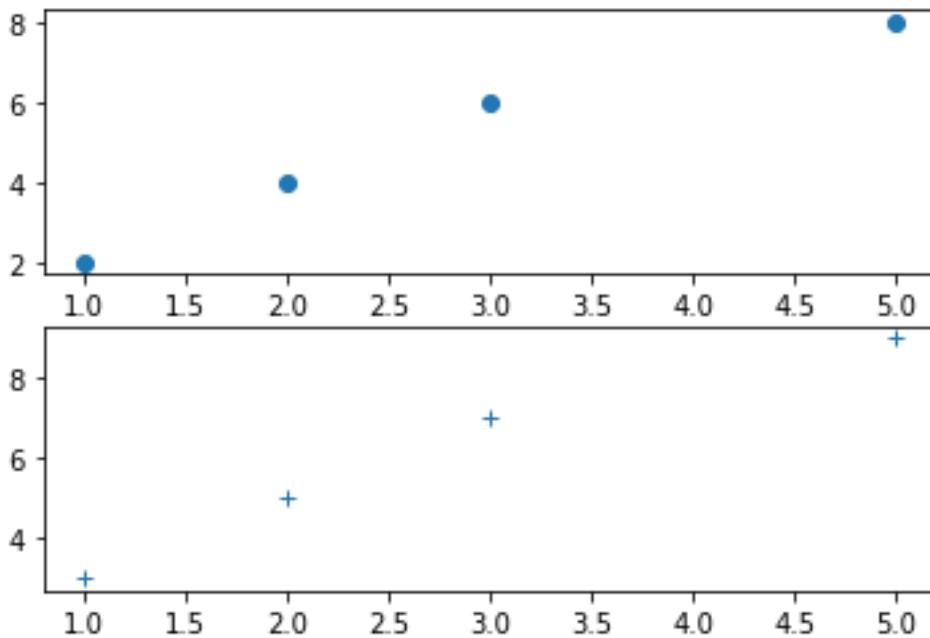
```
x1 = x  
y1 = [3, 5, 7, 9]  
ax.plot(x1, y1, '+')
```



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')
```

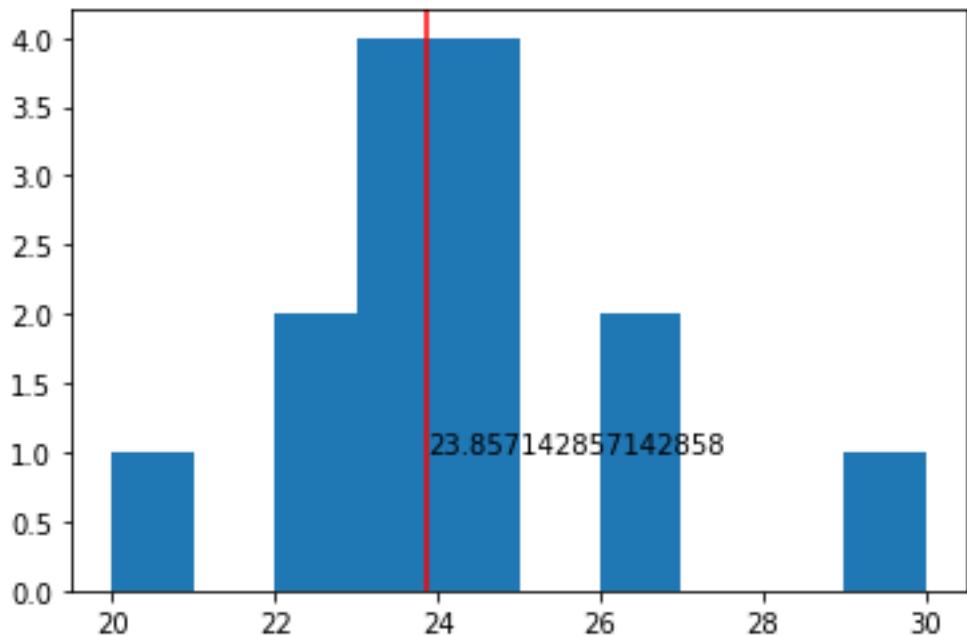
Vẽ thêm text tại vị trí x, y

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())
```



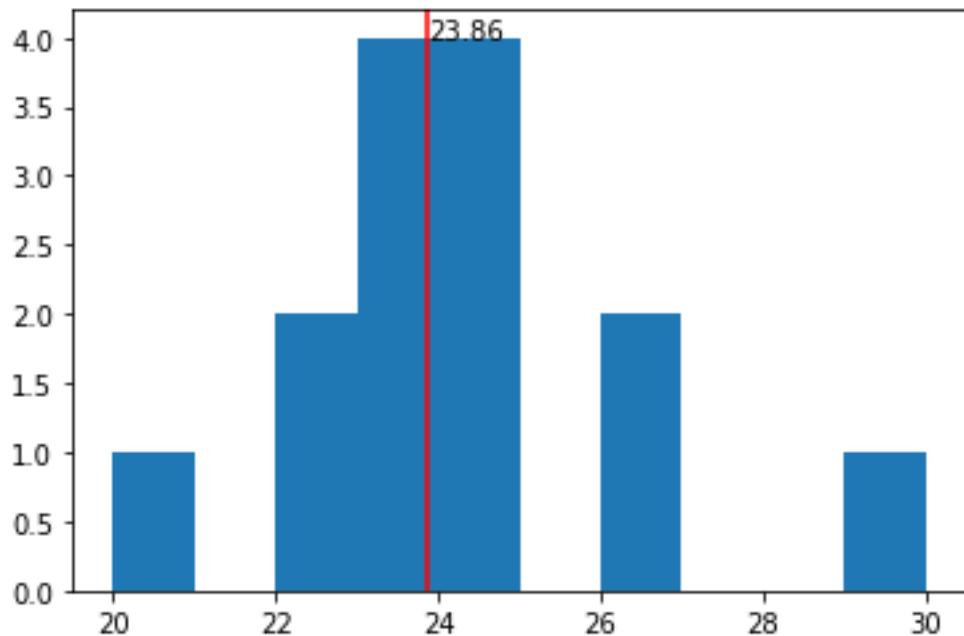
Cải tiến một chút cho biểu đồ:

```
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:

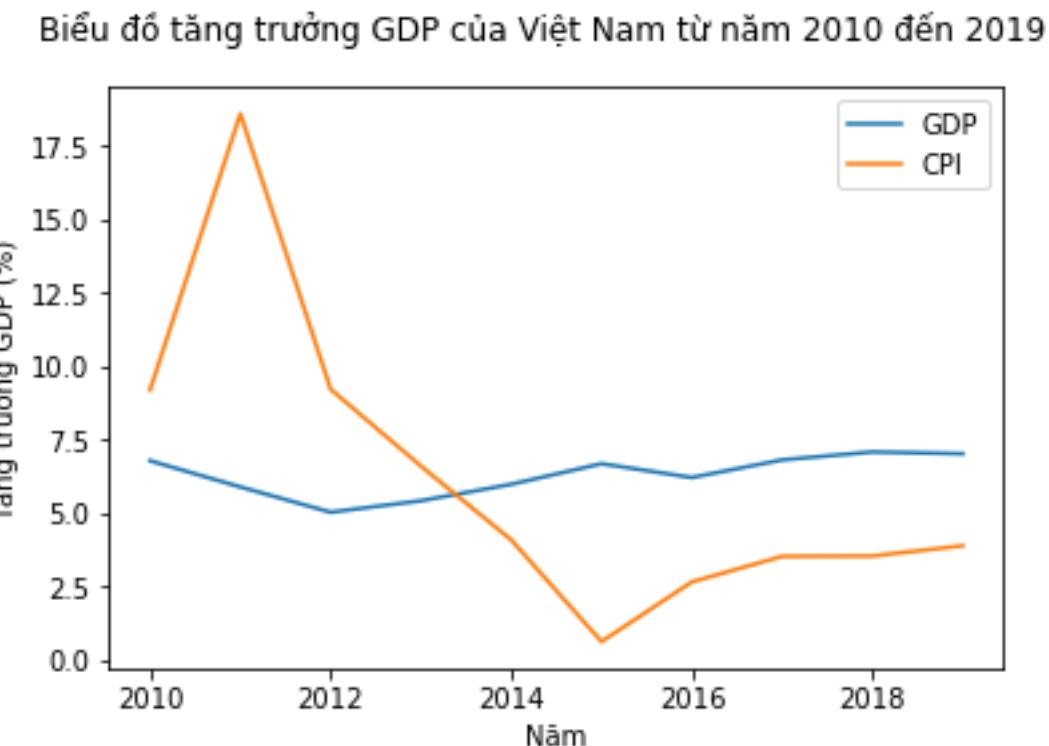
```
import pandas as pd
import matplotlib.pyplot as plt

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()
```



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()

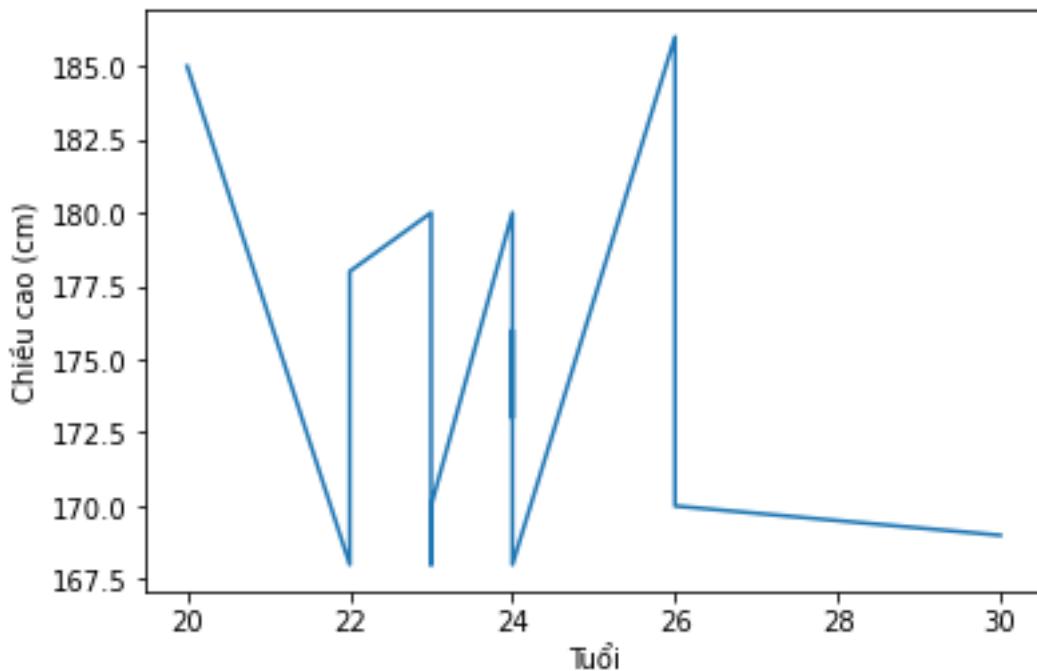
# Sắp xếp dữ liệu theo tuổi
df = df.sort_values(by = 'age')

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')
```

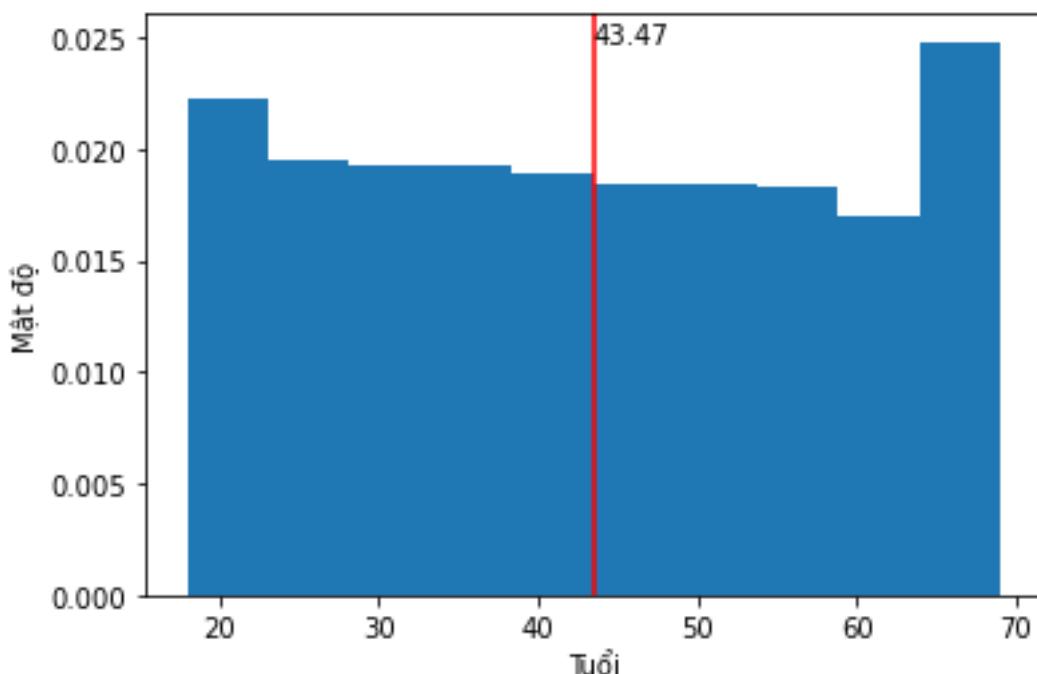
Xem phân bố của 1 biến với hàm .hist

```
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()
```

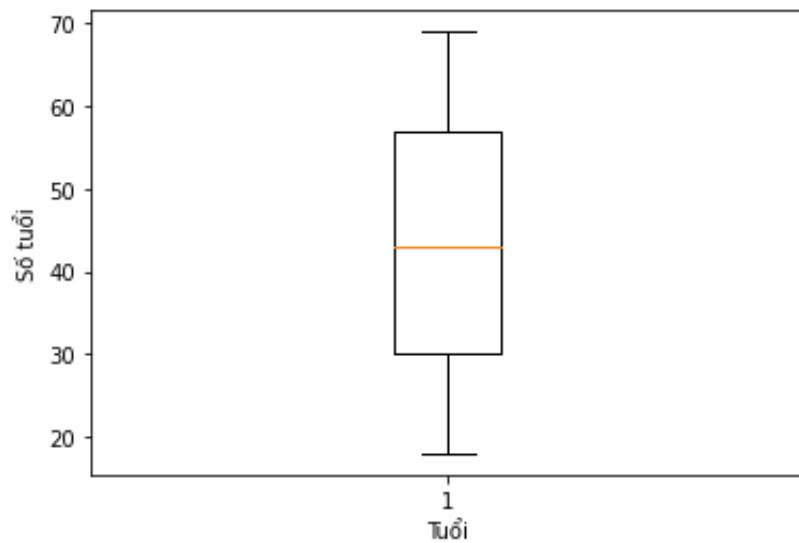


Xem dữ liệu tuổi với boxplot

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

plt.show()
```

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



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`

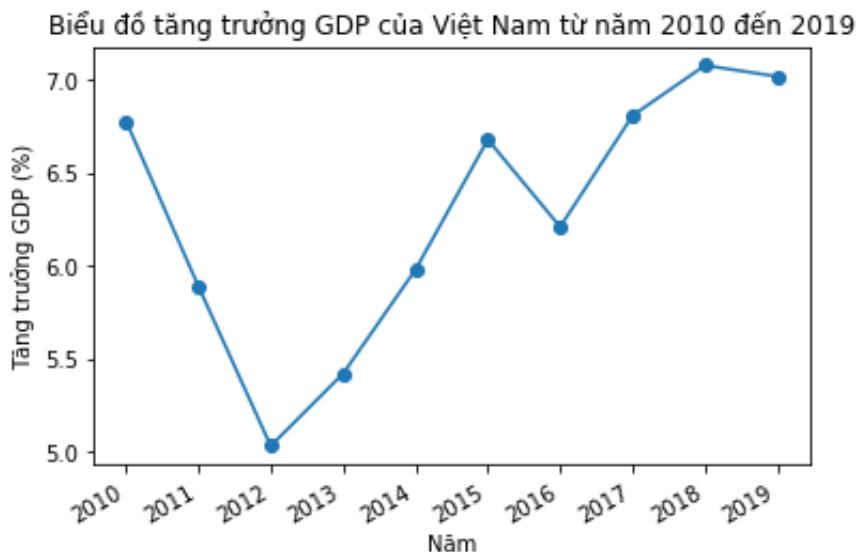
```
import pandas as pd
import matplotlib.pyplot as plt

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)

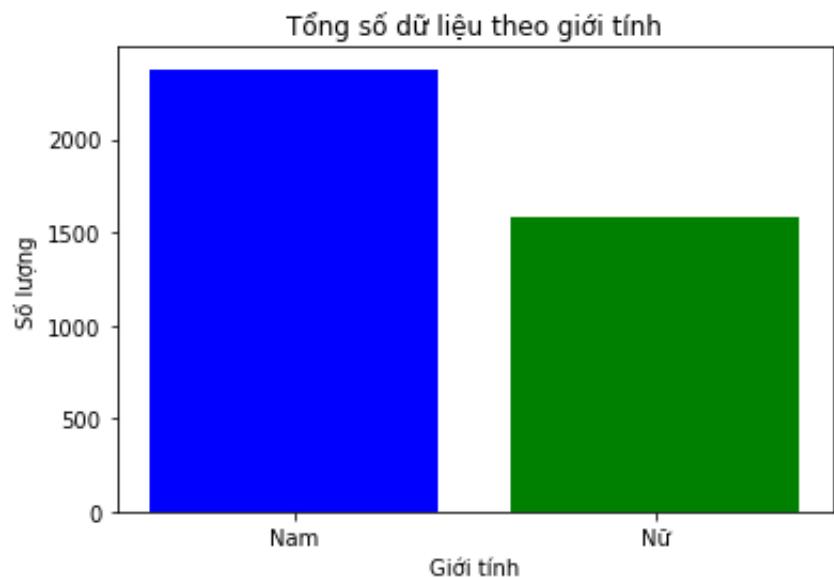
import matplotlib.pyplot as plt

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'])

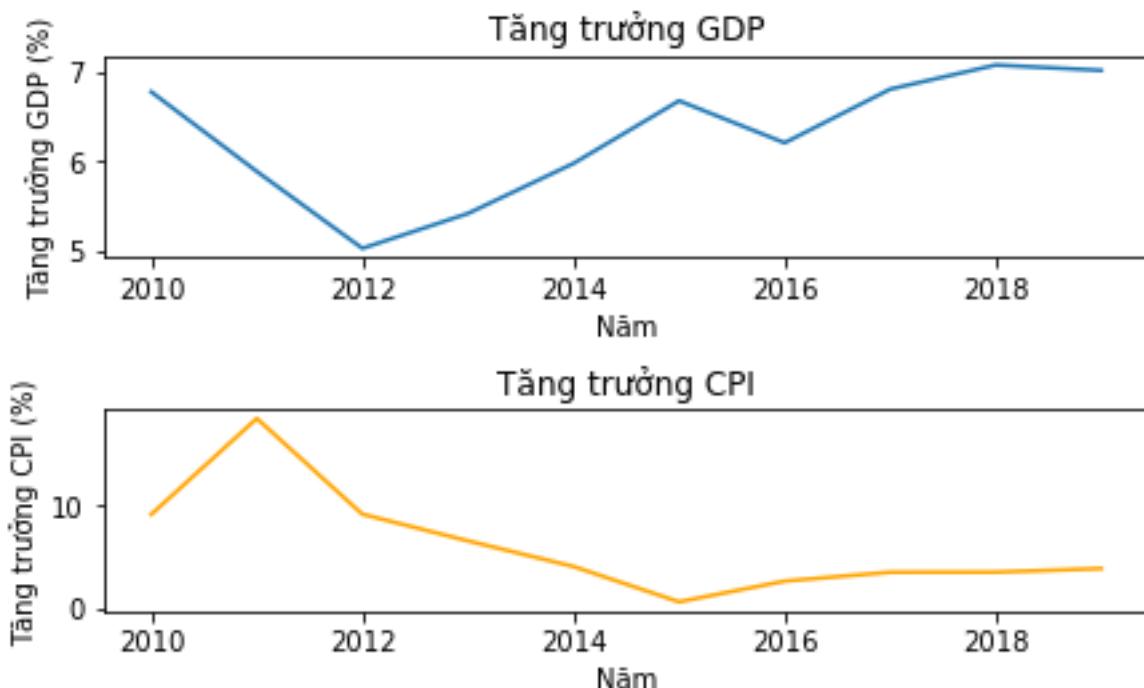
ax2.set_title('Tăng trưởng CPI')
ax2.set_xlabel('Năm')
ax2.set_ylabel('Tăng trưởng CPI (%)')
ax2.plot(df['year'], df['cpi'], color='orange')
```

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

```
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})

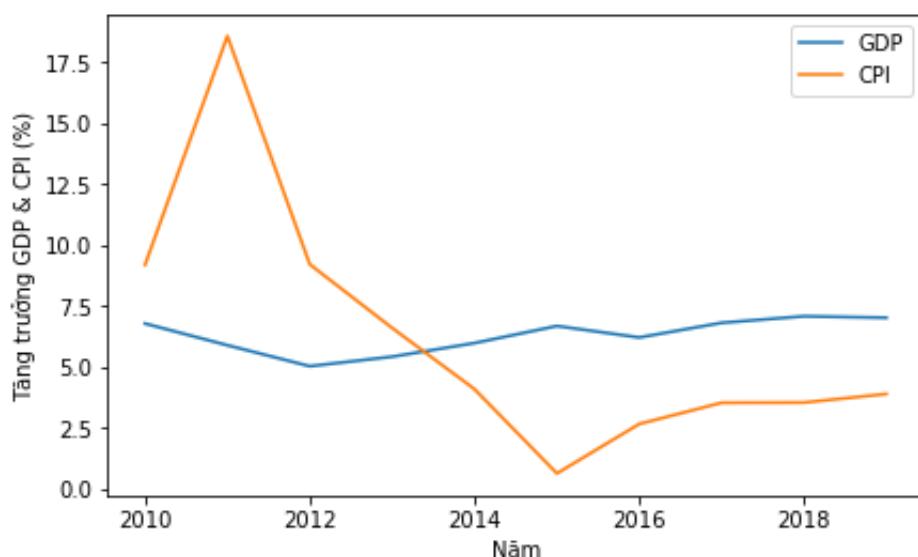
fig, ax = plt.subplots(1, constrained_layout=True)
ax.plot(df['year'], df['gdp'], label = 'GDP')
ax.plot(df['year'], df['cpi'], label = 'CPI')
```

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

```
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()
```

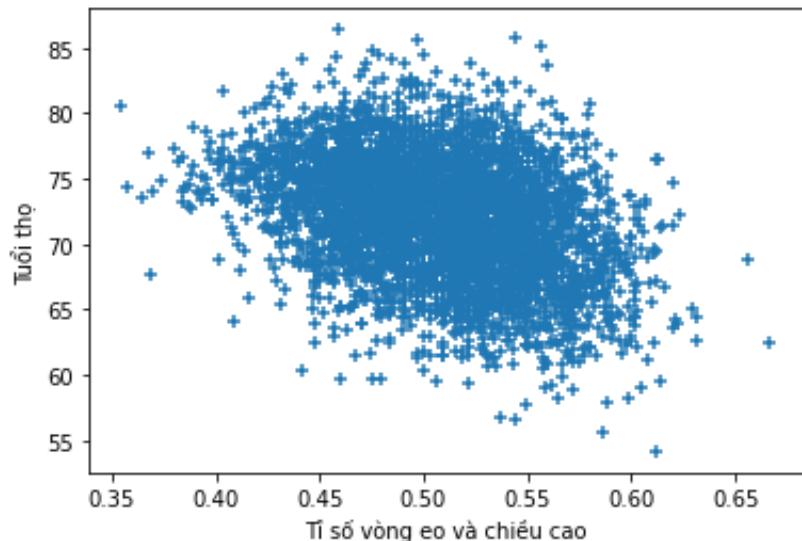
Biểu đồ tăng trưởng GDP và CPI của Việt Nam từ năm 2010 đến 2019



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('Ti 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

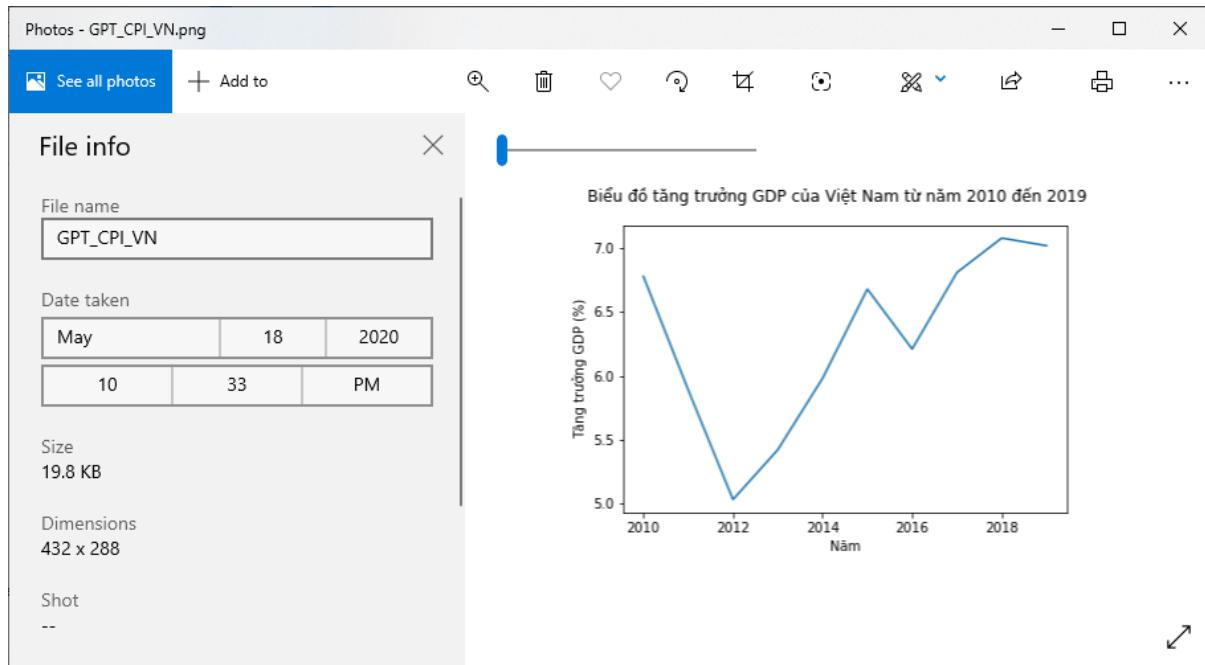
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019]

# 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")
```

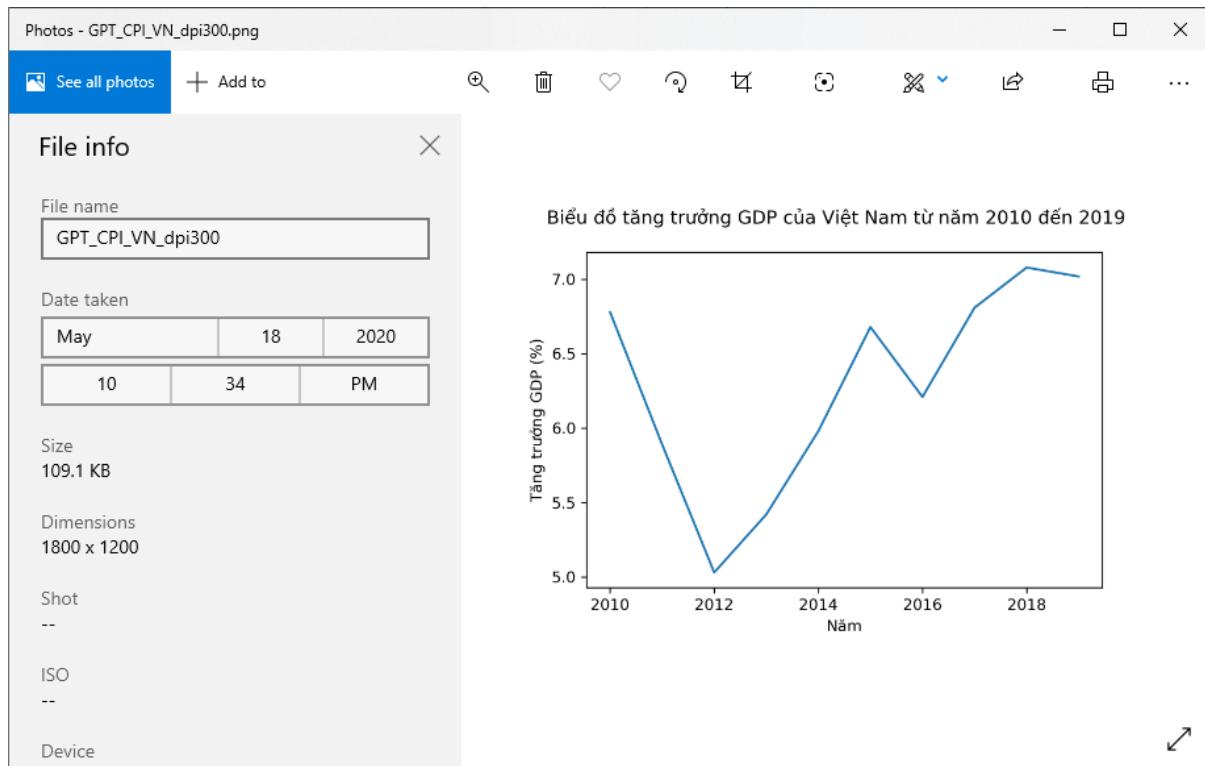
Kết quả:

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)
```

Bài 10: Giới thiệu Bokeh

Giới thiệu

Bokeh là một thư viện trong Python giúp trực quan hóa dữ liệu có tương tác (interactive data visualization). Đặc biệt Bokeh giúp bạn mang các biểu đồ lên web một cách dễ dàng.

Vài khái niệm cơ bản

Document	Bokeh xem biểu đồ chỉ là một phần của trang web cần trình bày. Vì thế Bokeh tổ chức biểu đồ và nhiều thông tin khác nữa để xuất kết quả, thông dụng nhất thì kết quả là một website, dưới dạng Bokeh Document (tài liệu)
Application	Đây là phần mã nguồn đảm nhận phần nghiệp vụ vẽ biểu đồ và xử lý các sự kiện tương tác (như người dùng thay đổi điều kiện dữ liệu, dữ liệu nguồn có thay đổi, v.v...) để kết ra Bokeh Document.
Server	Gồm 3 thành phần Bokeh Document và Application ở trên để cung cấp dịch vụ hiển thị biểu đồ cho người dùng qua môi trường mạng.
Figure	Là đối tượng được ví như là một bức tranh mà bạn là họa sĩ sẽ vẽ biểu đồ và trang trí lên bức tranh.

Đọc thêm tài liệu tại:

https://docs.bokeh.org/en/latest/docs/user_guide/concepts.html
<https://docs.bokeh.org/en/latest/docs/reference/server.html>

Cài đặt

Tài liệu này dùng bokeh phiên bản 2.2.3 với lệnh cài đặt như sau:

```
pip install bokeh==2.2.3
```

Sử dụng bokeh plotting đơn giản

Quay lại dữ liệu mẫu GDP của Việt Nam, sử dụng thư viện bokeh để vẽ biểu đồ line như sau:

```
from bokeh.plotting import figure, output_file, show  
  
output_file("output.html")  
  
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
```

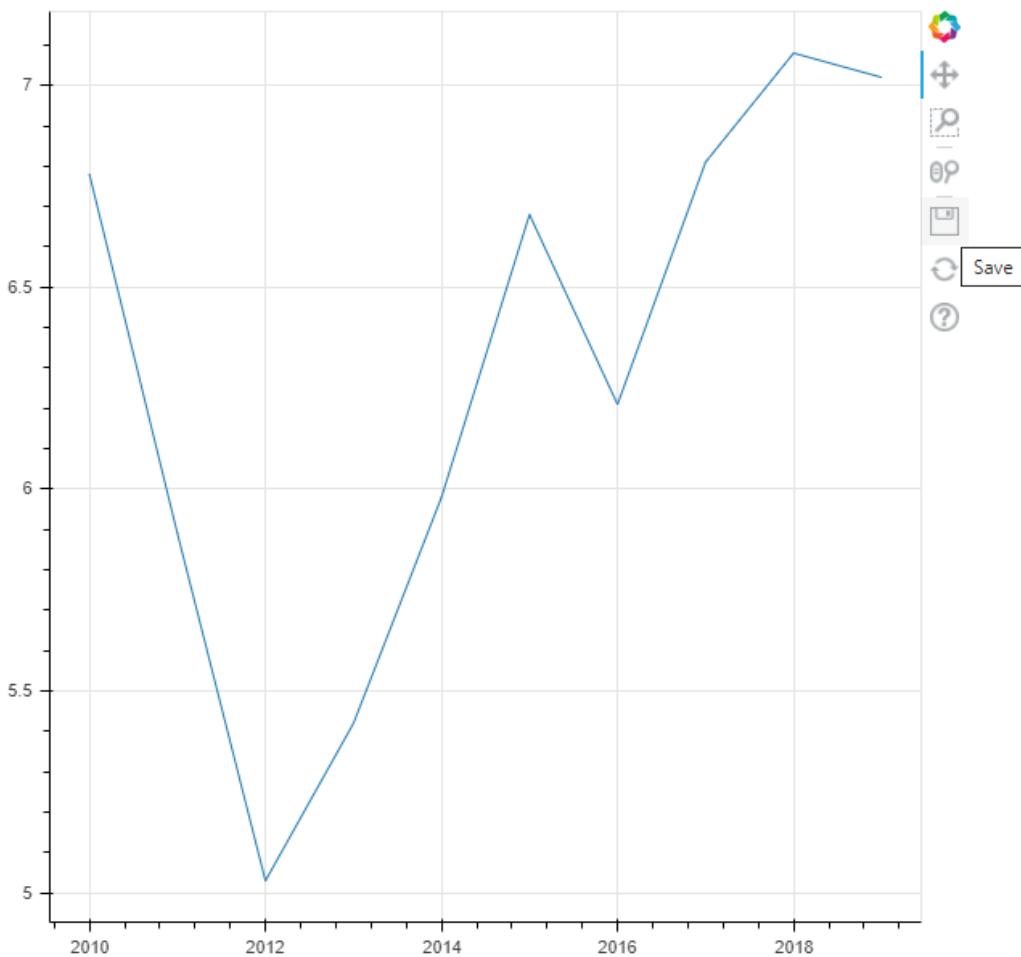
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
year = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019]

p = figure()
p.line(x=year, y=gdp)

show(p)
```

Kết quả:



Bạn để ý thấy bên phải có thanh công cụ. Trong đó có biểu tượng cái đĩa (disk) cho phép bạn lưu (Save) cái hình của biểu đồ thành file ảnh.

Phân tích mã nguồn

Dòng đầu tiên để import các hàm sau trong thư viện bokeh.plotting:

- Hàm `output_file(file_path)` để thiết lập biểu đồ sẽ được ghi thành file trong đường dẫn `file_path`. Trường hợp không có lệnh này thì bokeh sẽ tự tạo ra file tạm.

- Hàm `figure()` để tạo ra đối tượng để vẽ biểu đồ. Hãy tưởng tượng giống như một họa sĩ sẽ căng khung vải để vẽ bức tranh. Đối tượng được lưu trong biến `p` (viết tắt của `picture` hoặc `plot`).

Nếu không truyền tham số gì hết thì kích thước của biểu đồ mặc định là `600 x 600 pixel`. Muốn chỉ định rõ kích thước thì dùng thêm tham số `plot_width` và `plot_height` như sau:

```
figure(plot_width=300, plot_height=300)
```

- Hàm `p.line(x= array, y = array)` để vẽ một đường thẳng lên `picture` với thông số trục `x` và trục `y` như chỉ định.
- Hàm `show(p)` sẽ gọi trình duyệt mở file trong hàm `output_file(file_path)` ở trên để bạn xem.

Đưa biểu đồ lên Web

Sử dụng Bokeh Server bên ngoài

Thay vì trong ví dụ trên dùng hàm `output_file(...)` để lưu biểu đồ ra file HTML, chúng ta thử sửa dùng hàm `curdoc()` để thêm “bức tranh” vào tài liệu.

Hãy lưu đoạn code sau vào file. Ví dụ file: D:\MyPython\bokeh\bokeh_gpd.py

```
from bokeh.plotting import figure, curdoc

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]

p = figure()
p.line(x=year, y=gdp)

curdoc().add_root(p)
```

Đứng trong thư mục D:\MyPython\bokeh của cửa sổ lệnh có Python và đã cài thư viện Bokeh thực hiện lệnh sau:

```
bokeh serve --show bokeh_gpd.py
```

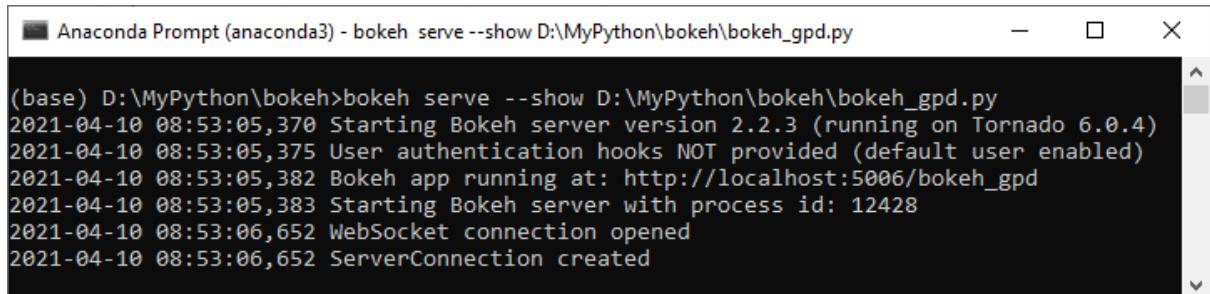
Hoặc có thể gõ đường dẫn đầy đủ của file .py như sau:

```
bokeh serve --show D:\MyPython\bokeh\bokeh_gpd.py
```

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

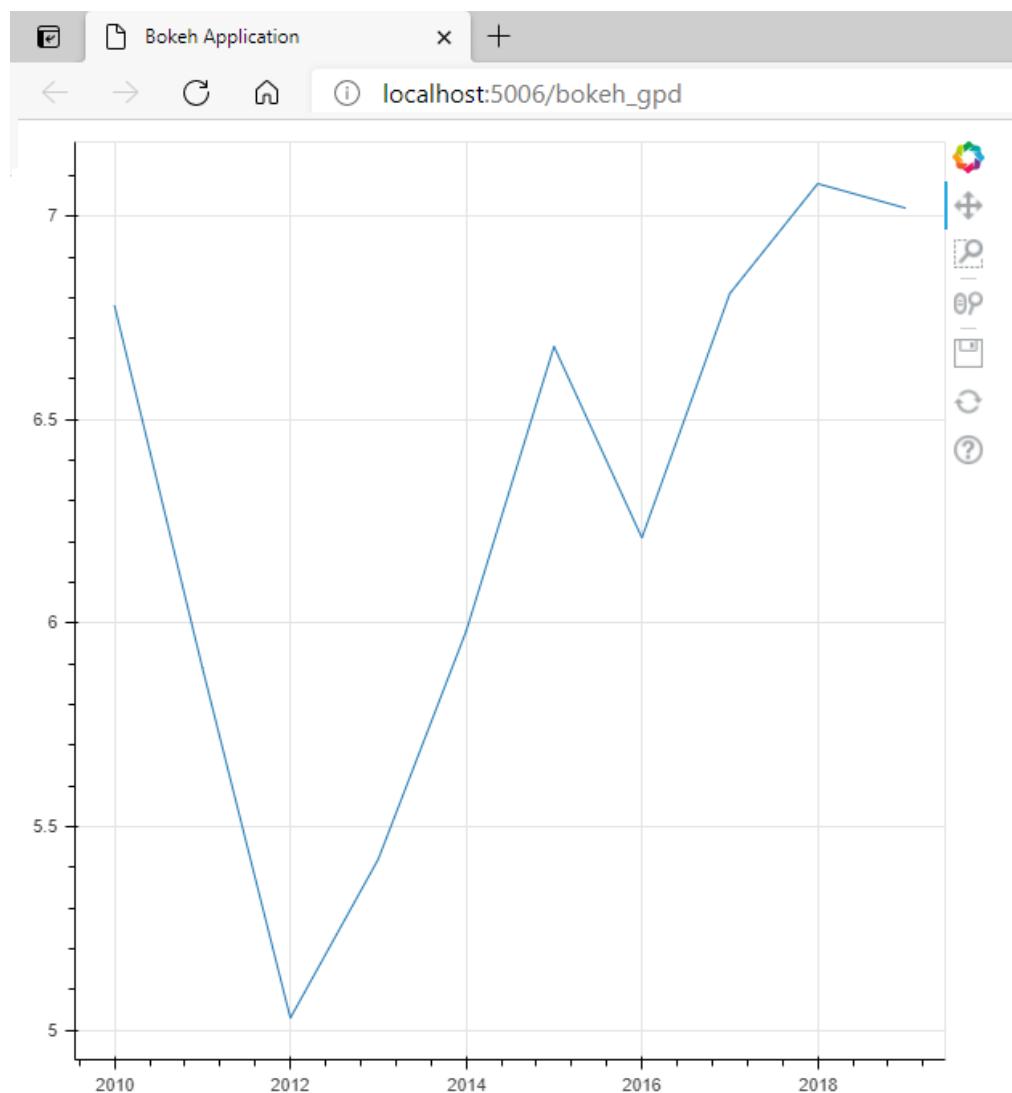
Cửa sổ lệnh sẽ hiển thị các kết quả sau cho thấy là lệnh bokeh sẽ mở một Bokeh server với port 5006 và phục vụ biểu đồ ở trên tại đường link:

http://localhost:5006/bokeh_gpd



```
Anaconda Prompt (anaconda3) - bokeh serve --show D:\MyPython\bokeh\bokeh_gpd.py
(base) D:\MyPython\bokeh>bokeh serve --show D:\MyPython\bokeh\bokeh_gpd.py
2021-04-10 08:53:05,370 Starting Bokeh server version 2.2.3 (running on Tornado 6.0.4)
2021-04-10 08:53:05,375 User authentication hooks NOT provided (default user enabled)
2021-04-10 08:53:05,382 Bokeh app running at: http://localhost:5006/bokeh_gpd
2021-04-10 08:53:05,383 Starting Bokeh server with process id: 12428
2021-04-10 08:53:06,652 WebSocket connection opened
2021-04-10 08:53:06,652 ServerConnection created
```

Đồng thời trình duyệt trên máy bạn sẽ tự mở được link ở trên với kết quả như sau:

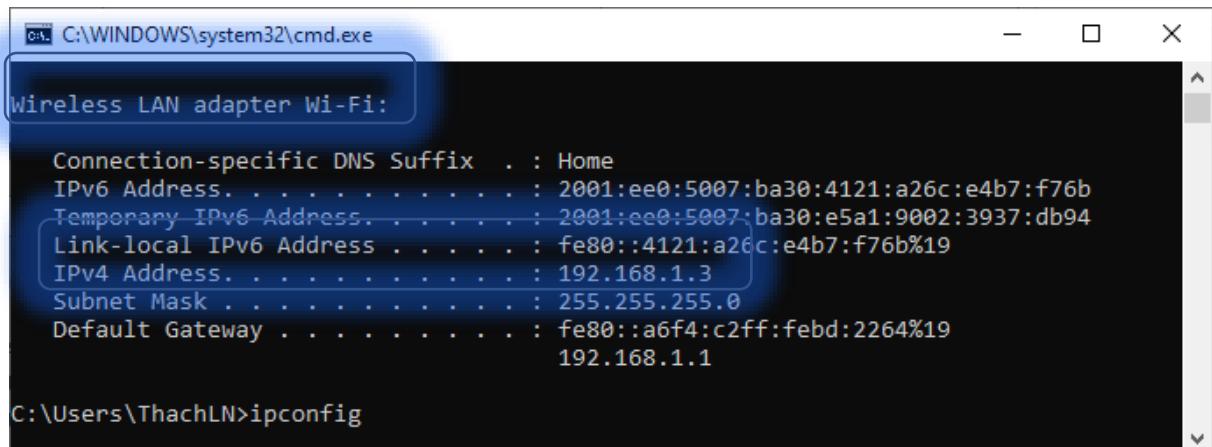


Nếu bạn muốn bạn bè, đồng nghiệp của mình đang dùng chung Network (ví dụ đang cùng vào chung một WiFi) thì chạy lại lệnh bokeh với tham số cho phép truy cập mạng như sau:

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

bokeh serve --show bokeh_gpd.py --allow-websocket-origin=192.168.1.3:5006

Tham số --allow-websocket-origin=192.168.1.3:5006 có nghĩa là cho phép máy tính của bạn có địa chỉ IP là 192.168.1.3 chạy chương trình bokeh với cổng 5006. Để kiểm tra chính xác IP mạng của máy bạn thì dùng lệnh ipconfig. Sau đó tìm đoạn có chữ “Wireless LAN adapter Wi-Fi” và mục IPv4 Address. Ví dụ:



```
C:\WINDOWS\system32\cmd.exe
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : Home
IPv6 Address . . . . . : 2001:ee0:5007:ba30:4121:a26c:e4b7:f76b
Temporary IPv6 Address . . . . . : 2001:ee0:5007:ba30:e5a1:9002:3937:db94
Link-local IPv6 Address . . . . . : fe80::4121:a26c:e4b7:f76b%19
IPv4 Address . . . . . : 192.168.1.3
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::a6f4:c2ff:feb:2264%19
                           192.168.1.1

C:\Users\ThachLN>ipconfig
```

Lúc này có thể mời bạn bè, đồng nghiệp vào địa chỉ sau để xem thành quả của bạn:

http://192.168.1.3:5006/bokeh_gpd

Để tắt bokeh server thì đứng trong cửa sổ lệnh đang chạy, nhấn phím Ctrl+C và đợi vài giây, bokeh server sẽ ngưng với thông báo sau:

Interrupted, shutting down

Như vậy bạn đã biết cách vẽ một biểu đồ đơn giản với thư viện Bokeh và Bokeh server để có thể truy cập biểu đồ từ mạng. Tuy nhiên theo như cách chạy lệnh Bokeh ở trên thì có chút bất tiện là mỗi lần code xong phải chạy lại lệnh. Phần tiếp theo sẽ hướng dẫn bạn khởi chạy server bằng code Python luôn.

Sử dụng Bokeh Server bên trong

Chạy thử ví dụ sau:

```
from bokeh.plotting import figure
from bokeh.server.server import Server
from bokeh.application import Application
from bokeh.application.handlers.function import FunctionHandler

def make_document(doc):
    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]
```

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

```
p = figure()
p.line(x=year, y=gdp)

doc.add_root(p)

doc.title = "Sample chart"
doc.add_root(p)

apps = {'/': Application(FunctionHandler(make_document))}

server = Server(apps, port=5000)
server.start()
```

Sau đó dùng trình duyệt mở địa chỉ sau bạn sẽ thấy kết quả tương tự như phần trước:

<http://localhost:5000>

Cấu trúc mã nguồn:

Để ý 2 lệnh cuối cùng:

```
server = Server(apps, port=5000)
server.start()
```

để khởi động Bokeh server với port 5000. Trong đó Bokeh server cần một tham số là Application (xem lại phần khái niệm cơ bản ở trên). Tên đầy đủ của Application ở đây là “Bokeh Server Tornado application”. Tra cứu tài liệu tại:

<https://docs.bokeh.org/en/latest/docs/reference/server/tornado.html#bokeh.server.tornado.BokehTornado>

bạn sẽ thấy cách khai báo Application này là dạng Dictionary như:

{ '/': applications }

Cụ thể trong ví dụ trên thì khai báo Application như sau:

```
apps = {'/': Application(FunctionHandler(make_document))}
```

Dấu ‘/’ có nghĩa là khi người dùng gõ đường dẫn URL trên trình duyệt như <http://localhost:5000/> (nếu không có dấu xuyệt / cuối thì coi như là có) thì Máy tính

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

(trong ngữ cảnh này gọi là máy chủ nhân, chính là cái máy tính mà bạn đang thực hành) sẽ nhận và xử lý URL gồm 2 phần:

- Phần “`http://localhost:5000`”: sẽ được máy có tên là localhost (tức là máy của bạn đang thực hành) có một phần mềm đang chạy ở port 5000 (trong trường này chính là Bokeh Server trong ví dụ) nhận và xử lý yêu cầu.
- Phần “`/`” (dấu duyệt phải ở cuối): gọi là yêu cầu (request). Dấu / gọi là root. Có thể URL đầy đủ không có dấu xuyệt thì root là rỗng. Khi gặp yêu cầu root này thì đối tượng **applications** sẽ xử lý. Cụ thể applications trong ví dụ trên là:

```
Application(FunctionHandler(make_document))
```

Đây là lệnh khởi tạo đối tượng Application⁶ với tham số là đối tượng Handler⁷. Cụ thể là:

```
FunctionHandler(make_document)
```

Lệnh này sử dụng đối tượng FuncHandler để chỉ định hàm lo xử lý việc tạo ra bokeh document. Bạn xem cách khai báo hàm `make_document` với tham số là doc như sau:

```
def make_document(doc):  
    p = figure()  
    ...  
    doc.add_root(p)  
  
    ...  
    doc.add_root(p)
```

Hàm `make_document` cập nhật tham số `doc` bằng cách gọi hàm `add_root(p)` với `p` là đối tượng Figure.

Stop server

Thực hiện lệnh Python sau:

⁶ Tra cứu tại:

<https://docs.bokeh.org/en/latest/docs/reference/application/application.html#bokeh.application.Application>

⁷

<https://docs.bokeh.org/en/latest/docs/reference/application/handlers/handler.html#bokeh.application.handlers.handler.Handler>

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

```
server.stop()
```

Mã nguồn đầy đủ có sẵn lệnh server.stop() được chú thích.

```
from bokeh.plotting import figure
from bokeh.server.server import Server
from bokeh.application import Application
from bokeh.application.handlers.function import FunctionHandler

def make_document(doc):
    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]

    p = figure()
    p.line(x=year, y=gdp)

    doc.add_root(p)

    doc.title = "Sample chart"
    doc.add_root(p)

apps = {'/': Application(FunctionHandler(make_document))}

print('Start bokeh server with port 5000...')
server = Server(apps, port=5000)
server.start()
print('Try the application with URL: http://localhost:5000')

# server.stop()
```

Nếu bạn chạy đoạn code trên 2 lần thì sẽ bị báo lỗi như sau:

```
OSErrror: [WinError 10048] only one usage of each socket address (protocol/network address/port) is normally permitted
```

Lý do là cái port 5000 đã được sử dụng và bokeh server đang chạy.

Muốn tắt server thì bôi lệnh đoạn `server.stop()` và thực thi nó. Nếu dùng IDE Spyder thì bôi và nhấn F9.

Bài 11: Khai phá Bokeh

Bài trước đã giúp bạn làm quen với vài khái niệm cốt lõi và biết tổng thể về việc dùng bokeh để vẽ một biểu đồ. Đồng thời bạn cũng biết xuất biểu đồ ra file HTML hoặc đưa lên web. Bài này sẽ giúp bạn sử dụng Bokeh để phục vụ cho các nhu cầu về phân tích dữ liệu, đặc biệt khai phá sức mạnh của Bokeh để trực quan hóa dữ liệu nhằm cảm nhận dữ liệu nhanh và sâu sắc hơn. Đây là tiền đề để giúp một người phân tích dữ liệu có thể đưa ra các ý tưởng mới, hoặc tí ra khám phá được thêm các thông tin ẩn đằng sau các con số.

Xem phân bố dữ liệu với biểu đồ histogram

Xem dữ liệu mẫu:

```
import pandas as pd
df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', delimiter=';')
df_desc = df.describe()
```

df_desc - DataFrame											
Index	age	duration	campaign	pdays	previous	emp.var.rate	ins.price.in	cons.conf.id	euribor3m	r.employe	
count	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188
mean	40.0241	258.285	2.56759	962.475	0.172963	0.0818855	93.5757	-40.5026	3.62129	5167.04	
std	10.4212	259.279	2.77001	186.911	0.494901	1.57096	0.57884	4.6282	1.73445	72.2515	
min	17	0	1	0	0	-3.4	92.201	-50.8	0.634	4963.6	
25%	32	102	1	999	0	-1.8	93.075	-42.7	1.344	5099.1	
50%	38	180	2	999	0	1.1	93.749	-41.8	4.857	5191	
75%	47	319	3	999	0	1.4	93.994	-36.4	4.961	5228.1	
max	98	4918	56	999	7	1.4	94.767	-26.9	5.045	5228.1	

Format Resize Background color Column min/max Save and Close Close

Xem cột mô tả cột age bạn cũng hình dung phần nào độ tuổi của người tham gia chiến dịch marketing của ngân hàng trên. Ví dụ, vài thông tin có thể mô tả ra đây:

- Tổng cộng có 41.188 người tham gia.
- Độ tuổi trung bình là: 40
- Tuổi thấp nhất là 17; tuổi cao nhất là 98
- 25% số người đó độ tuổi khoảng 32
- 50% số người có độ tuổi khoảng 38
- 75% số người có độ tuổi khoảng 47

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

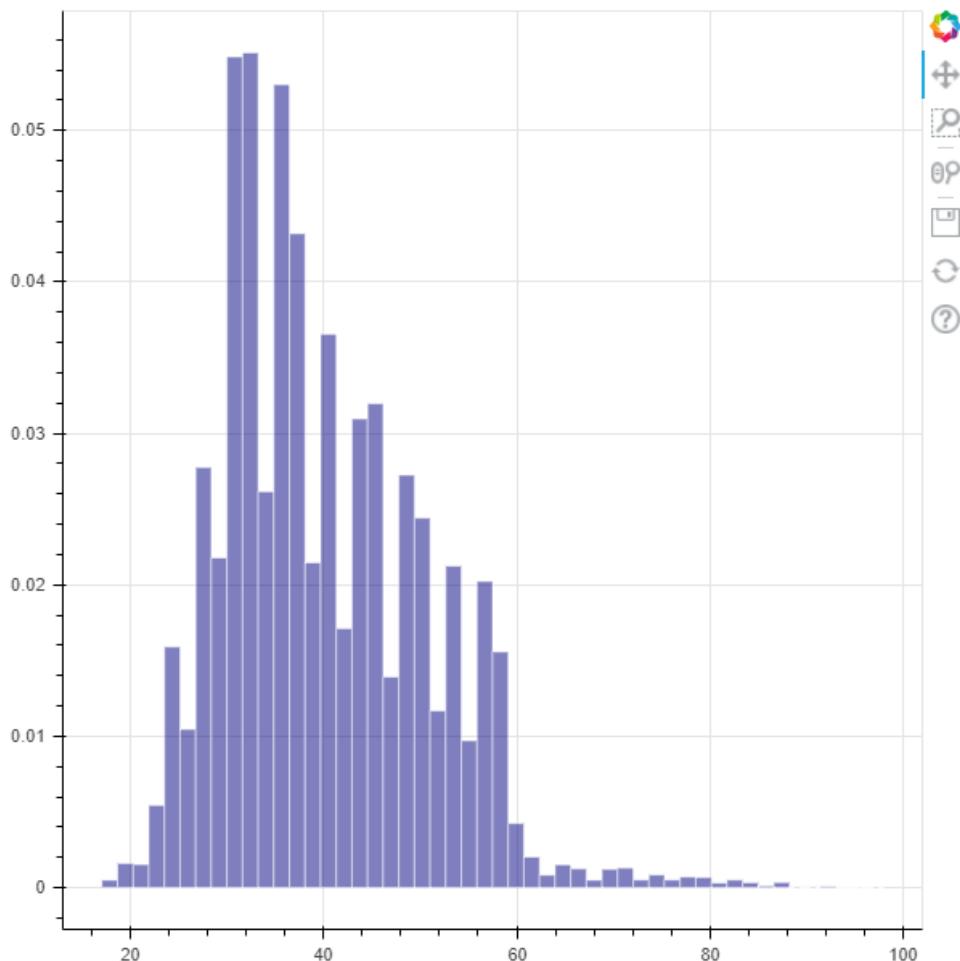
Nếu muốn xem biểu đồ phân bố tuổi của dữ liệu trên thì dùng đoạn code đầy đủ sau:

```
import numpy as np
import pandas as pd
from bokeh.plotting import figure, output_file, show

df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-
additional-full.csv', delimiter=';')
df_desc = df.describe()

hist, edges = np.histogram(df['age'], density=True, bins=50)

output_file("output.html")
p = figure()
p.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],
       fill_color="navy", line_color="white", alpha=0.5)
show(p)
```



Diễn giải:

- Đoạn code trên sử dụng hàm `numpy.histogram(...)` để tính toán mật độ phân bố dữ liệu tuổi. Tham số `bins = 50` cho biết độ mịn của các thanh đứng trong biểu đồ. Số này càng lớn thì số nhóm tuổi được thống kê sẽ còn lớn, tức là biểu đồ càng mịn.

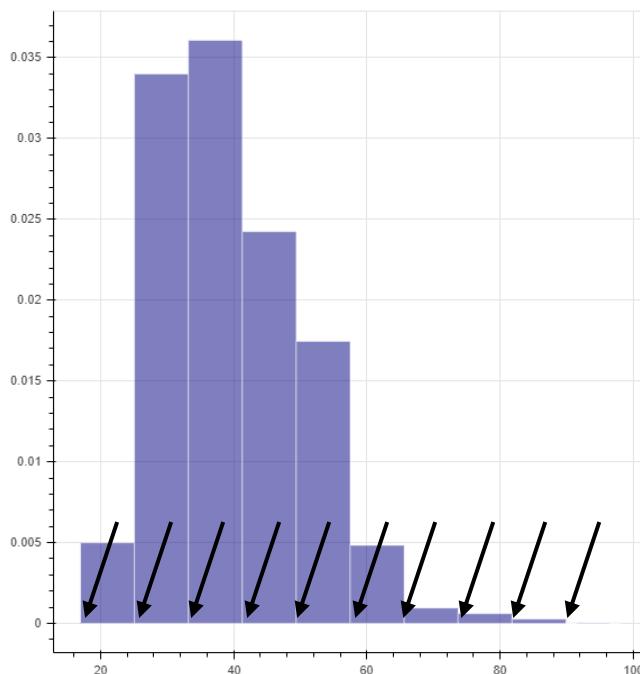
Tham số `bins` này có thể dùng chữ để chỉ phương pháp tính phân bố. Tra thêm tài liệu tại:

https://numpy.org/doc/stable/reference/generated/numpy.histogram_bin_edges.html#numpy.histogram_bin_edges.

Ví dụ sử dụng ‘auto’ như lệnh sau:

```
hist, edges = np.histogram(df['age'], density=True, bins='auto')
```

Kết quả `edges` chứa danh sách các giá trị trên trục x tại các vị trí biên của các thanh chữ nhật. Xem chỗ mũi tên trong hình bên dưới:



`hist` chứa danh sách các giá trị độ cao của các thanh chữ nhật. Thanh chữ nhật biểu diễn nhóm tuổi cần thống kê. Số thanh chữ nhật chính là số `bins` (nếu nó là số); hoặc do thuật toán xác định tự động nên `bins` là chữ (tên phương pháp cần áp dụng)

- Hàm để vẽ biểu đồ `figure.Quad(...)`

```
p.Quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],  
       fill_color="navy", line_color="white", alpha=0.5)
```

với ý nghĩa vài tham số như sau:

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

- top, bottom, left, right: tương ứng với số liệu cần để vẽ biểu đồ; độ cao, điểm đáy, danh sách cạnh trái của thanh chữ nhật, danh sách cạnh phải của thanh chữ nhật.
- fill_color: màu để phủ thanh chữ nhật.
- line_color: màu để kẻ đường bao thanh chữ nhật.
- alpha: độ trong suốt (transparent) của hình vẽ

Tham khảo thêm tài liệu tại:

<https://docs.bokeh.org/en/latest/docs/reference/plotting.html?highlight=quad#bokeh.plotting.Figure.quad>

Xem phân bố dữ liệu bằng biểu đồ Boxplot

Trong thư viện bokeh tự vẽ boxplot cho cột age của dữ liệu mẫu như sau:

```
import pandas as pd
from bokeh.plotting import figure, output_file, show

df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', delimiter=';')

output_file("output.html")

s = df['age']

q1 = s.quantile(q=0.25)
q2 = s.quantile(q=0.5)
q3 = s.quantile(q=0.75)

# Interquartile range
iqr = q3 - q1
upper = q3 + 1.5*iqr
lower = q1 - 1.5*iqr

out = s[(s > upper) | (s < lower)]

cats = ['age']
p = figure(x_range=cats)
# stems
```

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

```
p.segment(cats, upper, cats, q3, line_color="black")
p.segment(cats, lower, cats, q1, line_color="black")

# boxes
p.vbar(cats, 0.7, q2, q3, fill_color="#E08E79", line_color="black")
p.vbar(cats, 0.7, q1, q2, fill_color="#3B8686", line_color="black")

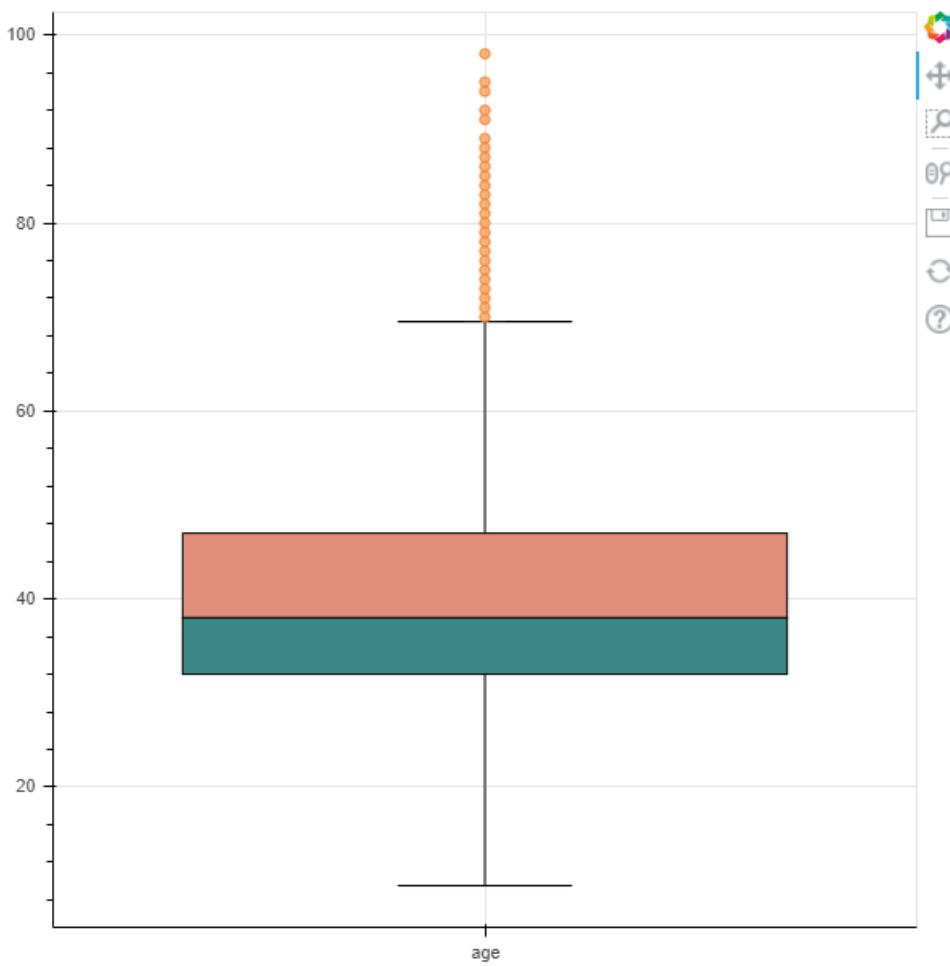
# whiskers (almost-0 height rects simpler than segments)
p.rect(cats, lower, 0.2, 0.01, line_color="black")
p.rect(cats, upper, 0.2, 0.01, line_color="black")

# prepare outlier data for plotting, we need coordinates for every
outlier.
if not out.empty:
    outy = pd.Series(out)
    outy = outy.unique()
    outx = [cats[0] for y in outy]
    # outliers
    p.circle(outx, outy, size=6, color="#F38630", fill_alpha=0.6)

# show the results
show(p)
```

Phần diễn giải mã nguồn sẽ được giải thích sau.

Tạm thời bạn xem kết quả của việc tự vẽ các thành phần của một boxplot cho cột age từ đoạn code trên như sau:



Biểu đồ so sánh 1 biến dùng Line Chart

```
import pandas as pd
from bokeh.plotting import figure, output_file, show

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]

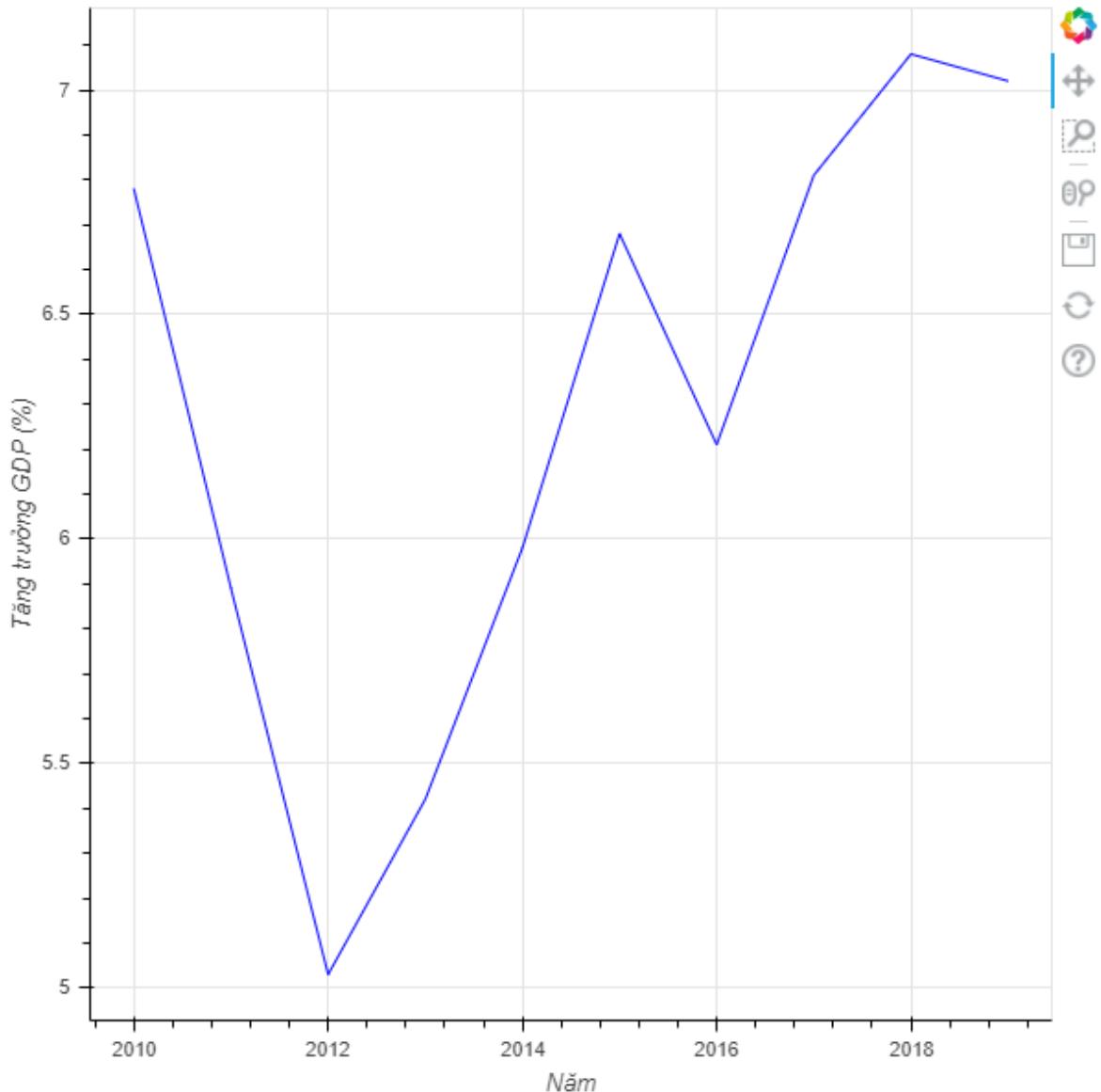
# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year})

output_file("output.html")
p = figure()

p.xaxis.axis_label = 'Năm'
p.yaxis.axis_label = 'Tăng trưởng GDP (%)'
```

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

```
p.line(  
    x = df['year'],  
    y = df['gdp'],  
    color = 'blue'  
)  
  
show(p)
```



Biểu đồ so sánh 2 biến dùng Line Chart

```
import pandas as pd  
from bokeh.plotting import figure, output_file, show  
  
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
```

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

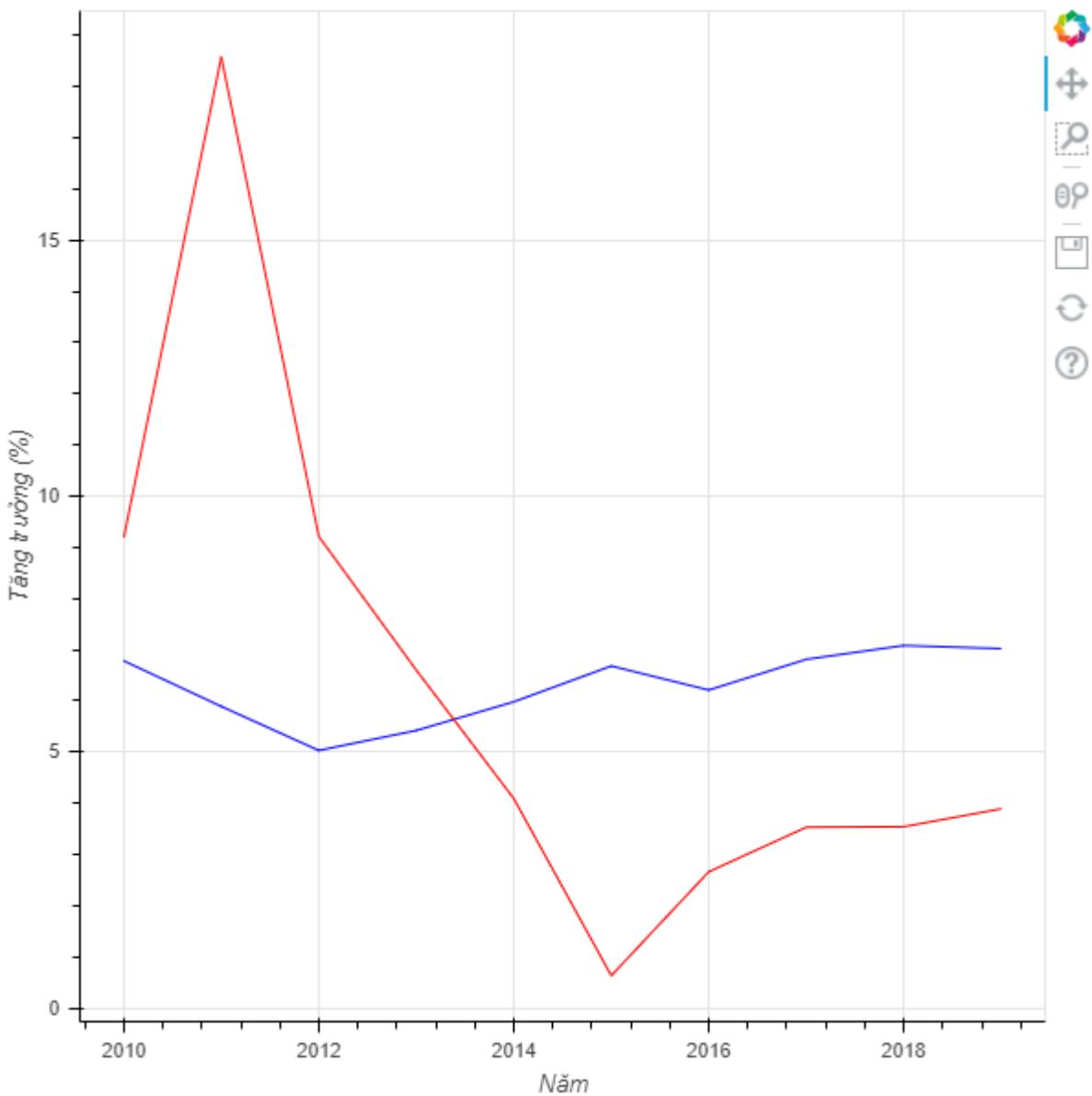
```
year = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019]
cpi = [9.19,18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54, 3.89]
df = pd.DataFrame({'gdp': gdp, 'year': year, 'cpi': cpi})

output_file("output.html")
p = figure()

p.xaxis.axis_label = 'Năm'
p.yaxis.axis_label = 'Tăng trưởng (%)'

p.line(
    x = df['year'],
    y = df['gdp'],
    color = 'blue'
)

p.line(
    x = df['year'],
    y = df['cpi'],
    color = 'red'
)
show(p)
```



Biểu đồ so sánh nhiều biến dùng Line Chart

Bokeh cung cấp hàm multi_line để plot dữ liệu từ các bộ dữ liệu tương ứng cho trục x và y. Trong ví dụ bên dưới hàm multi_line nhận tham số xs và ys.

xs là một array gồm 2 phần tử [x1, x2]. x1 và x2 lại mảng các năm.

ys là một array tương ứng gồm 2 phần tử [y1, y2] với y1 là mảng các giá trị gdp (tương ứng với từng năm trong x1); y2 là mảng các giá trị cpi (tương ứng với từng năm trong x2).

```
from bokeh.plotting import figure, output_file, show

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]
```

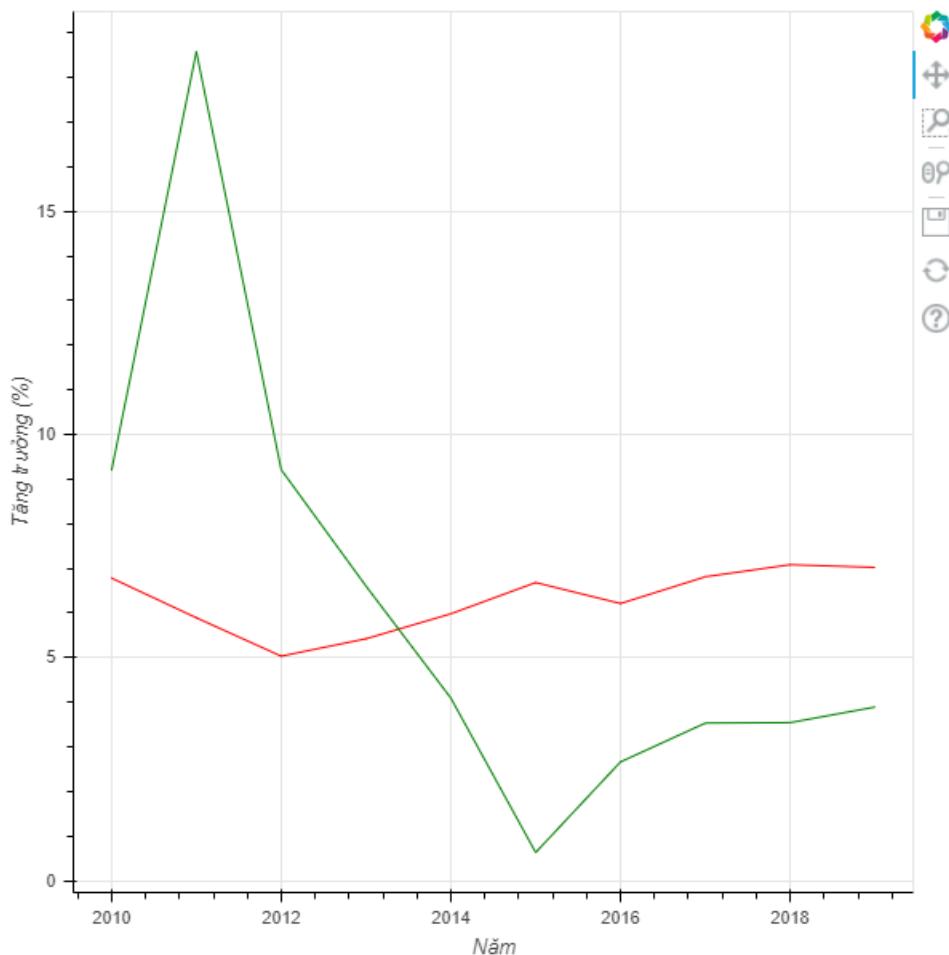
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
output_file("output.html")
p = figure()

p.xaxis.axis_label = 'Năm'
p.yaxis.axis_label = 'Tăng trưởng (%)'

p.multi_line(xs=[year, year], ys=[gdp, cpi], color=['red','green'])

show(p)
```



Biểu đồ so sánh nhiều biến dùng Line Chart với Data Source

Trên cơ sở ý tưởng dùng các cặp giá trị (x,y) tương ứng cho nhiều line khi dùng hàm multi_line ở trên thì phần này cải tiến một chút bằng cách dùng Data Source.

Xem và trải nghiệm đoạn code bên dưới:

```
from bokeh.models import ColumnDataSource
from bokeh.plotting import figure, output_file, show
```

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

```
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]

output_file("output.html")

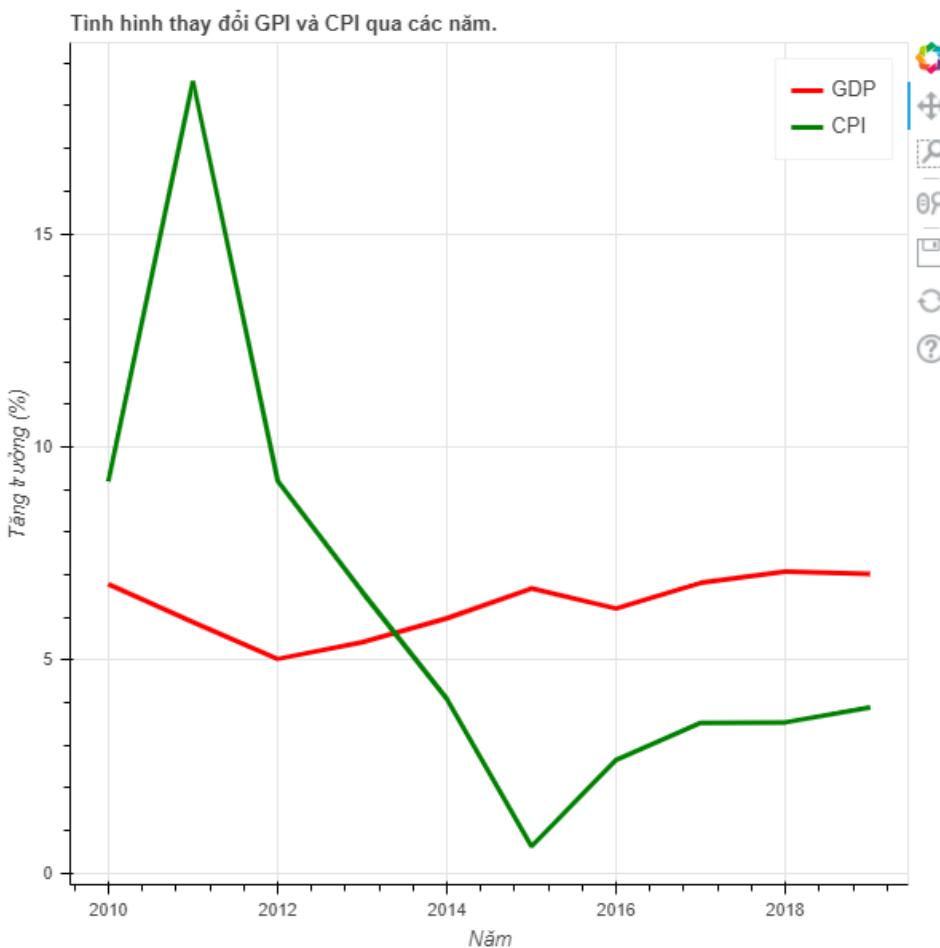
p = figure(title="Tình hình thay đổi GPI và CPI qua các năm.")

p.xaxis.axis_label = 'Năm'
p.yaxis.axis_label = 'Tăng trưởng (%)'

ds = ColumnDataSource(data={'x': [year, year], 'y': [gdp, cpi],
                           'color': ['red', 'green'],
                           'label': ['GDP', 'CPI']}))

p.multi_line('x', 'y', color='color', legend='label', line_width=3,
             source=ds)

show(p)
```



Giới thiệu Data Source trong Bokeh

Ý tưởng dùng Data Source nhằm mục đích cung cấp một nguồn dữ liệu cho biểu đồ. Khi nguồn dữ liệu thay đổi về nội dung thì biểu đồ sẽ được vẽ lại. Ý tưởng này sẽ giúp tạo ra các biểu đồ có tính tương tác (interactive chart). Đây là một trong các điểm mạnh của thư viện Bokeh.

Quay lại ví dụ trong phần trước, cần phân tích và điểm chính trong mã nguồn:

- Để sử dụng nguồn dữ liệu cho biểu đồ thì khai báo sử dụng lớp ColumnDataSource từ module bokeh.models:

```
from bokeh.models import ColumnDataSource
```

- Chuẩn bị một biến dạng Dictionary để ánh xạ các dữ liệu vào key như sau:

```
dict_data = {'x': [year, year], 'y': [gdp, cpi],  
            'color': ['red', 'green'],  
            'label': ['GDP', 'CPI']}
```

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

Phần **màu đỏ** là key, **màu xanh** là values (theo ví dụ). Giá trị theo dạng gì là tùy ý nghĩa của key và dùng vào mục gì trong hàm vẽ biểu đồ. Key là do bạn tự đặt để khai báo tương ứng cho hàm vẽ biểu đồ.

- Khởi tạo đối tượng ColumnDataSource với tham số là một trong các dạng dữ liệu sau:

- Dữ liệu dạng Dictionary đã chuẩn bị ở trên:

```
ds = ColumnDataSource(data=dict_data)
```

- Dữ liệu là DataFrame:

```
ds = ColumnDataSource(data=df)
```

- Dữ liệu dạng GroupBy:

```
group = df.groupby(['colA', 'ColB'])
ds = ColumnDataSource(data=group)
```

- Để vẽ biểu đồ nhiều line với hàm multi_line thì tham số source được khai báo dùng biến ds:

```
p.multi_line('x', 'y', color='color', legend='label',
line_width=3, source=ds)
```

Trong đó các giá trị 'x', 'y', 'color', 'label' chính là các key trong dict data đã chuẩn bị ở trên.

Thuộc tính quan trọng của ColumnDataSource

Thuộc tính data

Để truy cập dữ liệu của ColumnDataSource thì Bokeh cung cấp thuộc tính data. Ví dụ để thay đổi dữ liệu của Data Source thì dùng cú pháp sau:

```
ds.data = <new dict data>
```

Biểu đồ với dữ liệu thay đổi

Để minh họa ý tưởng sử dụng Data Source để vẽ biểu đồ với dữ liệu thay đổi theo thời gian thực thì chúng ta cải tiến ví dụ vẽ biểu đồ GDP và CPI một chút:

```
from bokeh.server.server import Server
from bokeh.application import Application
from bokeh.application.handlers.function import FunctionHandler
```

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

```
from bokeh.plotting import figure, ColumnDataSource
import random

def make_document(doc):
    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]

    ds = ColumnDataSource(data={'x': [year, year], 'y': [gdp, cpi],
                               'color': ['red', 'green'],
                               'label': ['GDP', 'CPI']}))

    def update():
        cpi = [9.19,18.58, 9.21, 6.60, 4.09, random.randint(1, 100),
               2.66, 3.53, 3.54, 3.89]
        new_ds = {'x': [year, year], 'y': [gdp, cpi],
                  'color': ['red', 'green'],
                  'label': ['GDP', 'CPI']}
        ds.data = new_ds

    doc.add_periodic_callback(update, 100)

    p = figure(title="Tình hình thay đổi GPI và CPI qua các năm.")

    p.xaxis.axis_label = 'Năm'
    p.yaxis.axis_label = 'Tăng trưởng (%)'

    p.multi_line('x', 'y', color='color', legend='label',
                 line_width=3, source=ds)

    doc.title = "Now with live updating!"
    doc.add_root(p)
```

```
apps = {'/': Application(FunctionHandler(make_document))}

server = Server(apps, port=5000)
server.start()
print('View dynamic chart at http://localhost:5000')
```

Hãy thực thi đoạn lệnh trên và mở trình duyệt với địa chỉ sau để xem kết quả:

<http://localhost:5000>

Để tắt bokeh server thì thực hiện lệnh sau:

```
server.stop()
```

Vẽ biểu đồ nhiều line bằng hàm vline_stack

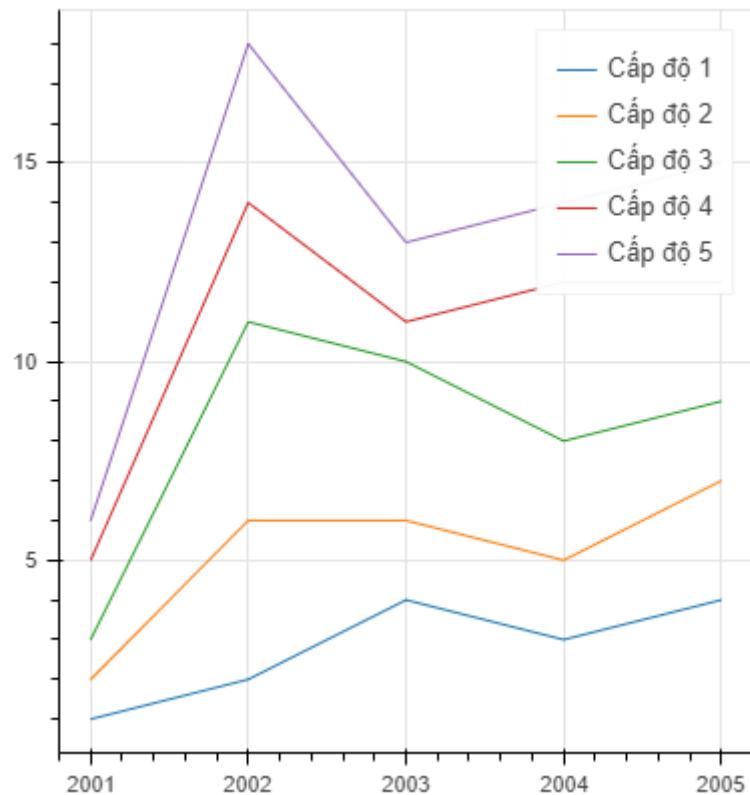
```
from bokeh.models import ColumnDataSource
from bokeh.plotting import figure, show
from bokeh.palettes import Category10_10

source = ColumnDataSource(data=dict(
    x=[2015, 2016, 2017, 2018, 2019],
    y1=[1, 2, 4, 3, 4],
    y2=[1, 4, 2, 2, 3],
    y3=[1, 5, 4, 3, 2],
    y4=[2, 3, 1, 4, 3],
    y5=[1, 4, 2, 2, 3],
))
levels = [1, 2, 3, 4, 5]
labels = ['Cấp độ %s' % x for x in levels]

p = figure(plot_width=400, plot_height=400, title='Lịch sử hoàn thành công việc theo cấp độ')

p.vline_stack(['y1', 'y2', 'y3', 'y4', 'y5'], x='x', source=source,
color=Category10_10[0:len(levels)], legend_label=labels)

show(p)
```



Biểu đồ so sánh - Bar Chart

Sử dụng hàm vbar của figure

Một cách đơn giản là dùng hàm vbar để vẽ các thanh đứng (vertical) như ví dụ bên dưới với các thông tin:

- x là mảng chỉ định các vị trí của thanh
- top là mảng chỉ định độ cao các thanh
- width: độ rộng các thanh

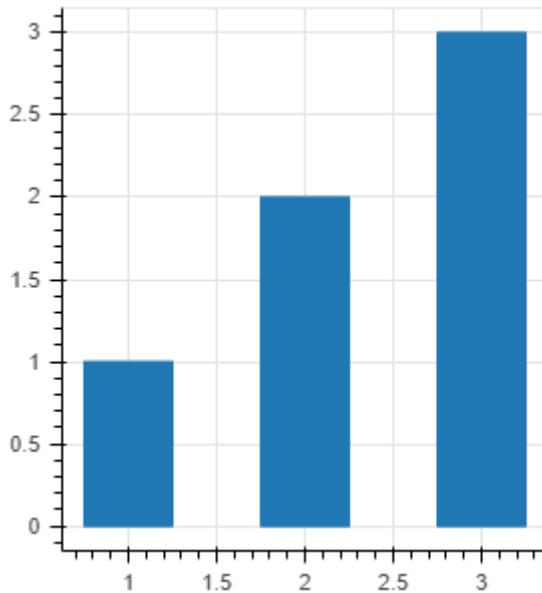
Tra cứu các tham số khác tại:

```
https://docs.bokeh.org/en/latest/docs/reference/plotting.html?highlight=vbar#bokeh.plotting.Figure.vbar
```

```
from bokeh.plotting import figure, show

plot = figure(plot_width=300, plot_height=300)
plot.vbar(x=[1, 2, 3], width=0.5, top=[1,2,3])

show(plot)
```

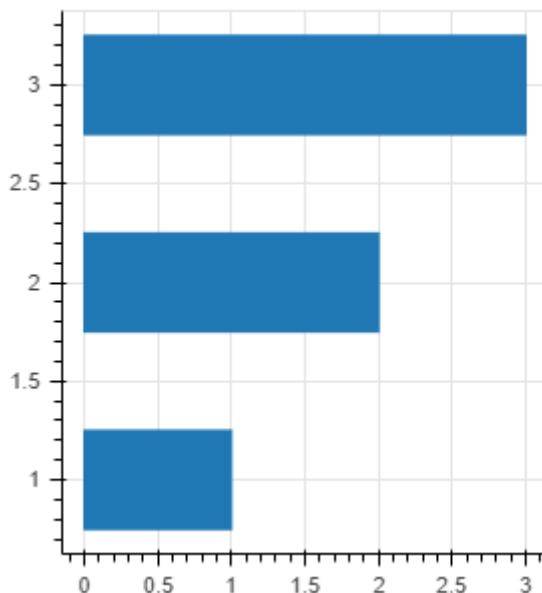


Sử dụng hàm hbar

```
from bokeh.plotting import figure, show

plot = figure(plot_width=300, plot_height=300)
plot.hbar(y=[1, 2, 3], height=0.5, right=[1,2,3])

show(plot)
```



Sử dụng bar chart với data source

```
from bokeh.plotting import figure, show
from bokeh.models import ColumnDataSource

# create a Python dict as the basis of your ColumnDataSource
data = {'x_values': [1, 2, 3],
        'y_values': [1, 2, 3]}

# create a ColumnDataSource by passing the dict
source = ColumnDataSource(data=data)

plot = figure(plot_width=300, plot_height=300)
plot.vbar(x='x_values', width=0.5, top='y_values', source=source)

show(plot)
```

Đưa phần Legend ra khỏi biểu đồ

Các biểu đồ trong các phần trước thì phần Legend (các chú giải màu sắc, đường nét trong biểu đồ) nếu có thì nó được đặt bên trong biểu đồ. Điều này đôi lúc nội dung của biểu đồ bị che mất. Để đưa phần Legend ra khỏi biểu đồ thì cần thiết lập layout cho biểu đồ với vị trí Legend cụ thể. Ví dụ đoạn code bên dưới thiết lập Legend được trình bày theo hàng ngang và ở phía trên của buổi đồ.

```
from bokeh.models import Legend
p = figure()
p.add_layout(Legend(orientation = 'horizontal'), 'above')
```

Xem cách sử dụng trong phần tiếp theo.

Sử dụng bar chart nhiều màu

```
from bokeh.io import show
from bokeh.models import ColumnDataSource, Legend
from bokeh.palettes import Category10_10
from bokeh.plotting import figure
from bokeh.transform import factor_cmap

fruits = ['Apples', 'Pears', 'Nectarines']
counts = [5, 3, 4]
```

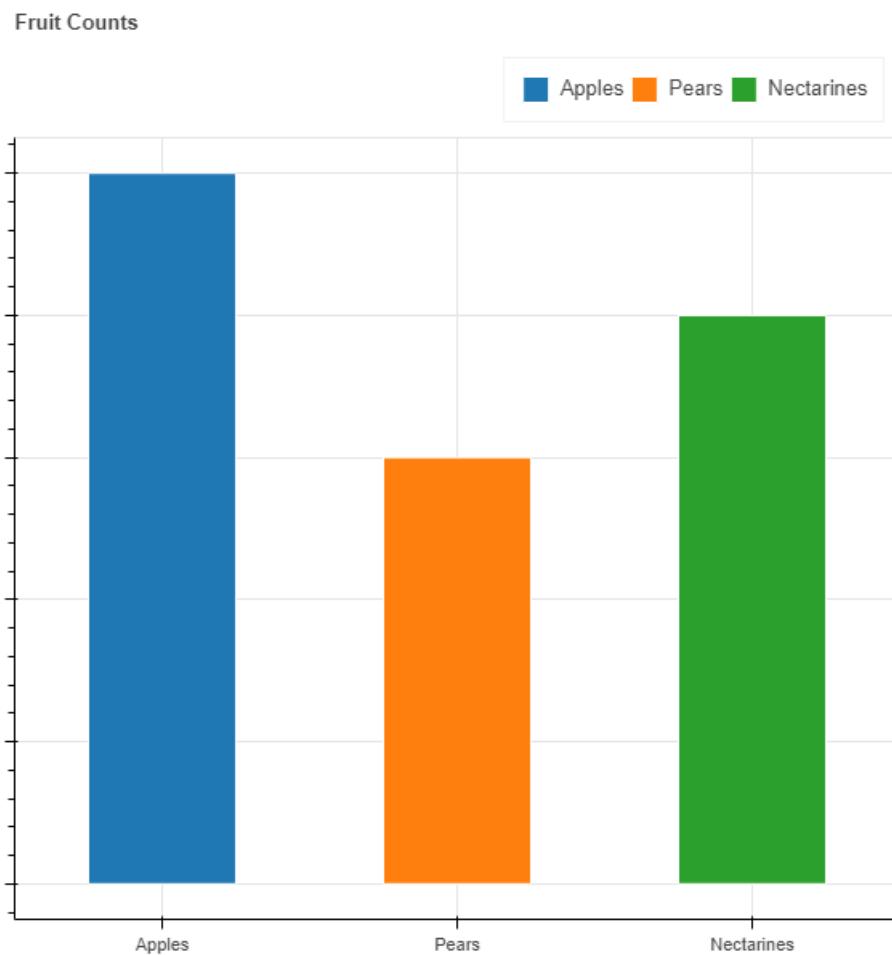
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
source = ColumnDataSource(data=dict(fruits=fruits, counts=counts))

p = figure(x_range=fruits, title="Fruit Counts")
p.add_layout(Legend(orientation = 'horizontal'), 'above')

colors = factor_cmap('fruits', palette=Category10_10, factors=fruits)
p.vbar(x='fruits', top='counts', width=0.5, source=source,
legend="fruits",
line_color='white', fill_color=colors)

show(p)
```



Vẽ bar chart theo nhóm

```
from bokeh.core.properties import value
from bokeh.io import show
from bokeh.models import ColumnDataSource
from bokeh.plotting import figure
from bokeh.transform import dodge
from bokeh.palettes import Category10_10

fruits = ['Apples', 'Pears', 'Nectarines']
years = ['2015', '2016', '2017']

data = {'fruits' : fruits,
        '2015'   : [2, 1, 4],
        '2016'   : [5, 3, 3],
        '2017'   : [3, 2, 4]}

source = ColumnDataSource(data=data)

p = figure(x_range=fruits, y_range=(0, 10), plot_height=250,
           title="Fruit Counts by Year")

p.vbar(x=dodge('fruits', -0.25, range=p.x_range), top='2015',
       width=0.2, source=source,
       color=Category10_10[0], legend=value("2015"))

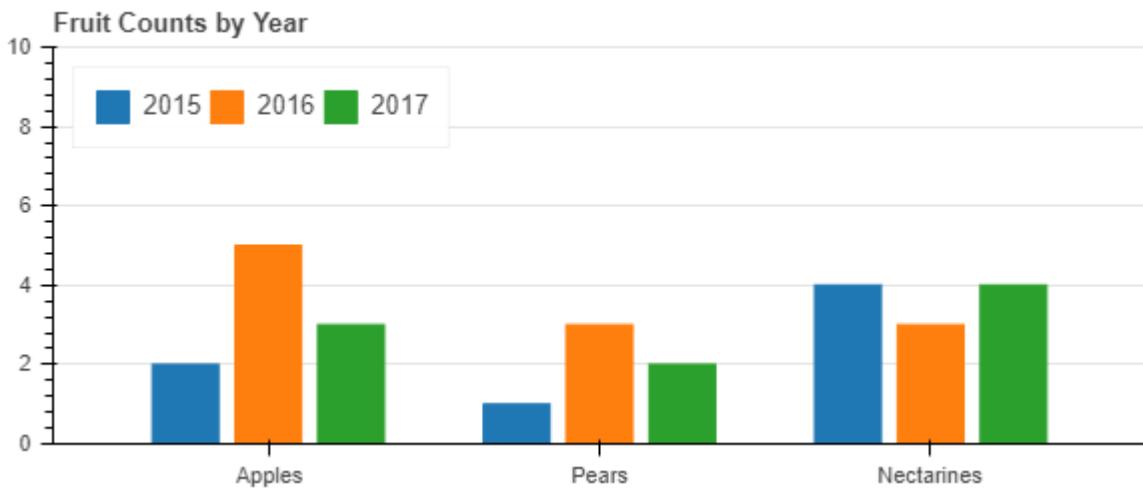
p.vbar(x=dodge('fruits', 0.0, range=p.x_range), top='2016',
       width=0.2, source=source,
       color=Category10_10[1], legend=value("2016"))

p.vbar(x=dodge('fruits', 0.25, range=p.x_range), top='2017',
       width=0.2, source=source,
       color=Category10_10[2], legend=value("2017"))

p.x_range.range_padding = 0.1
p.xgrid.grid_line_color = None
p.legend.location = "top_left"
p.legend.orientation = "horizontal"
```

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

```
show(p)
```



Vẽ bar chart theo nhóm chồng lên nhau

```
from bokeh.models import Legend
from bokeh.plotting import figure, show
from bokeh.palettes import Category10_10

projects = ['Dự án %s' % (x+1) for x in range(6)]
status = [ 'Đang làm', 'Hoàn thành', 'Tạm hoãn', 'Hủy', 'Duyệt' ]
colors = Category10_10[0:5]

data = {'projects' : projects,
        'Đang làm'    : [2, 1, 4, 3, 2, 4],
        'Hoàn thành'   : [5, 3, 4, 2, 4, 6],
        'Tạm hoãn'    : [3, 2, 4, 4, 5, 3],
        'Hủy'         : [3, 2, 4, 4, 5, 3],
        'Duyệt'        : [3, 2, 4, 4, 5, 3],
        }

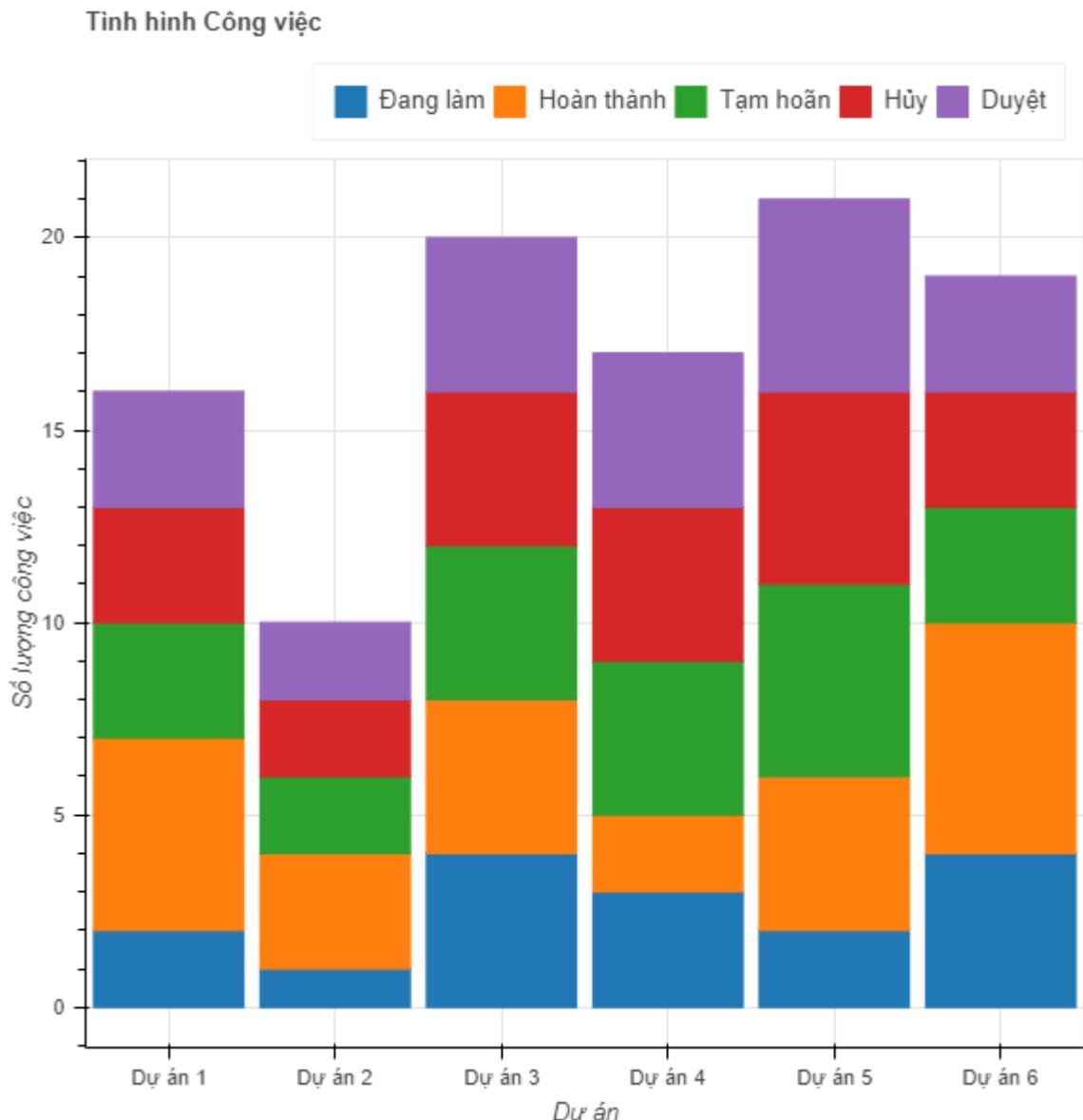
p = figure(x_range=projects, title="Tình hình Công việc")
p.add_layout(Legend(orientation = 'horizontal'), 'above')

p.vbar_stack(status, x='projects', width=0.9, color=colors,
              source=data, legend_label=status)

p.xaxis.axis_label = 'Dự án'
p.yaxis.axis_label = 'Số lượng công việc'
```

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

show (p)



Xem thêm cách vẽ các thành phần cơ bản tại link:

https://docs.bokeh.org/en/latest/docs/user_guide/plotting.html

Xem thêm cách vẽ các loại biểu đồ tại link:

<https://docs.bokeh.org/en/latest/docs/reference/plotting.html>

Trình bày giao diện với Bokeh

Một trong các điểm mạnh của thư viện Bokeh là hỗ trợ kiến trúc giúp chúng ta có thể trình bày trực quan kết quả phân tích dữ liệu dưới dạng một ứng dụng web tương tác hoàn chỉnh.

Phần này sẽ giúp bạn làm quen với cách thiết kế một trang web gồm nhiều thành phần, trình bày dưới dạng nhiều bố cục (layout) khác nhau.

Bố cục dạng cột

Để trình bày các biểu đồ trên cùng một cột thì dùng hàm `column(..)` để liệt kê các đối tượng Figure trong các tham số như ví dụ sau:

```
from bokeh.io import output_file, show
from bokeh.layouts import column
from bokeh.plotting import figure

output_file("layout_column.html")

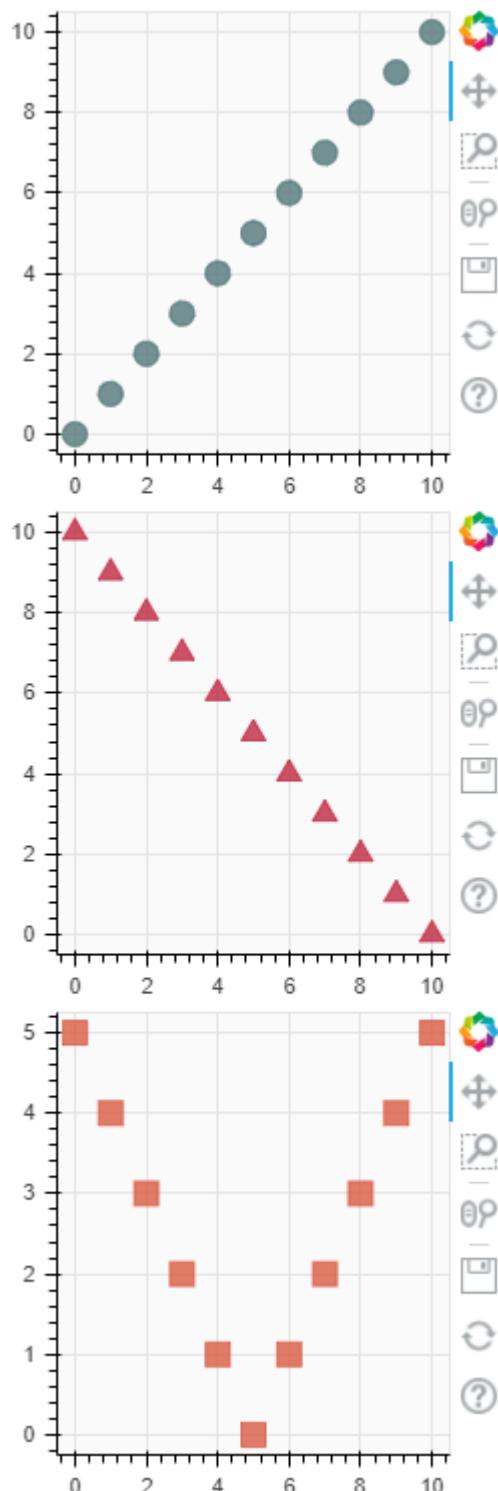
x = list(range(11))
y0 = x
y1 = [10 - i for i in x]
y2 = [abs(i - 5) for i in x]

# create three plots
s1 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s1.circle(x, y0, size=12, color="#53777a", alpha=0.8)

s2 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s2.triangle(x, y1, size=12, color="#c02942", alpha=0.8)

s3 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s3.square(x, y2, size=12, color="#d95b43", alpha=0.8)

# put the results in a column and show
layout = column(s1, s2, s3)
show(layout)
```



Bố cục dạng dòng

```
from bokeh.io import output_file, show
from bokeh.layouts import row
from bokeh.plotting import figure

output_file("layout_row.html")
```

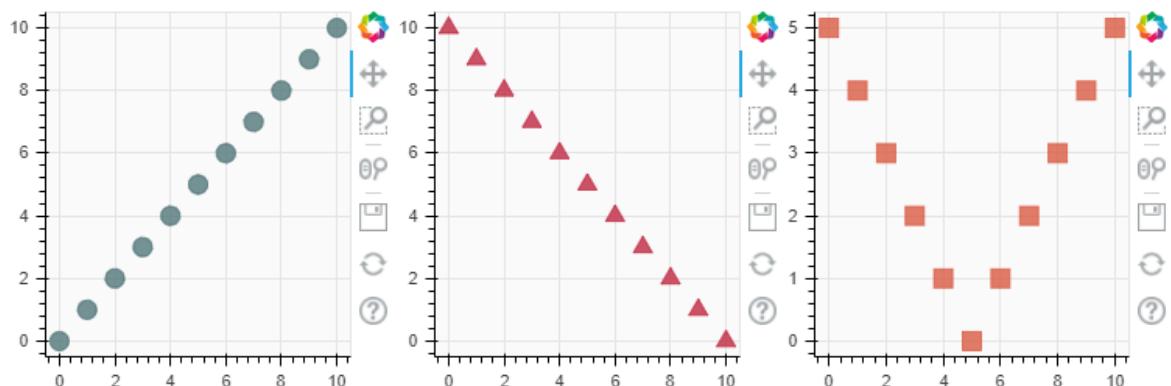
```
x = list(range(11))
y0 = x
y1 = [10 - i for i in x]
y2 = [abs(i - 5) for i in x]

# create three plots
s1 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s1.circle(x, y0, size=12, color="#53777a", alpha=0.8)

s2 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s2.triangle(x, y1, size=12, color="#c02942", alpha=0.8)

s3 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s3.square(x, y2, size=12, color="#d95b43", alpha=0.8)

# put the results in a row and show
layout = row(s1, s2, s3)
show(layout)
```



Kết hợp dòng và cột

Trang trí biểu đồ

Thiết lập tiêu đề, tên trục x và y

```
import pandas as pd
from bokeh.plotting import figure
from bokeh.models.annotations import Title
```

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

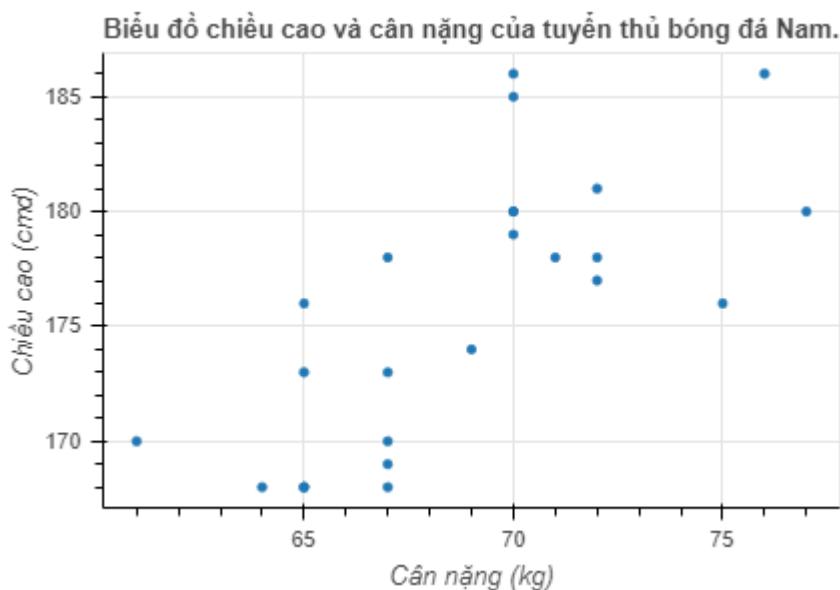
```
from bokeh.io import show

df =
pd.read_csv('https://thachln.github.io/datasets/TuyenVN_2019.csv')

x = df['Weight']
y = df['Height']

# plot it
p = figure()
p.circle(x,y)

p.xaxis.axis_label = 'Cân nặng (kg)'
p.yaxis.axis_label = 'Chiều cao (cm)'
p.title = Title(text = 'Biểu đồ chiều cao và cân nặng của tuyển thủ bóng đá Nam.')
show(p)
```



Thay đổi vị trí của legend

```
p.legend.location = "top_left"
```

Ngày 3 - Phân tích mô tả

Trong ngày 2 chúng ta đã cảm nhận được phần nào các loại biểu đồ cơ bản trong phân tích dữ liệu. Đặc biệt là làm quen với cách thể hiện biểu đồ với R và Python.

Trong bài này chúng ta trải nghiệm ở góc độ ứng dụng thông qua một số tình huống cụ thể. Từ đó chúng ta luyện tập thêm một số kỹ năng phân tích thông qua biểu đồ để mô tả dữ liệu trực quan hơn.

Phân tích mô tả (Descriptive analysis) đúng như tên gọi của nó thì đây là hướng phân tích nhằm mô tả thông tin về dữ liệu. Cụ thể là phân tích một cách định lượng (quantitatively) và tổng hợp thông tin theo hướng thống kê (statistically) những gì mà dữ liệu có thể có. Ví dụ trong tay bạn có dữ liệu bán hàng của công ty bạn thì bạn sẽ phân tích để mô tả nhiều thông tin đáng giá bằng cách đi tìm câu trả lời cho các câu hỏi sau:



Mặt hàng nào được bán chạy nhất?



Tháng này tình hình kinh doanh có khác biệt so với tháng này năm ngoái?



Doanh số trung bình của mỗi mặt hàng là bao nhiêu?

Nếu kết hợp thêm chi phí tiền lương và thời gian làm việc của nhân viên thì bạn có thể trả lời thêm các câu hỏi phức tạp hơn như:



Mỗi đồng lương bạn chi ra cho nhân viên thì sẽ mang lại doanh số bao nhiêu?



Mỗi giờ làm việc trung bình của nhân viên thì sẽ mang lại cho công ty bao nhiêu tiền?



Một năm kinh nghiệm trung bình của nhân viên tương ứng với doanh số bao nhiêu?

Tổng hợp thông tin theo hướng thống kê cụ thể là chúng ta tính toán các chỉ số cơ bản mà tôi đã giúp các bạn làm quen, ôn lại trong Bài 1. Bài này tôi tóm tắt lại 10 chỉ số quan trọng sau đây:

Chỉ số	Giải thích
Mean	Số trung bình
Median	Số trung vị
Mode	Giá trị lặp lại nhiều nhất
Percentile	Bách phân vị. Thường dùng bách phân vị 25%, 75%. Bách phân vị 50% chính là Median
Quartiles	Gồm 4 chỉ số: Minimum: Số nhỏ nhất Lower quartile: Bách phân vị 25%

	Upper quartile: Bách phân vị 75%
	Maximum: Số lớn nhất
Standard deviation	Độ lệch chuẩn
Variance	Phương sai
Range	Giải giá trị: được tính bằng hiệu của Maximum và Minimum
Proportion	Tỉ số
Correlation	Coefficient of correlation – Hệ số tương quan

Ngày thứ ba này sẽ gồm 4 bài:

Bài 12: Minh họa phân tích tả từ dữ liệu về một dự án (hay còn gọi là nghiên cứu) từ bộ phận Marketing của một Ngân hàng. Dữ liệu và code mẫu tham khảo từ cuốn sách Hands-On Data Science for Marketing (Packt Publishing, 2019) của Yoon Hyup Hwang.

Bài 13: Giúp bạn nắm được cách so sánh hai tỉ lệ.

Bài 14: Tóm tắt lại lý thuyết về Mô hình kiểm định giả thuyết.

Bài 15: Mô tả một vài ứng dụng minh họa để quen với Phân tích phương sai, Phân tích t-test và Kiểm định giả thuyết.

Bài 16: Ứng dụng Kiểm định giả thuyết bằng code Python

Bài 17: Đi tìm mối tương quan trong dữ liệu.

Bài 13, 14, 15 tôi dùng khá nhiều kiến thức, tài liệu từ các lớp học Phân tích dữ liệu cơ bản, Phân tích dữ liệu nâng cao của nhóm các Bác sĩ và anh Nguyễn Văn Tuấn. Các lớp học này tổ chức tại trường Đại Học Tôn Đức Thắng ở Sài Gòn.

Bài 12: Phân tích mô tả dữ liệu Bank Marketing

Dữ liệu Bank Marketing

Để minh họa cho bài này tôi dùng dữ liệu về Bank Marketing của UCI⁸. Tải 2 file bank.zip và bank-additional.zip trong mục Data Folder.

The screenshot shows the UCI Machine Learning Repository homepage. At the top, there's a logo with a blue antelope and the text "UCI Machine Learning Repository". Below it, it says "Center for Machine Learning and Intelligent Systems". On the right side, there are links for "About", "Citation Policy", "Donate a Data Set", and "Contact". There's also a search bar and a "Google" button. A sidebar on the right has radio buttons for "Repository" and "Web", and a link "View ALL Data Sets". The main content area is titled "Bank Marketing Data Set" and includes links for "Download: Data Folder" and "Data Set Description". Below that is a "Abstract" section with the text: "The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit (variable y)."

Đây là dữ liệu thu thập được từ một dự án tìm kiếm khách hàng (direct marketing campaign) bằng điện thoại để chào một dịch vụ hoặc sản phẩm của ngân hàng cho khách hàng ở Bồ Đào Nha. Dịch vụ hoặc sản phẩm theo thuật ngữ ngân hàng ở đây là **deposit**.

Dữ liệu gồm các biến đầu vào (input variables):

#	Biến	Ý nghĩa (kiểu dữ liệu, giải thích)
Nhóm thông tin về khách hàng (bank client data)		
1	age	tuổi (numeric)
2	job	nghề nghiệp
3	marital	tình trạng hôn nhân. 'divorced' (<i>ly dị hoặc góa phụ</i>), 'married' (<i>đã lập gia đình</i>), 'single' (<i>độc thân</i>), 'unknown' (<i>không biết</i>).
4	education	Trình độ học vấn
5	default	Có quan hệ tín dụng hay không: 'no','yes','unknown'. Được hiểu là có đang vay tiền ngân hàng không?
6	housing	Có khoản vay nhà ở không? 'no','yes','unknown'
7	loan	Có khoản vay cá nhân không? 'no','yes','unknown'
Nhóm thông tin về lần liên lạc gần đây trong chiến dịch		
8	contact	Kênh liên lạc: 'cellular','telephone'

⁸ Link: <https://archive.ics.uci.edu/ml/datasets/bank+marketing>

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

9	month	Tháng liên lạc gần đây nhất: 'jan', 'feb', 'mar', ..., 'nov', 'dec'
10	day_of_week	Ngày liên lạc gần đây nhất: 'mon', 'tue', 'wed', 'thu', 'fri'
11	duration	Thời lượng cuộc gọi gần đây: tính bằng giây.
12	campain	Số lần liên lạc với khách hàng này. Giá trị số có nghĩa là lần liên lạc gần nhất)
13	pdays	Số ngày tính từ lần liên lạc trong chiến dịch gần đây. Giá trị 999 có nghĩa là không có liên lạc trong chiến dịch trước.
14	previous	Số lần liên lạc trước khi chiến dịch này xảy ra.
15	poutcome	Kết quả của chiến dịch marketing lần trước: 'failure', 'nonexistent', 'success'
Nhóm thông tin về kinh tế và xã hội		
16	emp.var.rate	Tỉ lệ công ăn việc làm - employment variation rate - quarterly indicator (numeric)
17	cons.prive.idx	Chỉ số giá tiêu dùng - consumer price index - monthly indicator (numeric)
18	cons.conf.idx	Chỉ số niềm tin tiêu dùng - consumer confidence index - monthly indicator (numeric)
19	euribor3m	Tỉ lệ chào bán liên ngân hàng Euro - euribor 3 month rate - daily indicator (numeric). Là tỉ lệ tham chiếu được xây dựng từ lãi suất trung bình mà các ngân hàng Châu Âu cung cấp cho vay ngắn hạn không có tài sản bảo đảm trên thị trường liên ngân hàng
20	nr.employed	Số người lao động - number of employees - quarterly indicator (numeric)
Biến kết quả		
21	y	biến nhị phân: 'yes', 'no'. Khách hàng đồng ý sử dụng dịch vụ yes, không đồng ý là no.

Hai phần tiếp theo sẽ áp dụng các kiến thức đã học ở trên, đặc biệt là kỹ thuật vẽ biểu đồ để hỗ trợ phân tích mô tả cho dữ liệu Bank Marketing. Phần vẽ biểu đồ thì tôi không tập trung vào việc trang trí khi không cần thiết để code không quá rối. Bạn hoàn toàn có thể áp dụng các kỹ thuật trang trí biểu đồ ở phần trước để hoàn thiện hơn “bức tranh” theo ý bạn muốn.

Phân tích Bank Marketing với Python

Phần này giúp cho các bạn yêu thích Python trải nghiệm một chút để quen tay.

Xem cột dữ liệu và làm quen với vài dữ liệu

Đọc dữ liệu vào data frame dùng thư viện pandas như sau:

```
import pandas as pd  
df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', sep=';')
```

Hoặc đọc file đã xử lý sang file csv với dấu phẩy làm phân cách mặc định như sau:

```
import pandas as pd  
df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-additional-full-processed.csv')
```

Quan sát vài dòng dữ liệu bằng hàm head của thư viện pandas:

```
df.head()
```

Output:

```
age          job marital ... euribor3m nr.employed    y  
0   56  housemaid  married ...      4.857     5191.0  no  
1   57    services  married ...      4.857     5191.0  no  
2   37    services  married ...      4.857     5191.0  no  
3   40    admin.  married ...      4.857     5191.0  no  
4   56    services  married ...      4.857     5191.0  no
```

[5 rows x 21 columns]

Cột y là biến kết quả (thuật ngữ trong ngữ cảnh này là desired target).

Cột y có giá trị là yes | no, cần mã hóa thành dạng số 1 | 0. Dòng code sau sẽ thêm cột conversion sẽ có giá trị 1 | 0 tương ứng với yes | no trong cột y. Sau đó head lại xem kết quả:

```
df['conversion'] = df['y'].apply(lambda x: 1 if x == 'yes' else 0)  
df.head()
```

```
age          job marital education ... euribor3m nr.employed    y conversion  
0   56  housemaid  married basic.4y ...      4.857     5191.0  no          0  
1   57    services  married high.school ...      4.857     5191.0  no          0  
2   37    services  married high.school ...      4.857     5191.0  no          0  
3   40    admin.  married basic.6y ...      4.857     5191.0  no          0  
4   56    services  married high.school ...      4.857     5191.0  no          0
```

[5 rows x 22 columns]

Quan sát một số chỉ số cơ bản

Lệnh

Kết quả

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

<code>df['age'].mean()</code>	40.02406040594348
<code>df['age'].median()</code>	38.0
<code>df['age'].mode()</code>	0 31 dtype: int64
<code>df['age'].quantile()</code>	38.0
<code>df['age'].min()</code>	17
<code>df['age'].max()</code>	98
<code>df['age'].std()</code>	10.421249980934235
<code>df['age'].var()</code>	108.60245116512178

Có thể dùng hàm .describe() cho một cột dữ liệu như age:

```
df['age'].describe()
```

```
count    41188.00000
mean      40.02406
std       10.42125
min      17.00000
25%      32.00000
50%      38.00000
75%      47.00000
max      98.00000
Name: age, dtype: float64
```

Ghi chú:

- Thư viện pandas trong Python cũng có sẵn các hàm để tính các chỉ số thống kê cơ bản. Trong đó các hàm được gọi trực tiếp từ dữ liệu. Đây chính là triết lý của lập trình hướng đối tượng (OOP - Object Oriented Programming). OOP cho phép chúng ta gọi hàm (hay còn gọi là call message) từ một đối tượng (object) bằng cú pháp:

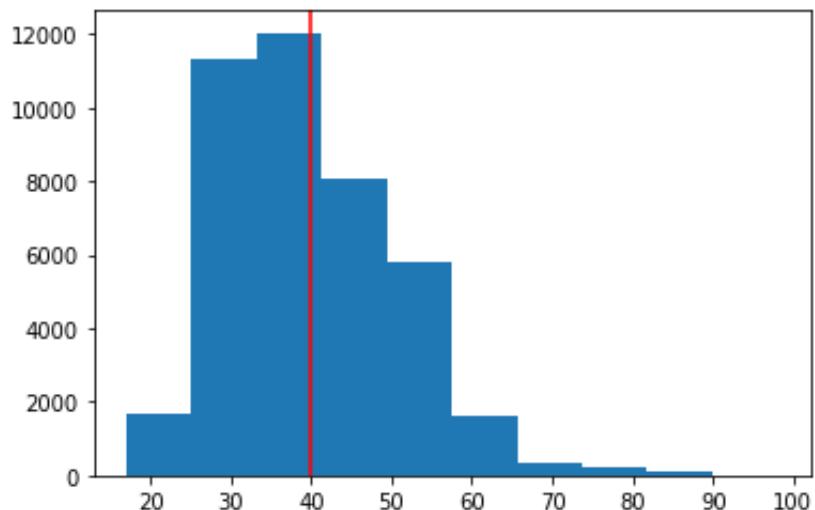
```
<đối tượng>.<hàm>
```

- Hàm .quantile(0.5) chính là

Xem phân bố độ tuổi và đường trung bình

```
import matplotlib.pyplot as plt
plt.hist(df['age'])
plt.axvline(df['age'].mean(), color='red')

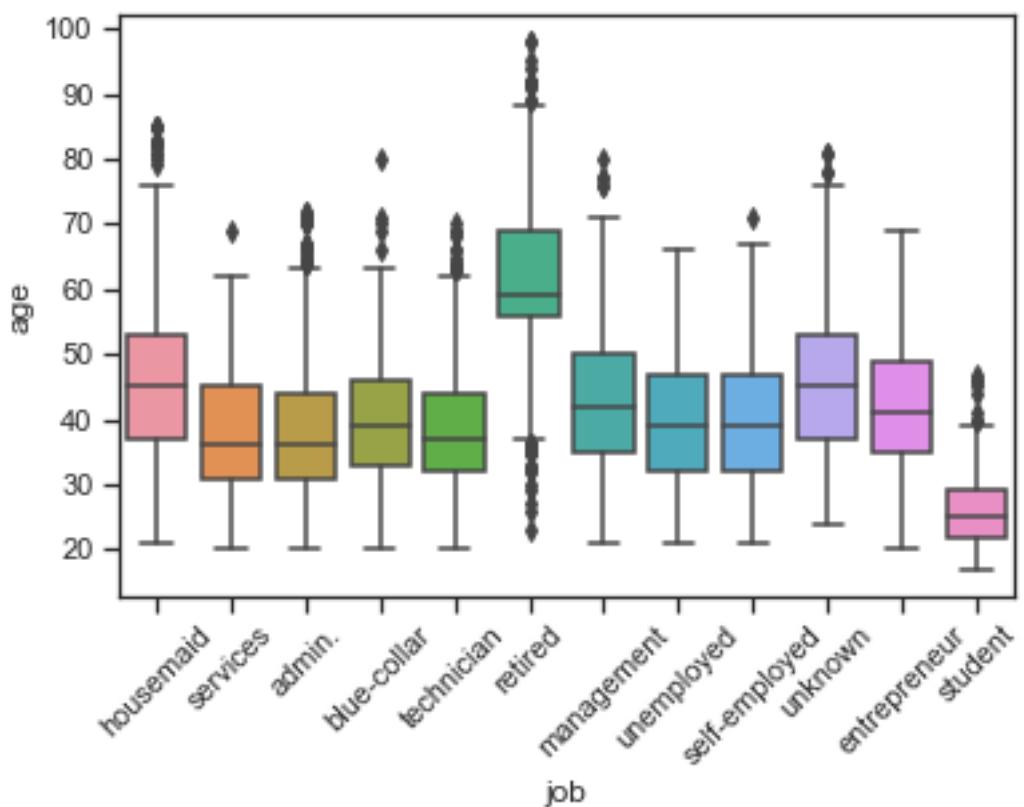
plt.show()
```



Xem phân bố tuổi theo công việc

```
import pandas as pd
import seaborn as sns

ax = sns.boxplot(x = 'job', y="age", data = df)
ax.set_xticklabels(ax.get_xticklabels(), rotation=45)
```



Quan sát dữ liệu theo ngày trong tuần

```
import pandas as pd
```

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

```
import matplotlib.pyplot as plt

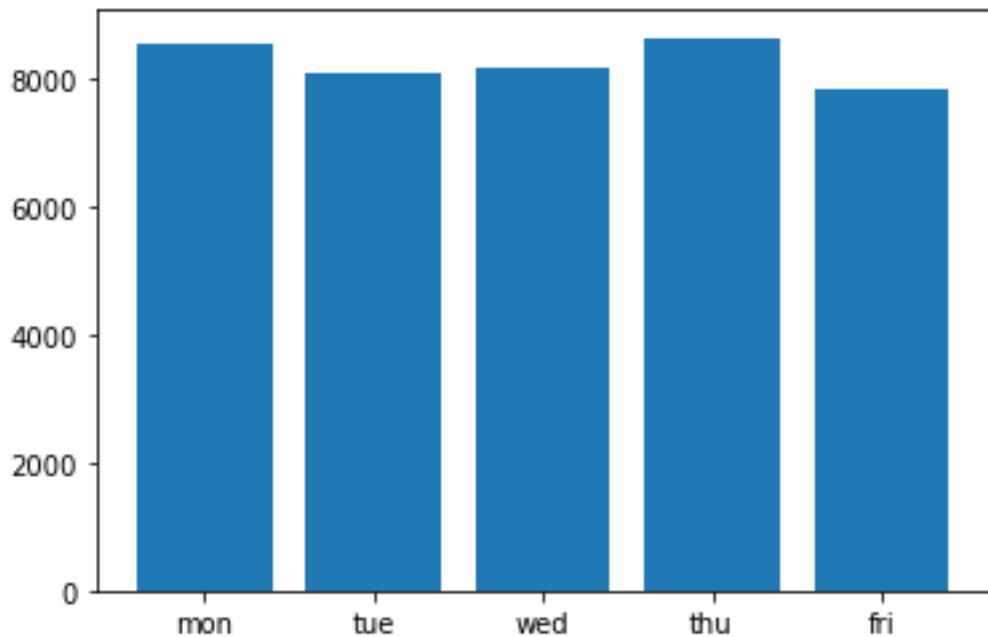
df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-
additional-full.csv', sep=';')

# Chuẩn bị danh mục ngày theo thứ tự
cats = ['mon', 'tue', 'wed', 'thu', 'fri']

# Đếm dữ liệu theo ngày và xếp thứ tự theo danh mục cats
count_dow = df['day_of_week'].value_counts().reindex(cats)
print(count_dow)
plt.bar(count_dow.index, count_dow)

plt.show()
```

```
mon    8514
tue    8090
wed    8134
thu    8623
fri    7827
Name: day_of_week, dtype: int64
```



Quan sát dữ liệu theo tháng

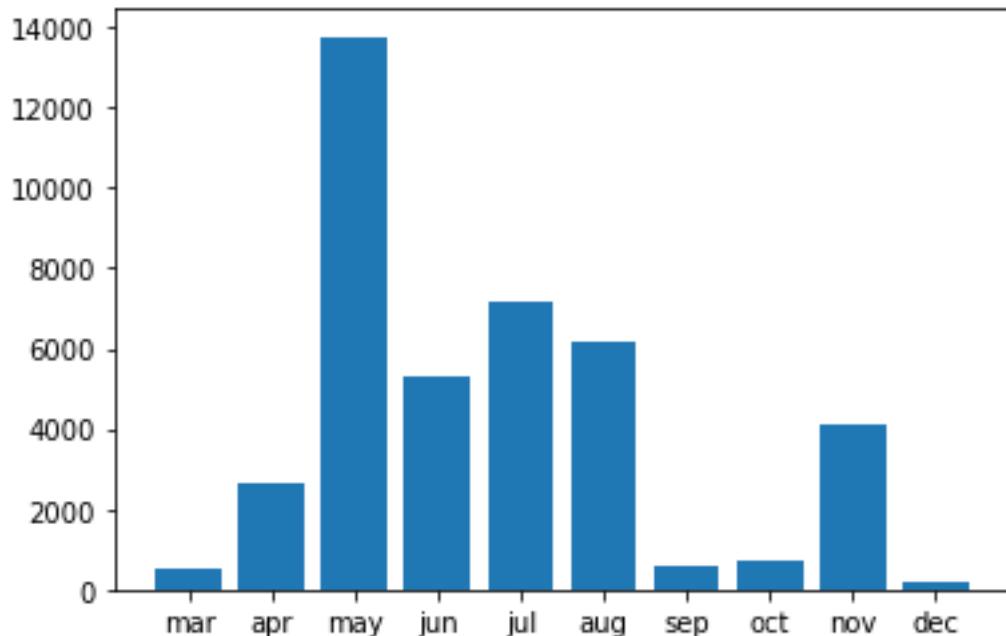
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-
additional-full.csv', sep=';')
```

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

```
df.columns  
# Chuẩn bị danh mục tháng theo thứ tự  
cats = ['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep',  
'oct', 'nov', 'dec']  
  
# Đếm dữ liệu theo tháng và xếp thứ tự theo danh mục cats  
count_month = df['month'].value_counts().reindex(cats)  
print(count_month)  
plt.bar(count_month.index, count_month)  
  
plt.show()
```

```
jan      NaN  
feb      NaN  
mar     546.0  
apr    2632.0  
may   13769.0  
jun    5318.0  
jul    7174.0  
aug    6178.0  
sep    570.0  
oct    718.0  
nov   4101.0  
dec    182.0  
Name: month, dtype: float64
```



Tính tỉ lệ chuyển đổi – conversion rate

Trong kinh doanh, tỉ lệ chuyển đổi là tỉ lệ người trở thành khách hàng trên tổng số người mà công ty tiếp cận. Trong trường hợp này là **tỉ lệ số người đồng ý sử dụng dịch vụ trên tổng số người được gọi điện**.

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

Để tính tổng số người được gọi điện (tương ứng với số dòng dữ liệu trong dataset) thì dùng hàm `shape` của data frame:

```
df.shape
```

```
(41188, 22)
```

Hàm `shape` trả lại bộ (tuple) 2 giá trị gồm số dòng và số cột của data frame. Phần tử của tuple được đánh số từ 0. Để lấy ra số dòng thì lấy phần tử thứ nhất của tuple bằng cách dùng dấu ngoặc vuông như sau:

```
df.shape[0]
```

```
41188
```

Để lấy ra số người đồng ý dùng dịch vụ thì chỉ cần tính tổng của cột `conversion`:

```
df['conversion'].sum()
```

```
4640
```

Tính tỉ lệ chuyển đổi như sau:

```
df['conversion'].sum() / df.shape[0] * 100
```

```
11.265417111780131
```

Nếu dùng hàm `print` để hiển thị ra thông báo và làm tròn 2 số phần thập phân thì lệnh và kết quả như sau:

```
print("Tỉ lệ chuyển đổi: %0.2f%%" % (df['conversion'].sum() / df.shape[0] * 100))
```

```
Tỉ lệ chuyển đổi: 11.27%
```

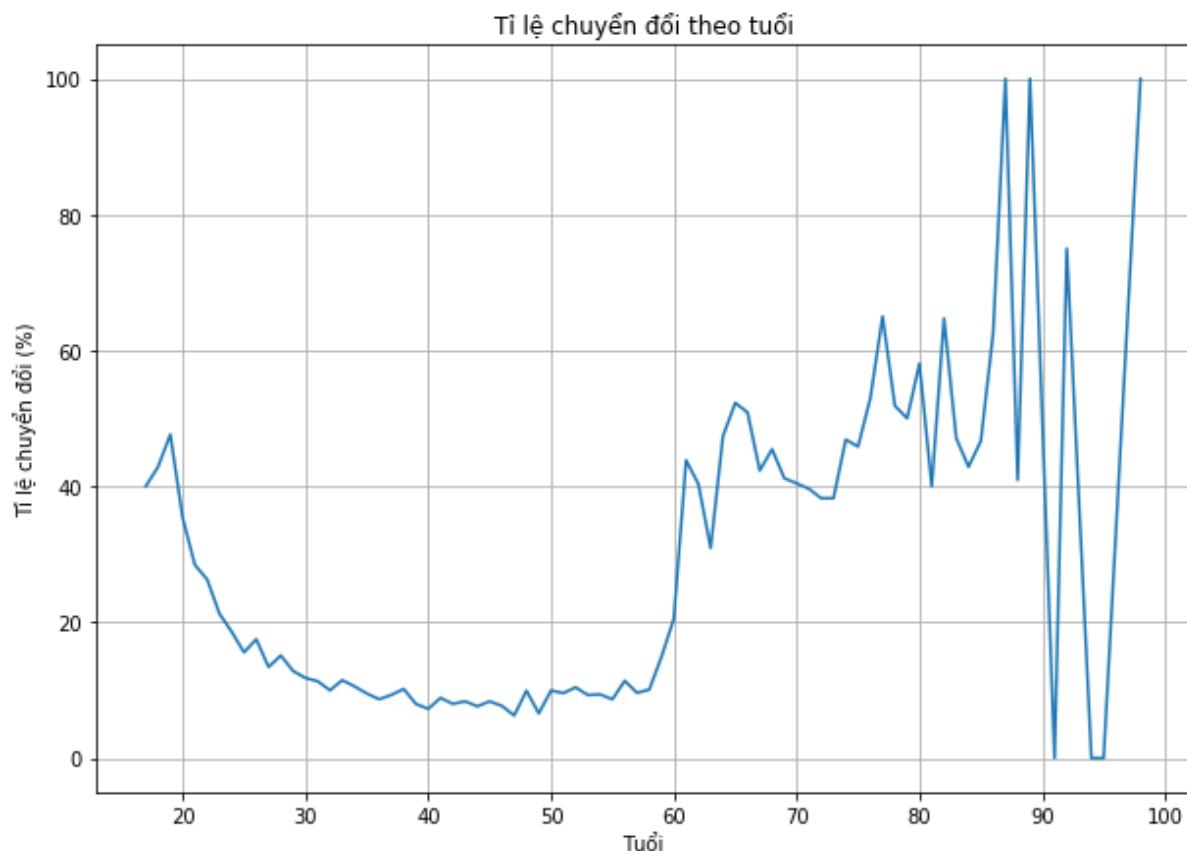
Conversion rate theo độ tuổi

Tương tự cách `group_by` và tính `sum`, `tính count ở trên, đoạn code sau sẽ tính tỉ lệ chuyển đổi theo tuổi và vẽ biểu đồ như sau:`

```
conversions_by_age = df.groupby(by='age')['conversion'].sum() / df.groupby(by='age')['conversion'].count() * 100.0
ax = conversions_by_age.plot(
    grid=True,
    figsize=(10, 7),
    title='Tỉ lệ chuyển đổi theo tuổi'
)
```

```
ax.set_xlabel('Tuổi')
ax.set_ylabel('Tỉ lệ chuyển đổi (%)')

plt.show()
```



Conversion rate theo số lần liên lạc

Gom nhóm theo cột campaign rồi tính tổng theo cột conversion:

```
sum_conversion_by_campaign =
df.groupby(by='campaign') ['conversion'].sum()
```

Giá trị của cột campaign là số lần liên lạc (bao gồm gọi điện đến số di động và số bàn).

Gom nhóm theo cột campaign rồi đếm tổng số dòng dữ liệu theo cột conversion:

```
count_conversion_by_campaign =
df.groupby(by='campaign') ['conversion'].count()
```

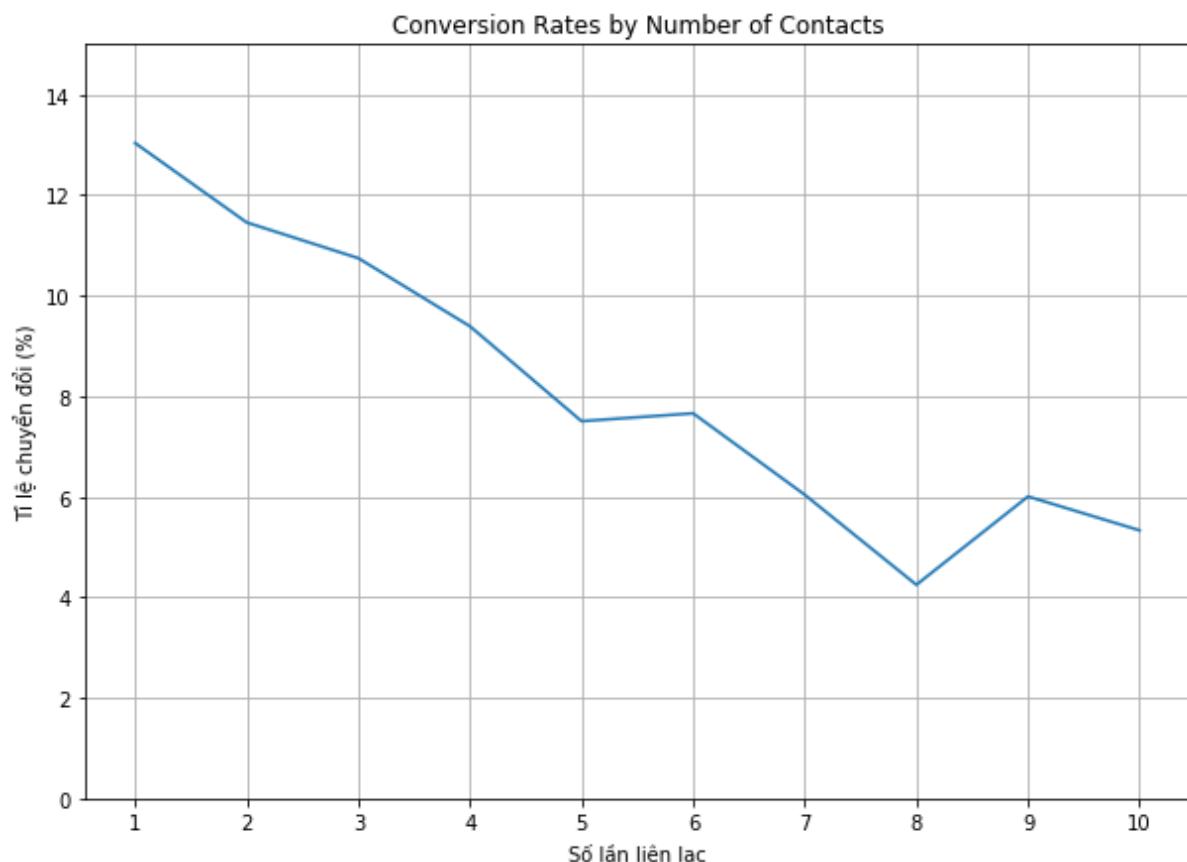
Tính tỉ lệ % từ hai số trên:

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

```
conversions_by_contacts = sum_conversion_by_campaign /  
count_conversion_by_campaign * 100.0
```

Xem biểu đồ:

```
import matplotlib.pyplot as plt  
  
ax = conversions_by_contacts[:10].plot(  
    grid=True,  
    figsize=(10, 7),  
    xticks=conversions_by_contacts.index[:10],  
    title='Tỉ lệ chuyển đổi theo số lần liên lạc'  
)  
  
ax.set_ylim([0, 15])  
ax.set_xlabel('Số lần liên lạc')  
ax.set_ylabel('Tỉ lệ chuyển đổi (%)')  
plt.show()
```



Conversion rate theo nhóm tuổi

Đầu tiên tạo thêm biến phân nhóm age_group:

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

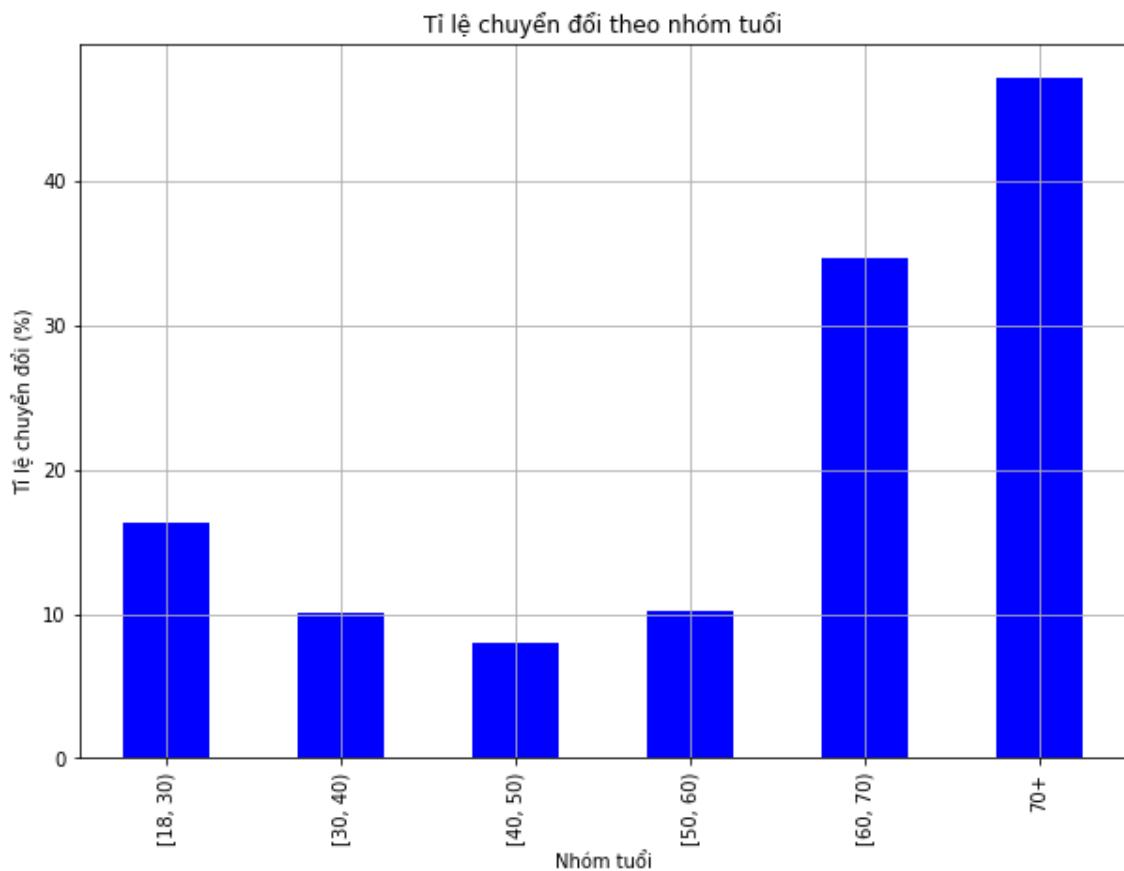
```
df['age_group'] = df['age'].apply(  
    lambda x: '[18, 30)' if x < 30 else '[30, 40)' if x < 40 \  
        else '[40, 50)' if x < 50 else '[50, 60)' if x < 60 \  
        else '[60, 70)' if x < 70 else '70+'  
)
```

Chú ý dấu \ ở cuối mỗi dòng ý nói chưa kết thúc lệnh.

Tiếp theo tổng hợp số liệu và vẽ biểu đồ:

```
conversions_by_age_group =  
df.groupby(by='age_group')['conversion'].sum() / df.groupby(  
    by='age_group'  
)['conversion'].count() * 100.0  
  
ax = conversions_by_age_group.loc[  
    ['[18, 30)', '[30, 40)', '[40, 50)', '[50, 60)', '[60, 70)',  
    '70+']  
].plot(  
    kind='bar',  
    color='blue',  
    grid=True,  
    figsize=(10, 7),  
    title='Tỉ lệ chuyển đổi theo nhóm tuổi'  
)  
  
ax.set_xlabel('Nhóm tuổi')  
ax.set_ylabel('Tỉ lệ chuyển đổi (%)')  
  
plt.show()
```

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



Khai thác thêm về Group By

Trong phần phân tích dữ liệu Bank Marketing ở trên thì bạn đã trải nghiệm qua hàm Group By của thư viện Pandas. Phần này sẽ khai thác sâu thêm về Group By, hy vọng mang lại nhiều điều hứng thú và bổ ích.

Phân tích Vụ đắm tàu Titanic với Python

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

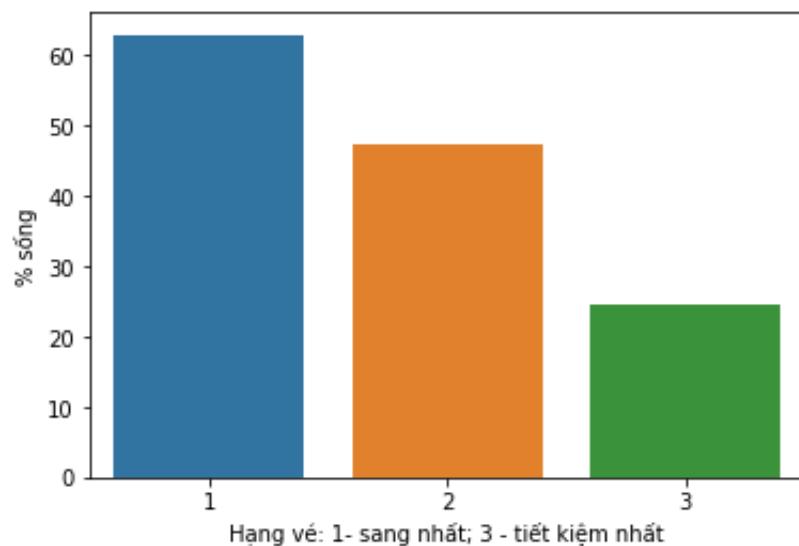
df =
pd.read_csv('https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv')
df.head()

rateSurvived_by_Pclass = df.groupby(by='Pclass')['Survived'].sum() /
df.groupby(
    by='Pclass'
```

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

```
) ['Survived'].count() * 100.0\n\nrateSurvived_by_Pclass\n\n\nplt.figure().suptitle('Biểu đồ tỉ lệ sống sót theo hạng vé trong vụ\ndắm tàu Titanic')\nsns.barplot([1, 2, 3], rateSurvived_by_Pclass)\nplt.xlabel('Hạng vé: 1- sang nhất; 3 - tiết kiệm nhất')\nplt.ylabel('% sống')
```

Biểu đồ tỉ lệ sống sót theo hạng vé trong vụ đắm tàu Titanic



Bài 13: Phân tích dữ liệu Marketing #2

Bài này giả định là bạn đọc đã quen với các kỹ thuật sau:

- Sử dụng thư viện pandas trong Python để chuyển đổi dữ liệu (transform data) nói chung, các thuộc tính (attributes) từ nguồn dữ liệu thô thành các thuộc tính và giá trị như bạn mong muốn.
- Biết cách tổ chức dữ liệu thành dạng bản (tabular form) với các trường dữ liệu đúng và sạch (correct and clean).

Trên cơ sở kiến thức, kỹ thuật đó thì bài này giúp bạn trải nghiệm quá trình phân tích dữ liệu để giải quyết các vấn đề đặt ra cho doanh nghiệp (business problem⁹). Để đảm bảo vấn đề được giải quyết, tìm được kết quả tốt nhất (the best outcomes) thì chúng ta cần thấu hiểu về dữ liệu, các câu hỏi liên quan đến dữ liệu và đặc biệt xác định rõ vấn đề nào có thể giải quyết từ nguồn dữ liệu này.

Để không mất thời gian thì chúng ta vào ngay qui trình đầu tiên nhé – qui trình Khám phá dữ liệu (Exploratory Data Analysis). Minh họa trong bài là này một dữ liệu về Marketing. Chúng ta sẽ trải nghiệm các kỹ thuật để khám phá và phân tích dữ liệu bằng cách giải quyết một số vấn đề cho doanh nghiệp như:

- Xác định các thuộc tính hữu ích cho marketing.
- Phân tích các chỉ số năng suất trọng yếu (KPI – Key Performance Indicator)
- Thực hiện các phân tích so sánh (Comparative analysis).
- Phân tích để tạo ra các báo cáo để thấu hiểu và trực quan hóa dữ liệu (insights and visualizations)

Về kỹ thuật thì sẽ khai thác các thư viện sau trong Python:

- Pandas
- Matplotlib
- Seaborn

Về phần mềm thì sẽ sử dụng Anaconda Spyder.

Xem lại các bài trước hoặc tham khảo ở đây <https://ThachLN.github.io>, nội dung ngày 2.

⁹ Business problem nên hiểu rộng ra một chút là không chỉ dùng lại cho Doanh nghiệp. Các nghiên cứu khoa học, các nghiên cứu xã hội nói chung là cũng hướng đến một mục tiêu, một kết quả, một giá trị nào đó. Nói chung là khi làm một việc có giá trị nào đó trong ngắn hạn, hoặc dài hạn thì gọi là Business.

Xác định đúng các thuộc tính (Identifying the Right Attributes)

Khi bạn nhận được bộ dữ liệu marketing đã được cấu trúc hóa một cách chính chu, việc đầu tiên cần làm là thấu hiểu chúng¹⁰. Để hiểu qua về dữ liệu (giả định là bạn đã chuẩn bị được dưới dạng DataFrame trong Python) thì có thể sử dụng các hàm sau của Pandas:

Các hàm/thuộc tính cho DataFrame:

Hàm/Thuộc tính	Chức năng
info()	
describe()	
columns	
head(n)	
tail(n)	
groupby(col)[cols].agg_func	

Các hàm/thuộc tính áp dụng cho cột dữ liệu (gọi là Series):

Hàm/Thuộc tính	Chức năng
unique()	
count()	
min()	
max()	
mean()	
median()	
mode()	
quantile(x)	

Để bạn quen tay thì tôi dùng dữ liệu mẫu ở đây:

<https://thachln.github.io/datasets/sales.csv>

Đọc dữ liệu từ Excel vào DataFrame

```
import pandas as pd  
df = pd.read_csv('https://thachln.github.io/datasets/sales.csv')
```

¹⁰ Đôi lúc cần phải học, tham vấn về chuyên gia trong lĩnh vực mà mình đang phân tích. Cụ thể ở đây là cần hỏi các chuyên gia về Marketing để hiểu các khái niệm, công việc trong chuyên ngành.

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

```
df.head()
```

	Year	Product line	Product type	Product	Order method type	Retailer country	Revenue	Planned revenue	Product cost
0	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United States	5819.7	6586.16	1733
1	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United Kingdom	nan	nan	nan
2	2005	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United States	10904.3	11363.5	2990
3	2005	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United Kingdom	27987.8	28855.7	7593
4	2006	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United States	nan	nan	nan
5	2006	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United Kingdom	8750.77	8926.96	2349
6	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Canada	nan	nan	nan
7	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Mexico	13497.6	14598.1	3841
8	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Brazil	nan	nan	nan
9	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Japan	nan	nan	nan
10	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Korea	8433.46	8788.64	2312
11	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	China	nan	nan	nan
12	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Singapore	7422.67	7735.28	2035
13	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Australia	nan	nan	nan
14	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Netherlands	nan	nan	nan
15	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Sweden	nan	nan	nan

Dùng chức năng xem biến df trong Spyder để nhìn qua toàn bộ DataFrame:

Index	Year	Product line	Product type	Product	Order method type	Retailer country	Revenue	Planned revenue	Product cost
0	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United States	5819.7	6586.16	1733
1	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United Kingdom	nan	nan	nan
2	2005	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United States	10904.3	11363.5	2990
3	2005	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United Kingdom	27987.8	28855.7	7593
4	2006	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United States	nan	nan	nan
5	2006	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United Kingdom	8750.77	8926.96	2349
6	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Canada	nan	nan	nan
7	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Mexico	13497.6	14598.1	3841
8	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Brazil	nan	nan	nan
9	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Japan	nan	nan	nan
10	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Korea	8433.46	8788.64	2312
11	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	China	nan	nan	nan
12	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Singapore	7422.67	7735.28	2035
13	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Australia	nan	nan	nan
14	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Netherlands	nan	nan	nan
15	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	Sweden	nan	nan	nan

Xem các cột của dữ liệu qua thuộc tính columns

```
df.columns
```

```
Index(['Year', 'Product line', 'Product type', 'Product', 'Order method type',
       'Retailer country', 'Revenue', 'Planned revenue', 'Product cost',
       'Quantity', 'Unit cost', 'Unit price', 'Gross profit',
       'Unit sale price'],
      dtype='object')
```

Xem tổng quan về thông tin dữ liệu qua lệnh info()

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17823 entries, 0 to 17822
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   Year            17823 non-null   int64  
 1   Product line    17823 non-null   object  
 2   Product type    17823 non-null   object  
 3   Product          17823 non-null   object  
 4   Order method type 17823 non-null   object  
 5   Retailer country 17823 non-null   object  

```

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

```
6    Revenue           6045 non-null   float64
7  Planned revenue    6045 non-null   float64
8  Product cost       6045 non-null   float64
9   Quantity          5860 non-null   float64
10  Unit cost          6045 non-null   float64
11  Unit price         6045 non-null   float64
12  Gross profit       6045 non-null   float64
13  Unit sale price    6045 non-null   float64
dtypes: float64(8), int64(1), object(5)
memory usage: 1.9+ MB
```

Xem tổng quan mô tả về dữ liệu bằng hàm .describe()

```
df_desc = df.describe()
print(df_desc.to_string())
```

	Year	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Unit sale p
rice count	17843.000000	6.056000e+03	6.056000e+03	6.056000e+03	5869.000000	6056.000000	6056.000000	6.056000e+03	6056.00
mean	2005.168694	1.037380e+05	1.057855e+05	5.695909e+04	4685.638023	58.847932	48.954047	4.332107e+04	44.84
std	0.962363	1.834570e+05	1.879756e+05	1.116934e+05	8945.261554	348.055978	62.809969	7.180056e+04	58.39
min	2004.000000	0.000000e+00	0.000000e+00	3.360000e+01	5.000000	0.850000	3.660000	-1.336560e+04	0.00
25%	2004.000000	1.367160e+04	1.384909e+04	5.761365e+03	625.000000	2.760000	7.000000	7.013587e+03	6.58
50%	2005.000000	4.158951e+04	4.194593e+04	1.908848e+04	1694.000000	9.000000	18.000000	1.895358e+04	17.66
75%	2006.000000	1.118490e+05	1.142314e+05	5.783603e+04	4843.000000	34.970000	66.770000	4.999680e+04	62.76
max	2009.000000	3.644349e+06	3.477910e+06	2.061750e+06	164142.000000	7833.000000	265.140000	1.416160e+06	265.14

df_desc - DataFrame									
Index	Year	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Unit sale price
count	17843	6056	6056	6056	5869	6056	6056	6056	6056
mean	2005.17	103738	105785	56959.1	4685.64	58.8479	48.954	43321.1	44.8482
std	0.962363	183457	187976	111693	8945.26	348.056	62.81	71800.6	58.398
min	2004	0	0	33.6	5	0.85	3.66	-13365.6	0
25%	2004	13671.6	13849.1	5761.36	625	2.76	7	7013.59	6.58
50%	2005	41589.5	41945.9	19088.5	1694	9	18	18953.6	17.6641
75%	2006	111849	114231	57836	4843	34.97	66.77	49996.8	62.76
max	2009	3.64435e+06	3.477910e+06	2.061750e+06	164142	7833	265.14	1.416160e+06	265.14

Lệnh describe() cho phép bạn xem nhanh các giá trị count, mean, std, min, bách phân vị 25%, 50%, 75%, max của từng cột trong DataFrame.

Xem các giá trị Bách phân vị

```
df_quantile = df.quantile([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])
```

```
print(df_quantile.to_string())
```

	Year	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Unit sale price
0.1	2004.0	4501.280	4704.000	1693.950	259.8	1.880000	6.00	2503.620	5.490000
0.2	2004.0	10186.540	10386.000	3993.600	495.6	2.420000	6.59	5346.600	6.000000
0.3	2004.0	18015.600	18222.500	7544.435	779.2	2.800000	8.00	8739.870	7.000000
0.4	2005.0	28532.760	28797.220	12474.000	1147.4	5.600000	12.81	13396.050	11.309091
0.5	2005.0	41589.515	41945.925	19088.480	1694.0	9.000000	18.00	18953.585	17.664083
0.6	2005.0	59410.560	61320.000	29118.000	2561.8	15.930000	35.00	27096.630	24.390000
0.7	2006.0	89618.070	90968.730	44680.965	3886.4	33.303333	54.93	40553.015	51.400000
0.8	2006.0	144943.200	146112.330	74850.000	6308.8	60.000000	90.09	62426.210	83.780000
0.9	2006.0	262016.030	265156.535	145916.050	11517.6	79.560000	129.72	107349.570	123.230000
1.0	2009.0	3644349.300	3477909.780	2061750.000	164142.0	7833.000000	265.14	1416159.780	265.140000

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

Index	Year	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Net sale price
0.1	2004	4501.28	4704	1693.95	259.8	1.88	6	2503.62	5.49
0.2	2004	10186.5	10386	3993.6	495.6	2.42	6.59	5346.6	6
0.3	2004	18015.6	18222.5	7544.43	779.2	2.8	8	8739.87	7
0.4	2005	28532.8	28797.2	12474	1147.4	5.6	12.81	13396	11.3091
0.5	2005	41589.5	41945.9	19088.5	1694	9	18	18953.6	17.6641
0.6	2005	59410.6	61320	29118	2561.8	15.93	35	27096.6	24.39
0.7	2006	89618.1	90968.7	44681	3886.4	33.3033	54.93	40553	51.4
0.8	2006	144943	146112	74850	6308.8	60	90.09	62426.2	83.78
0.9	2006	262016	265157	145916	11517.6	79.56	129.72	107350	123.23
1.0	2009	3.64435e+06	3.47791e+06	2.06175e+06	164142	7833	265.14	1.41616e+06	265.14

Xem thử các năm đang có trong dữ liệu

```
df['Year'].unique()  
array([2004, 2005, 2006, 2007], dtype=int64)
```

Xem thử các dòng sản phẩm

```
df['Product line'].unique()  
array(['Golf Equipment', 'Camping Equipment', 'Outdoor Protection',  
       'Mountaineering Equipment'], dtype=object)
```

Xem thử các loại sản phẩm

```
df['Product type'].unique()  
array(['Golf Accessories', 'Sleeping Bags', 'Cooking Gear', 'First Aid',  
       'Insect Repellents', 'Climbing Accessories'], dtype=object)
```

Xem các sản phẩm

```
df['Product'].unique()  
array(['Course Pro Golf and Tee Set', 'Hibernator Self - Inflating Mat',  
       'TrailChef Deluxe Cook Set', 'Deluxe Family Relief Kit',  
       'Course Pro Golf Bag', 'TrailChef Water Bag',  
       'TrailChef Kitchen Kit', 'TrailChef Cook Set',  
       'TrailChef Single Flame', 'TrailChef Double Flame',  
       'Hibernator Camp Cot', 'BugShield Lotion Lite',  
       'Compact Relief Kit', 'Insect Bite Relief', 'Course Pro Umbrella',  
       'Course Pro Gloves', 'Firefly Climbing Lamp',  
       'Firefly Rechargeable Battery', 'Granite Chalk Bag',  
       'TrailChef Canteen', 'TrailChef Cup', 'TrailChef Kettle',  
       'TrailChef Utensils', 'Hibernator Lite', 'Hibernator Extreme',  
       'Hibernator Pad', 'Hibernator Pillow', 'BugShield Natural',  
       'BugShield Spray', 'BugShield Lotion', 'BugShield Extreme',  
       'Calamine Relief', 'Aloe Relief', 'Granite Carabiner',  
       'Granite Belay', 'Granite Pulley', 'Firefly Charger', 'Hibernator'],
```

```
    dtype=object)
```

Xem các kiểu đặt hàng (kênh bán hàng)

```
df['Order method type'].unique()  
array(['Sales visit', 'Telephone', 'Web', 'Special', 'Mail', 'E-mail',  
       'Fax'], dtype=object)
```

Xem các quốc gia

```
df['Retailer country'].unique()  
array(['United States', 'United Kingdom', 'Canada', 'Mexico', 'Brazil',  
       'Japan', 'Korea', 'China', 'Singapore', 'Australia', 'Netherlands',  
       'Sweden', 'Finland', 'Denmark', 'France', 'Germany', 'Belgium',  
       'Switzerland', 'Austria', 'Italy', 'Spain'], dtype=object)
```

Tự viết hàm để lọc các unique values

Phản trước bạn đã tự gọi hàm unique() cho từ Series của mỗi cột. Dưới đây tôi viết một hàm tổng quát để tạo ra một DataFrame mới mà chỉ chứa các unique values trong mỗi cột.

```
def unique_columns(df, end_col_name = None):  
    u_df = pd.DataFrame()  
  
    for col in df.columns:  
        a_df = pd.DataFrame(df[col].unique(), columns=[col])  
        u_df = pd.concat([u_df, a_df], axis=1)  
  
        if col == end_col_name:  
            break;  
  
    return u_df
```

Cách sử dụng như sau:

```
unique_df = unique_columns(df)
```

Lệnh trên sẽ tự động duyệt hết tất cả các cột trong DataFrame df. Vì thế khi xem biến kết quả unique_df bạn sẽ thấy đủ các cột chữ số như Revenue,... như bên dưới.

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

Index	Year	Product line	Product type	Product	Order method typ	Retailer country	Revenue
0	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United States	5819.7
1	2005	Camping Equipment	Sleeping Bags	Hibernator Self - Inflating Mat	Telephone	United Kingdom	nan
2	2006	Outdoor Protection	Cooking Gear	TrailChef Deluxe Cook Set	Web	Canada	10904.3
3	2007	Mountaineering Equipment	First Aid	Deluxe Family Relief Kit	Special	Mexico	27987.8
4	nan	nan	Insect Repellents	Course Pro Golf Bag	Mail	Brazil	8750.77
5	nan	nan	Climbing Accessories	TrailChef Water Bag	E-mail	Japan	13497.6
6	nan	nan	nan	TrailChef Kitchen Kit	Fax	Korea	8433.46
7	nan	nan	nan	TrailChef Cook Set	nan	China	7422.67
8	nan	nan	nan	TrailChef Single Flame	nan	Singapore	21185.8
9	nan	nan	nan	TrailChef Double Flame	nan	Australia	6013.69
10	nan	nan	nan	Hibernator Camp Cot	nan	Netherlands	20542
11	nan	nan	nan	BugShield Lotion Lite	nan	Sweden	3962.88
12	nan	nan	nan	Compact Relief Kit	nan	Finland	0
13	nan	nan	nan	Insect Bite Relief	nan	Denmark	30444
14	nan	nan	nan	Course Pro Umbrella	nan	France	21183.3
15	nan	nan	nan	Course Pro Gloves	nan	Germany	16307.5

Trường hợp bạn không muốn DataFrame trả về không chứa các cột số bên phải như Revenue,... thì bạn thêm thông số thứ hai là tên của cột mà bạn muốn tổng hợp cuối cùng (tính từ bên trái sang) như sau:

```
unique_df = unique_columns(df, 'Retailer country')
```

Xem lại biến unique_df:

Index	Year	Product line	Product type	Product	Order method typ	Retailer country	
0	2004	Golf Equipment	Golf Accessories	Course Pro Golf and Tee Set	Sales visit	United States	
1	2005	Camping Equipment	Sleeping Bags	Hibernator Self - Inflating Mat	Telephone	United Kingdom	
2	2006	Outdoor Protection	Cooking Gear	TrailChef Deluxe Cook Set	Web	Canada	
3	2007	Mountaineering Equipment	First Aid	Deluxe Family Relief Kit	Special	Mexico	
4	nan	nan	Insect Repellents	Course Pro Golf Bag	Mail	Brazil	
5	nan	nan	Climbing Accessories	TrailChef Water Bag	E-mail	Japan	
6	nan	nan	nan	TrailChef Kitchen Kit	Fax	Korea	
7	nan	nan	nan	TrailChef Cook Set	nan	China	
8	nan	nan	nan	TrailChef Single Flame	nan	Singapore	
9	nan	nan	nan	TrailChef Double Flame	nan	Australia	
10	nan	nan	nan	Hibernator Camp Cot	nan	Netherlands	
11	nan	nan	nan	BugShield Lotion Lite	nan	Sweden	
12	nan	nan	nan	Compact Relief Kit	nan	Finland	
13	nan	nan	nan	Insect Bite Relief	nan	Denmark	
14	nan	nan	nan	Course Pro Umbrella	nan	France	
15	nan	nan	nan	Course Pro Gloves	nan	Germany	

Đếm số dòng dữ liệu theo giá trị của một cột bằng hàm value_count()

Xem số dòng dữ liệu theo giá trị của năm:

```
df['Year'].value_counts()
```

2006	5451
2005	5451
2004	5451
2007	1470
Name: Year, dtype: int64	

Đếm số dòng dữ liệu theo dòng sản phẩm

```
df['Product line'].value_counts()
```

Camping Equipment	8580
Outdoor Protection	4410
Mountaineering Equipment	3087
Golf Equipment	1766
Name: Product line, dtype: int64	

Đếm số dòng dữ liệu theo loại sản phẩm

```
df['Product type'].value_counts()
```

Cooking Gear	5892
Climbing Accessories	3087
Sleeping Bags	2686
Insect Repellents	2205
First Aid	2205
Golf Accessories	1770
Name: Product type, dtype: int64	

Đếm số dòng dữ liệu theo kênh đặt hàng

```
df['Order method type'].value_counts()
```

Special	2554
Telephone	2552
web	2551
Fax	2549
E-mail	2547
Mail	2547
Sales visit	2545
Name: Order method type, dtype: int64	

Đếm số dòng dữ liệu theo quốc gia bán lẻ

```
df['Retailer country'].value_counts()
```

United States	865
United Kingdom	865
Singapore	847
Sweden	847
Italy	847
Germany	847
China	847
Japan	847
Mexico	847
France	847
Brazil	847

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

Austria	847
Switzerland	847
Finland	847
Australia	847
Spain	847
Netherlands	847
Belgium	847
Canada	847
Korea	847
Denmark	847
Vietnam	22

Name: Retailer country, dtype: int64

Tổng hợp số liệu

Tổng hợp số liệu bằng cách gom nhóm theo giá trị của một thuộc tính bằng hàm groupby.

Tổng hợp số liệu bằng groupby sum

```
df_gs = df.groupby('Retailer country')['Revenue', 'Planned revenue',
                                         'Product cost', 'Quantity', 'Gross profit'].sum()
print(df_gs.to_string())
```

Retailer country	Revenue	Planned revenue	Product cost	Quantity	Gross profit
Australia	1.526422e+07	1.552855e+07	8.367046e+06	6.494670e+05	6.384807e+06
Austria	1.631419e+07	1.663918e+07	8.923177e+06	7.190840e+05	6.871597e+06
Belgium	1.415299e+07	1.434713e+07	7.695760e+06	6.221500e+05	5.964513e+06
Brazil	1.686686e+07	1.718625e+07	9.210809e+06	7.443530e+05	7.092849e+06
Canada	3.918371e+07	3.975547e+07	2.143600e+07	1.701123e+06	1.667051e+07
China	4.350234e+07	4.432347e+07	2.392515e+07	1.935454e+06	1.800364e+07
Denmark	8.455457e+06	8.657223e+06	4.695595e+06	3.684790e+05	3.496915e+06
Finland	2.714528e+07	2.768705e+07	1.487934e+07	1.207265e+06	1.133519e+07
France	3.595367e+07	3.640336e+07	1.964643e+07	1.620252e+06	1.496895e+07
Germany	3.509449e+07	3.565769e+07	1.921340e+07	1.576459e+06	1.463705e+07
Italy	2.601864e+07	2.649849e+07	1.422387e+07	1.142868e+06	1.091007e+07
Japan	4.603330e+07	4.691096e+07	2.525632e+07	2.047615e+06	1.920845e+07
Korea	3.174933e+07	3.226426e+07	1.738811e+07	1.432122e+06	1.318978e+07
Mexico	2.660842e+07	2.720830e+07	1.455972e+07	1.173878e+06	1.117549e+07
Netherlands	2.506122e+07	2.554392e+07	1.371657e+07	1.121981e+06	1.046439e+07
Singapore	2.886032e+07	2.945750e+07	1.590164e+07	1.270886e+06	1.195985e+07
Spain	2.367628e+07	2.415171e+07	1.292956e+07	1.052358e+06	9.962396e+06
Sweden	9.718640e+06	9.804136e+06	5.258636e+06	4.329620e+05	4.079646e+06
Switzerland	1.065317e+07	1.079256e+07	5.785330e+06	4.656830e+05	4.489794e+06
United Kingdom	3.988824e+07	4.080191e+07	2.210011e+07	1.656837e+06	1.654299e+07
United States	1.075452e+08	1.104916e+08	5.956920e+07	4.549587e+06	4.471462e+07
Vietnam	4.914138e+05	5.179206e+05	2.625046e+05	9.146556e+03	2.289092e+05

Tổng hợp số liệu bằng groupby mean

```
df_gm = df.groupby('Retailer country')['Revenue', 'Planned revenue',
                                         'Product cost', 'Quantity', 'Unit cost', 'Unit price', 'Gross profit',
                                         'Unit sale price'].mean()
print(df_gm.to_string())
```

Retailer country	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Unit sale price
Australia	65794.049353	66933.410603	36064.853879	2873.747788	46.569968	46.828448	27520.718060	43.073775
Austria	63727.294570	64996.808203	34856.158633	2887.887550	53.321432	50.122734	26842.177109	45.285907
Belgium	58970.775000	59779.727417	32065.665792	2681.681034	48.276413	48.253708	24852.139042	44.445764
Brazil	106752.290633	108773.738987	58296.261646	4865.052288	79.542414	46.980759	44891.451203	43.213174
Canada	97959.284950	99388.685350	53589.993850	4395.666667	69.936654	47.781300	41676.264075	44.512998
China	192488.245442	196121.556814	105863.507566	8878.229358	110.158067	53.785133	79662.112257	49.097133
Denmark	41448.317206	42437.366471	23017.621569	1870.451777	41.523586	51.148088	17141.741373	46.517176
Finland	130506.174038	133110.827644	71535.290048	5976.559406	83.147766	47.003173	54496.092644	43.090212
France	89436.994279	90555.617189	48871.704900	4154.492308	59.163594	47.227438	37236.201219	43.252405
Germany	88399.215718	89817.861788	48396.474484	4084.090674	62.195636	50.447154	36869.137758	46.217439
Italy	85307.013082	86880.299279	46635.626656	3874.128814	60.447363	48.496754	35770.735410	44.434877
Japan	118034.096308	120284.509077	64759.799179	5416.970899	83.429044	47.312692	49252.429359	42.988547
Korea	123538.232763	125541.879144	67658.013035	5705.665339	68.223581	48.365798	51322.112646	44.200622
Mexico	96407.300906	98580.800616	52752.620761	4380.141791	54.945672	46.842319	40490.898152	42.961267
Netherlands	79559.427175	81091.815937	43544.674952	3678.626230	58.807858	49.378000	33220.274286	44.626565
Singapore	99176.345945	101228.521924	54644.816151	4522.725979	69.325948	49.999931	41099.151168	45.866685

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

Spain	79718.122424	81318.879529	43533.866364	3654.020833	61.862859	46.400303	33543.420673	42.942191
Sweden	40494.333792	40850.566583	21910.982792	1858.206009	47.705907	53.585125	16998.525167	49.215721
Switzerland	58533.873736	59299.777143	31787.526923	2616.196629	42.835353	46.807912	24669.197418	42.730779
United Kingdom	140947.836996	144205.602968	78092.249223	6046.850365	25.784710	49.579470	58455.790742	45.249123
United States	221286.423807	227349.010638	122570.362428	9700.611940	26.623756	50.446646	92005.395165	46.354537
Vietnam	44673.983381	47083.689560	23864.053906	1016.283951	39.786293	78.185099	20809.929475	74.036946

df_gm - DataFrame

Retailer country	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Unit sale price
Australia	65794	66933.4	36064.9	2873.75	46.57	46.8284	27520.7	43.0738
Austria	63727.3	64996.8	34856.2	2887.89	53.3214	50.1227	26842.2	45.2859
Belgium	58970.8	59779.7	32065.7	2681.68	48.2764	48.2537	24852.1	44.4458
Brazil	106752	108774	58296.3	4865.05	79.5424	46.9808	44891.5	43.2132
Canada	97959.3	99388.7	53590	4395.67	69.9367	47.7813	41676.3	44.513
China	192488	196122	105864	8878.23	110.158	53.7851	79662.1	49.0971
Denmark	41448.3	42437.4	23017.6	1870.45	41.5236	51.1481	17141.7	46.5172
Finland	130506	133111	71535.3	5976.56	83.1478	47.0032	54496.1	43.0902
France	89437	90555.6	48871.7	4154.49	59.1636	47.2274	37236.2	43.2524
Germany	88399.2	89817.9	48396.5	4084.09	62.1956	50.4472	36869.1	46.2174
Italy	85307	86880.3	46635.6	3874.13	60.4474	48.4968	35770.7	44.4349
Japan	118034	120285	64759.8	5416.97	83.429	47.3127	49252.4	42.9885
Korea	123538	125542	67658	5705.67	68.2236	48.3658	51322.1	44.2006
Mexico	96407.3	98580.8	52752.6	4380.14	54.9457	46.8423	40490.9	42.9613
Netherlands	79559.4	81091.8	43544.7	3678.63	58.8079	49.378	33220.3	44.6266

Format Resize Background color Column min/max Save and Close Close

Tổng hợp bằng groupby min

```
df_gmin = df.dropna().groupby('Retailer country')[['Revenue', 'Planned revenue', 'Product cost', 'Quantity', 'Unit cost', 'Unit price', 'Gross profit', 'Unit sale price']].min()

print(df_gmin.to_string())
```

Retailer country	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Unit sale price
Australia	0.00	294.00	120.78	49.0	0.85	3.66	-558.00	0.000000
Austria	0.00	0.00	33.60	5.0	0.85	3.66	-360.00	0.000000
Belgium	0.00	0.00	70.18	6.0	0.85	3.66	-280.72	0.000000
Brazil	966.00	966.00	455.63	138.0	0.85	3.66	510.37	3.192857
Canada	198.00	198.00	93.39	33.0	0.85	3.66	53.40	2.875000
China	0.00	618.00	291.49	103.0	0.85	3.66	-840.00	0.000000
Denmark	0.00	738.00	312.96	40.0	0.85	3.66	-2561.74	0.000000
Finland	486.00	486.00	223.56	81.0	0.85	3.66	262.44	3.141429
France	0.00	230.12	90.24	43.0	0.85	3.66	-190.40	0.000000
Germany	234.00	0.00	110.37	32.0	0.85	3.66	-1119.04	0.000000
Italy	0.00	330.00	154.56	55.0	0.85	3.66	-224.00	0.000000
Japan	0.00	0.00	67.20	22.0	0.85	3.66	-1225.18	0.000000
Korea	276.00	276.00	130.18	46.0	0.85	3.66	-3831.84	3.137143
Mexico	432.00	432.00	203.76	51.0	0.85	3.66	-542.00	2.615000
Netherlands	0.00	308.57	113.28	40.0	0.85	3.66	-13365.60	0.000000
Singapore	576.00	576.00	271.68	32.0	0.85	3.66	1.28	3.142857
Spain	348.00	348.00	160.08	40.0	0.85	3.66	103.68	2.200000
Sweden	0.00	0.00	67.76	23.0	0.85	3.66	-1438.56	0.000000
Switzerland	0.00	162.00	74.52	27.0	0.85	3.66	-2790.00	0.000000
United Kingdom	0.00	264.00	124.52	44.0	0.85	3.66	-4009.68	0.000000
United States	0.00	156.90	57.60	24.0	0.85	3.66	-186.00	0.000000
Vietnam	7422.67	7735.28	2035.60	508.0	2.80	10.64	5387.07	10.210000

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

df_gmin - DataFrame

Retailer country	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Unit sale price
Australia	0	294	120.78	49	0.85	3.66	-558	0
Austria	0	0	33.6	5	0.85	3.66	-360	0
Belgium	0	0	70.18	6	0.85	3.66	-280.72	0
Brazil	966	966	455.63	138	0.85	3.66	510.37	3.19286
Canada	198	198	93.39	33	0.85	3.66	53.4	2.875
China	0	618	291.49	103	0.85	3.66	-840	0
Denmark	0	738	312.96	40	0.85	3.66	-2561.74	0
Finland	486	486	223.56	81	0.85	3.66	262.44	3.14143
France	0	230.12	90.24	43	0.85	3.66	-190.4	0
Germany	234	0	110.37	32	0.85	3.66	-1119.04	0
Italy	0	330	154.56	55	0.85	3.66	-224	0
Japan	0	0	67.2	22	0.85	3.66	-1225.18	0
Korea	276	276	130.18	46	0.85	3.66	-3831.84	3.13714
Mexico	432	432	203.76	51	0.85	3.66	-542	2.615
Netherlands	0	308.57	113.28	40	0.85	3.66	-13365.6	0
Singapore	576	576	271.68	32	0.85	3.66	1.28	3.14286

Format Resize Background color Column min/max Save and Close Close

Tổng hợp số liệu theo năm

```
df_gyearsum = df.groupby('Year')[['Revenue', 'Planned revenue', 'Product cost', 'Quantity', 'Unit cost', 'Unit price', 'Gross profit', 'Unit sale price']].sum()

print(df_gyearsum.to_string())
```

Year	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Unit sale price
2004	1.528977e+08	1.567331e+08	8.538058e+07	7.318558e+06	97750.174438	92781.460000	6.248213e+07	84841.786496
2005	1.908502e+08	1.947044e+08	1.029861e+08	8.453776e+06	113381.098628	103147.420000	8.112626e+07	93071.003707
2006	2.228721e+08	2.270020e+08	1.209524e+08	8.786835e+06	134772.241249	82126.840000	9.556398e+07	76786.074337
2007	6.112591e+07	6.167953e+07	3.536268e+07	2.931694e+06	10041.912620	17549.950000	2.295112e+07	16087.343791
2008	1.730949e+05	1.824672e+05	9.168415e+04	2.989000e+03	153.350000	304.030000	8.141075e+04	287.850000
2009	3.183189e+05	3.354533e+05	1.708204e+05	6.157556e+03	284.299219	556.006094	1.474985e+05	526.556406

df_gyearsum - DataFrame

Year	Revenue	Planned revenue	Product cost	Quantity	Unit cost	Unit price	Gross profit	Unit sale price
2004	1.52898e+08	1.56733e+08	8.53806e+07	7.31856e+06	97750.2	92781.5	6.24821e+07	84841.8
2005	1.9085e+08	1.94704e+08	1.02986e+08	8.45378e+06	113381	103147	8.11263e+07	93071
2006	2.22872e+08	2.27002e+08	1.20952e+08	8.78684e+06	134772	82126.8	9.5564e+07	76786.1
2007	6.11259e+07	6.16795e+07	3.53627e+07	2.93169e+06	10041.9	17550	2.29511e+07	16087.3
2008	173095	182467	91684.1	2989	153.35	304.03	81410.8	287.85
2009	318319	335453	170820	6157.56	284.299	556.006	147498	526.556

Format Resize Background color Column min/max Save and Close Close

Bài 14: So sánh 2 tỉ lệ

Phần này đề cập một chút lý thuyết khi gặp vấn đề cần so sánh như sau:

- (1) Một hãng dược D tạo ra một thuốc mới cần thử nghiệm trên một nhóm bệnh nhân. Sau đó chia ngẫu nhiên thành hai nhóm, nhóm A có dùng thuốc, nhóm B dùng giả dược. Theo dõi trong một thời gian rồi so sánh kết quả của 2 nhóm. Câu hỏi đặt ra là kết quả của 2 nhóm có sự khác biệt không?
 - (2) Một nhóm nhà giáo dục cần đánh giá chất lượng hai bộ sách giáo khoa thì họ làm tương tự như hãng dược D ở trên. Chọn ngẫu nhiên hai nhóm học sinh. Nhóm A học bộ sách thứ nhất. Nhóm B học bộ sách thứ hai. Kết thúc khóa học thì làm bài kiểm tra và phỏng vấn từng học sinh. Câu hỏi đặt ra tương tự là chất lượng học sinh giữa hai nhóm có sự khác biệt không? Từ đó suy luận ngược ra chất lượng của sách giáo khoa với giả định các yếu tố khác giữa hai nhóm là như sau.
 - (3) Một hãng bia cần đánh giá khẩu vị lựa chọn của khách hàng bèn làm cuộc khảo sát từ hai nhóm khách hàng ngẫu nhiên. Sau đó tổng hợp ý kiến của khách hàng trong hai nhóm. Câu hỏi đặt ra là kết quả đánh giá chất lượng hai loại bia có sự khác biệt không?
- Tình huống này chính tôi từng là khách hàng khi một bạn nhân viên của công ty nghiên cứu mời tới văn phòng và cho uống thử mấy loại bia (hình như là 3 loại) sau đó bạn ấy nhờ tôi điền các phiếu cho ý kiến (lâu quá rồi cũng không nhớ cái phiếu đó hỏi cái gì).
- (4) Trong các thương vụ đầu tư dự án cũng vậy. Sau khi tính toán chi phí, lợi nhuận của hai dự án. Câu hỏi đặt ra là kết quả hai dự án có sự khác biệt không?
 - (5) Trong kinh doanh cũng có thể có câu hỏi là doanh số, lợi nhuận của hai sản phẩm trong cùng thời gian có sự khác biệt không?

Mở rộng vấn đề số 1 thì trong lĩnh vực y khoa thì có nhiều dạng nghiên cứu sẽ có nhu cầu so sánh hai tỉ lệ như:

- Nghiên cứu lâm sàng đối chứng ngẫu nhiên
- Nghiên cứu cắt ngang
- Nghiên cứu bệnh chứng

Nghiên cứu thuốc Zoledronate chống gãy xương

Ví dụ một nghiên cứu được công bố trên website PMC¹¹ về một loại thuốc Zoledronate và gãy xương.

¹¹ <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2324066/>

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

Nghiên cứu này theo dõi 1065 người được điều trị (control) và 1062 người dùng giả dược (thuật ngữ y khoa gọi là Placebo, hoặc nhóm chứng) trong vòng 3 năm. Ghi nhận ai bị gãy xương, ai không gãy xương.

Nhìn qua bảng 2 bên dưới để cảm nhận một chút số liệu:

Table 2

Rates of Fracture and Death in the Study Groups.*

Variable	Placebo (N = 1062)	Zoledronic Acid (N = 1065)	Hazard Ratio (95% CI)	P Value
no. (<i>cumulative rate or %</i>)				
Fracture				
Any	139 (13.9)	92 (8.6)	0.65 (0.50–0.84)	0.001
Nonvertebral	107 (10.7)	79 (7.6)	0.73 (0.55–0.98)	0.03
Hip	33 (3.5)	23 (2.0)	0.70 (0.41–1.19)	0.18
Vertebral	39 (3.8)	21 (1.7)	0.54 (0.32–0.92)	0.02
Death	141 (13.3)	101 (9.6)	0.72 (0.56–0.93)	0.01

*Rates of clinical fracture were calculated by Kaplan–Meier methods at 24 months and therefore are not simple percentages. Because of variable follow-up, the number and percentage of patients who died are provided on the basis of 1057 patients in the placebo group and 1054 patients in the zoledronic acid group in the safety population.

Biến cố quan trọng nhất trong nghiên cứu này là tử vong (Death) – quan sát dòng cuối cùng. Sau 3 năm theo dõi thì nhóm chứng có 141 người chết (tương ứng 13.3%), nhóm được điều trị bằng Zoledronic có 101 người chết (tương ứng 9.6%). Do lúc đầu chia nhóm là ngẫu nhiên và số lượng gần như nhau nên kết quả này cho thấy có 40 người được cứu sống trong nhóm điều trị.

Câu hỏi đặt ra là thuốc này có hiệu quả không?

Hazard Ratio là tỉ số rủi ro, tức là nguy cơ tử vong được tính bằng tỉ số người chết của nhóm điều trị và nhóm chứng = 101/141 = 0.72. Nói cách khác là thuốc này giảm nguy cơ tử vong 28% ($1 - 0.62 = 0.28$). Độ tin cậy (Confidence Interval) 95% dao động từ 0.56 đến 0.92. Chỉ số P bằng 0.01 (< 0.05).

Độ tin cậy 95% ở đây có nghĩa là nếu điều trị tiếp 100 bệnh nhân bằng thuốc này thì có 95% bệnh nhân có tỉ lệ giảm nguy cơ tử vong từ 7% đến 44% (7% được tính từ hiệu 100% - 93%; 44% được tính từ hiệu 100% - 56%). Chỉ số $P < 0.05$ ý nói là kết quả giảm nguy cơ tử vong 28% nếu dùng thuốc là có ý nghĩa thống kê.

Nghiên cứu mối liên quan giữa hút thuốc lá và ung thư phổi

Một nghiên cứu khác rất nổi tiếng từ năm 1950 bởi Sir Richard Doll: là mối liên quan giữa ung thư và thuốc lá¹². Kết quả nghiên cứu được tổng kết trong bảng sau:

	Lung Cancer	Controls
Smokers	647	622
Non-smokers	2	27
Total	649	649

Nghiên cứu này chọn ra 649 người bị ung thư phổi (lung cancer) và bắt cặp với 409 người không bị ung thư phổi (gọi là nhóm chứng – nhóm controls). Sau đó đặt câu hỏi “Có hút thuốc không?”. Trong nhóm ung thư phổi thì có 2 người không hút thuốc, 647 người hút thuốc. Trong nhóm chứng (không bị ung thư phổi) thì có 27 người không hút thuốc và 622 người hút thuốc.

Câu hỏi đặt ra là có mối liên quan giữa hút thuốc và ung thư phổi hay không?

Ý tưởng chính của nghiên cứu này là tính tỉ lệ người hút thuốc lá giữa hai nhóm. Sau đó so sánh hai tỉ lệ. Nếu hai tỉ lệ này không có sự khác biệt chứng tỏ không có mối liên quan giữa hút thuốc lá và ung thư phổi.

Các tình huống ở trên tựu trung lại là sánh hai tỉ lệ. Phần tiếp theo sẽ cùng nhau khám phá ý tưởng đằng sau các phương pháp và cách sử dụng Python để phân tích.

Khái niệm quần thể (population) và mẫu (sample)

Trong các nghiên cứu hay dự án nói chung thì việc lấy được số liệu của tất cả các đối tượng để hiểu và phân tích rất khó. Đặc biệt khi liên quan đến yếu tố chi phí. Từ đó có một hướng tiếp cận là nghiên cứu một nhóm đối tượng vừa phải vừa tiết kiệm chi phí mà vẫn đạt được mục tiêu đề ra. Có vài khái niệm liên quan như:

Quần thể - population

Population theo từ điển Oxford là:

Góc tiếng Anh

Population

- ▶ **NOUN** all the inhabitants of a particular town, area, or country
- ▶ inhabitant: a person or animal that lives or occupies a place.
- **the island has a population of about 78,000.** (hòn đảo có dân số khoảng bảy mươi tám nghìn người)

Nghĩa hẹp hơn với US: a person who fulfills the requirements for legal residency.

¹² <https://www.bmjjournals.org/content/2/4682/739>

Như vậy nghĩa thông thường của Population là: **dân số**, số người trong một vùng, hoặc trong một quốc gia.

- ▶ **Trong lĩnh vực Biology:** a community of animals, plants, or humans among whose members interbreeding occurs.
- ▶ **interbreed:** giao phối (động vật), lai giống.
- ▶ **Trong lĩnh vực Statistics:** a finite or infinite collection of items under consideration.

Nghĩa rộng của Population **quần thể**, tức là bao gồm tất cả các đối tượng mà chúng ta đề cập trong một phạm vi nào đó. Quần thể có thể bao gồm con người, con vật, cây cối, và nhiều thứ khác thuộc trong phạm vi chúng ta đang nghiên cứu.

Mẫu – Sample

Một nhóm nhỏ các đối tượng trong quần thể. Trong lĩnh vực thống kê thì có nhiều phương pháp lấy mẫu để đảm bảo tính ngẫu nhiên phục vụ cho mục tiêu nghiên cứu.

Tình huống

Chúng ta cần nghiên cứu chiều cao sinh viên Việt Nam thì sẽ rất tốn kém và mất thời gian nếu đi đo hết tất cả chiều cao của từng sinh viên. Nếu làm được thì quá tốt rồi. Tuy nhiên trong một thời gian định trước và ngân sách cho phép thì làm cách nào ước tính được chiều cao sinh viên Việt Nam một cách khoa học?

Câu hỏi đặt ra là chiều cao sinh viên năm cuối giữa các chuyên ngành hoặc các khoa có sự khác biệt không?

Phân tích so sánh hai nhóm bằng tỉ số z test

Ví dụ cho 2 nhóm các thông tin bên dưới:

		Sample (mẫu)	
		Nhóm 1	Nhóm 2
N	n ₁	n ₂	
Xác suất outcome	p ₁	p ₂	
Độ lệch chuẩn	s ₁	s ₂	

Outcome ở đây là biến cố, hoặc sự kiện kết quả trong nghiên cứu. Ví dụ: bị bệnh, bị tử vong, bị gãy xương. Trong các tình huống kinh doanh, giáo dục như: bán được hàng, đạt bài kiểm tra.

Độ lệch chuẩn trong từng nhóm được tính như sau:

$$s = \sqrt{\frac{p(1-p)}{N}}$$

Để so sánh hai nhóm thì đầu tiên cần tính **Hiệu số ảnh hưởng**:

$$d = p_1 - p_2$$

Sau đó tính **độ lệch chuẩn của d**:

$$s = \sqrt{s_1^2 + s_2^2}$$

s_1^2 : là phương sai của nhóm 1

s_2^2 : là phương sai của nhóm 2

Tính **tỉ số z test**:

$$z_{\text{test}} = \frac{d}{s}$$

KTC 95% của $d = d \pm 1.96 \times s$

(1.96 là hằng số của phân bố chuẩn)

Nếu chỉ số z test bằng 1 thì có thể kết luận không có sự khác biệt giữa 2 nhóm.

Nếu chỉ số z test bằng 2 đến 3 lần thì có thể thấy có sự khác biệt giữa 2 nhóm.

Phân bố chuẩn

Trên đây có nhắc tới phân bố chuẩn. Vậy phân bố chuẩn là gì? Để trả lời câu câu hỏi này thì chúng ta ôn lại vài khái niệm.

Hàm phân phối

Tình huống là khi bạn có một thông tin đầu vào (gọi là một biến) ví dụ bạn phỏng vấn vài trăm kỹ sư CNTT và hỏi lương¹³ của họ thì liệt kê ra như sau:

x: 2, 3, 2.5, 1.9, 4, 2.8, 2.9, 5, 5.2, 3.6, 4.1, ... (đơn vị: nghìn đô)

Câu hỏi đặt ra là các mức lương này có qui luật gì không? Nếu bạn có dữ liệu đủ lớn và được thu thập một cách ngẫu nhiên thì qui luật có khác với tình huống là thu thập dữ liệu trong một nhóm chuyên ngành hép không?

Câu hỏi mà các nhà nghiên cứu thống kê sẽ hỏi là: Phân phối (distribution), hay còn gọi là phân bố của tiền lương này như thế nào? Khi nói đến phân phối là nói đến khả năng, hay tần suất mà giá trị biến số có thể xảy ra. Ví dụ mức lương 1000 USD khả năng xảy ra là bao nhiêu phần trăm?

Kí hiệu $P(X = x)$ là xác suất của biến X có giá trị cụ thể x.

Như vậy nếu chúng ta rảnh thì có thể ngồi tính xác suất cho từng giá trị của X. Nếu tìm được hàm số nào đó dạng $f(x)$ để biểu diễn giá trị xác suất này thì sẽ rất thuận

¹³ Nguồn tin tại: <https://vov.vn/xa-hoi/luong-ky-su-cong-nghe-thong-tin-hang-nghin-do-moi-thang-1061726.vov> (thời điểm tháng 6/2020)

lợi cho việc tính toán, suy luận để khám phá ra thông tin mới. Hàm này gọi là **hàm phân phối xác suất**.

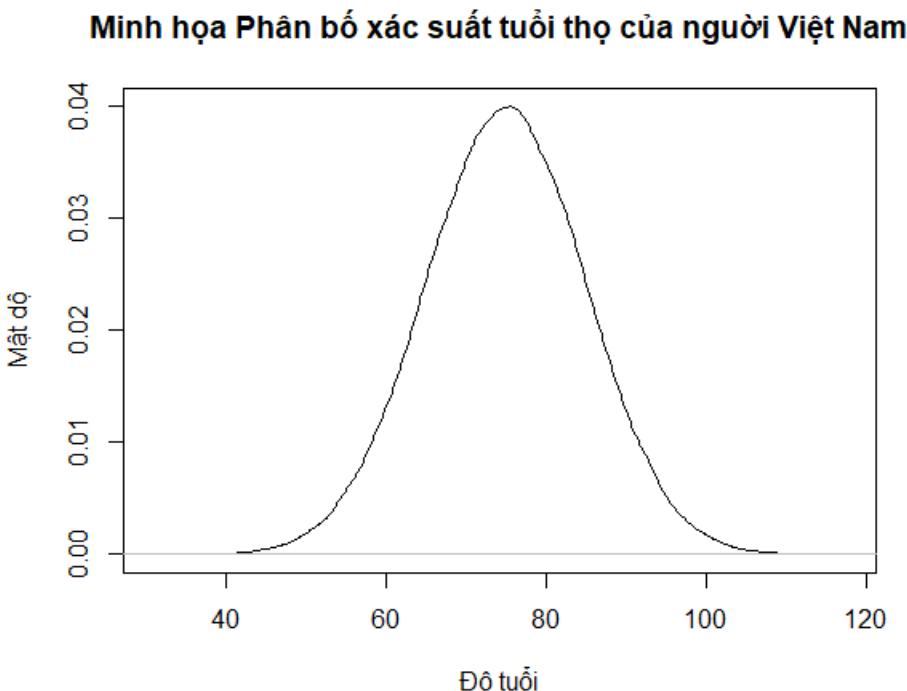
Nếu X là biến liên tục thì $f(x)$ gọi là **hàm mật độ xác suất** (probability density function). Nếu X là biến rời rạc (hoặc là biến phân loại, hoặc là danh mục) thì $f(x)$ gọi là probability mass function. Để phân biệt hai tên gọi này thì bạn hình dung chữ “density” có nghĩa là mật độ, có thể đo được bằng tần số (frequence) xuất hiện xác suất và chỉ đo được cho biến liên tục. Chữ “mass” có nghĩa là “lớn”, “số lượng lớn”. Tức là biến phân loại thì giá trị không liên tục nên hàm số cho biết xác suất của từng loại giá trị “lớn” đến mức nào.

Đối với một biến X, gọi x (nhỏ) là tất cả các giá trị của X thì cộng hết tất cả $f(x)$ là thì bạn đoán là bao nhiêu? Nhớ là $f(x)$ là xác suất của X khi $X = x$. Xác suất là khả năng xảy ra của biến cố. Khả năng thì xảy ra từ 0% đến 100%. Tức là $f(x)$ có giá trị từ 0.0 đến 1.0. Vì thế hình dung Tổng($f(x)$) sẽ bằng 1.

Như vậy bạn đã hình dung người ta định nghĩa **phân bố xác suất** rồi nhé! Thế còn phân bố chuẩn (normal distribution) là gì?

Hơi dài dòng một chút nhưng tôi tin là đa số các bạn sẽ dễ hình dung. Cá nhân tôi cho rằng từ quan sát trong thực tế cuộc sống thì các đối tượng nói chung là có chu kỳ sống của nó theo luật “sinh – phát – diệt”. Để thấy nhất là con người chúng ta nói riêng, động vật, cây cỏ, rồi cả các công ty, v.v...đều có chung quy luật là được sinh ra, phát triển một thời gian, sau đó thoái trào/già yếu, rồi sẽ biến mất hoặc chuyển sang hình thái/dạng khác. Phân lớn các đối tượng, thông tin nghiên cứu trong thống kê nói riêng và trong cuộc sống nói chung thì có liên quan đến qui luật này. Vì vậy các nhà toán học, các nhà thống kê học mới nghĩ ra hàm $f(x)$ để biểu diễn xác suất cho các biến cố của các đối tượng này. Hàm này gọi là **hàm phân phối chuẩn** (normal distribution). Chữ “normal” có nghĩa là bình thường, tức là bình thường như trong tự nhiên. Dịch “normal” ra tiếng Việt “chuẩn” là chuẩn theo tự nhiên, chuẩn theo điều bình thường.

Sơ đồ của luật “sinh – phát – diệt” sẽ có dạng hình quả chuông (bell) như sau:



Từ thực tế như vậy nên hàm normal distribution $f(x)$ được nhà toán học Gauss phát triển bởi 2 thông số: trung bình (μ) và độ lệch chuẩn (σ). Nếu biến X tuân theo luật phân bố chuẩn với trung bình μ và độ lệch chuẩn σ thì hàm mật độ phân bố của X được viết như sau:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Nếu thấy hàm số này phức tạp thì hãy bỏ qua. Bạn hình dung ý tưởng của nó là được rồi. Từ việc định nghĩa hàm “normal” như thế này thì các nhà toán học, thống kê học đưa ra rất nhiều các thuật toán khác để phân tích và suy diễn ra thông tin mới.

Trong thực tế thì dữ liệu chúng ta thu thập sẽ hiếm khi có phân phối xác suất “đẹp” như quả chuông. Đôi khi chúng ta cũng phải chấp nhận dữ liệu đạt ngưỡng nào đó gần gần như “quả chuông” để thừa hưởng các thuật toán phân tích, suy diễn của các nhà toán học, các nhà thống kê học để khám phá ra “thông tin mới”. Thông tin mới ở đây có thể không hoàn toàn chính xác như trong toán học nhưng nó có ích trong thực tiễn thì đáng để xem xét sử dụng.

Một câu hỏi tiếp theo là làm cách nào để biết dữ liệu có phải là phân bố chuẩn hay không?

Trong Python có thể sử dụng kiểm định thống kê có tên là “Shapiro test” thông qua hàm `shapiro.test`.

Giới hạn của hàm `shapiro.test` là chỉ phân tích dữ liệu dưới 5000 dòng. Quay lại dữ liệu của dự án Bank Marketing thì thử sử dụng `shapiro.test` với 5000 dòng dữ liệu xem như thế nào?

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

Đọc dữ liệu:

```
import pandas as pd  
fp = 'https://thachln.github.io/datasets/bank/bank-additional-  
full.csv'
```

Sử dụng module hàm `stats.shapiro` trong thư viện `scipy` với 5000 dòng đầu tiên của cột `age`:

```
from scipy import stats  
age = df['age']  
age5K = age.loc[0:4999]  
shapiro_test = stats.shapiro(age5K)  
print(shapiro_test)
```

Kết quả:

```
ShapiroResult(statistic=0.9733700156211853, pvalue=1.502039384011789e-29)
```

Vì trị số `p` nhỏ hơn 0.05 nên chúng ta có thể kết luận rằng biến số `age` trong dữ liệu Bank Marketing không tuân theo luật phân phối chuẩn. Chú ý ở đây là chỉ mới phân tích 5000 dòng dữ liệu.

Thử lấy mẫu ngẫu nhiên 5000 dòng rồi chạy lại Shapiro test xem sao:

```
from scipy import stats  
age = df['age']  
age5K = age.sample(n=5000)  
shapiro_test = stats.shapiro(age5K)  
print(shapiro_test)
```

```
ShapiroResult(statistic=0.950770914554596, pvalue=4.941021545202006e-38)
```

Bạn có thể thử nhiều lần, kết quả trị số `p` có thể vẫn nhỏ hơn 0.05.

Một phương pháp khác là Anderson-Darling, không giới hạn dữ liệu:

```
from scipy import stats  
age = df['age']  
anderson_test = stats.anderson(age)  
print(anderson_test)
```

```
AndersonResult(statistic=444.7290025091206, critical_values=array([0.576, 0.656  
, 0.787, 0.918, 1.092]), significance_level=array([15., 10., 5., 2.5, 1.  
]))
```

Kết quả báo cáo của hàm `stats.anderson` hơi phức tạp hơn một chút.

Diễn giải như sau:

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

Thực hiện kiểm định với significance levels lần lượt là: 15%, 10%, 5%, 2.5%, 1% thì kết quả critical values tương ứng là: 0.576, 0.656, 0.787, 0.918, 1.092.

Giá trị statistic (444.7290025091206) **lớn hơn** các kết quả critical values. Vì thế giả thuyết ban đầu (dữ liệu age tuân theo luật phân bố chuẩn) bị bác bỏ.

Như vậy có thể kết luận dữ liệu age trong dự án Bank Marketing không tuân theo luật phân bố chuẩn.

Đọc thêm hướng dẫn tại:

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.anderson.html>

Phân tích kết quả thuốc Zoledronate và gãy xương

Tổng kết số liệu và tính vài chỉ số như sau:

	Nhóm chứng	Nhóm điều trị
Tổng số người	1062	1065
Số người gãy xương	139	92
Tỉ lệ gãy xương	0.131	0.086
Độ lệch chuẩn	0.01035	0.00861
Độ lệch chuẩn tính bằng:	$\sqrt{\frac{0.131(1 - 0.131)}{1062}}$	$\sqrt{\frac{0.086(1 - 0.086)}{1065}}$
Hiệu số ảnh hưởng d =	0.131 – 0.086 = 0.045	
Độ lệch chuẩn của d. s=	$\sqrt{0.01035^2 + 0.00861^2} = 0.0135$	
KTC 95%	$0.045 \pm 1.96 \times 0.0135 = 0.0186 \sim 0.714$	
z test	$0.045 / 0.0135 = 3.33$	
P value	$2 \times (1 - pnorm(3.33)) = 0.001$	

Nhìn vào kết quả trên sẽ có một số diễn giải như sau:

- Nếu thuốc Zoledronic acid không có hiệu quả thì:
 - o $d = 0$
 - o KTC 95% của d sẽ dao động từ âm đến dương
- Nhưng thực tế cho thấy $d \neq 0$ và KTC 95% đều dương

- Do đó, có thể kết luận thuốc Zoledronic acid có hiệu quả giảm nguy cơ gãy xương.

Sử dụng Python

```
import numpy as np
from statsmodels.stats.proportion import proportions_ztest
count = np.array([139, 92])
nobs = np.array([1062, 1065])
zstat, pval = proportions_ztest(count, nobs)
zstat
pval
```

zstat
Out[2]: 3.2980488610898746

pval
Out[3]: 0.0009735919100495131

Sử dụng hàm **proportions_ztest** trong thư viện **statsmodels.stats.proportion** Python sẽ cho ra hai chỉ số:

- zstat: là tỉ số z test. Theo cách tính tay và Python trả lại kết quả như nhau ≈ 3.3.
- pval: trị số P. Theo cách tính tay, và Python trả lại kết quả như nhau ≈ 0.001.

Trên đây là một ví dụ phân tích kết quả nghiên cứu bằng cách so sánh hai nhóm, cụ thể là **so sánh hai tỉ lệ** dùng phương pháp **z test**. Từ đó suy luận ra kết quả nghiên cứu có mang lại lợi ích gì không? Các bạn hoàn toàn thể áp dụng cho các bài toán của mình.

Đây chỉ là một trong các cách phân tích để bạn làm quen. Chủ đề phân tích mô tả còn nhiều phương pháp được áp dụng tùy vào loại dữ liệu, phân bố dữ liệu. Chúng ta sẽ có dịp đi sâu vào chủ đề này sau.

Nhân tiện đã ôn tập Phân bố chuẩn ở trên thì ở đây bàn thêm một chút các loại phân bố khác như: Phân bố nhị thức (binomial distribution), Phân bố Poisson.

Phân bố Nhị thức

Tình huống đặt ra khi chúng ta quan sát hiện tượng mang tính phân loại, hay kết quả có tính danh mục (categorical variable) và nhiều khi chỉ có 2 giá trị. Ví dụ trong dây chuyền sản xuất thì thành phẩm ra có lỗi hay không? Hoặc trong sản xuất phần mềm thì các lập trình viên tạo ra các chức năng có lỗi hay không?

Ví dụ trong một công ty làm phần mềm A sau khi đào tạo nhân viên rồi mới đưa vào dự án làm chính thức. Đơn vị (unit) thành phẩm của phần mềm mà nhân viên làm ra là function (hàm). Nếu công ty A đã thống kê và biết rằng tỉ lệ các hàm bị lỗi ngay sau khi nhân viên hoàn thành việc viết code là 20% ($p = 0.20$). Trong một dự án cụ thể

gồm 150 hàm, phát hiện ra 10 hàm bị lỗi. Câu hỏi đặt ra là kết quả này có đáng ngạc nhiên?

Một ví dụ khác là công ty A mời hai đội thiết kế độc lập để cải tiến một sản phẩm. Sau đó mời 50 khách hàng dùng thử 2 sản phẩm X, Y và hỏi họ thích cái nào? Kết quả thu thập có 20 người thích X, 30 người thích Y. Vấn đề đặt ra là kết quả này đủ để kết luận là nhiều người thích sản phẩm X hơn Y hay chỉ là yếu tố ngẫu nhiên?

Phân bố Poisson

Tình huống thực tế khi số mẫu rất lớn và số biến cố nhỏ, tức là các biến cố rất ít khi xảy ra thì phân phối Poisson được dùng để mô tả xác suất này. Ví dụ trong nghiên cứu các bệnh hiếm gặp như xương thủy tinh, ung thư. Ngoài ra phân bố Poisson phù hợp các vấn đề mang tính số đếm (count) như số nhân viên đi làm trễ mỗi ngày, số dự án bị trễ deadline trong năm, số đơn hàng bị hủy trong tháng, số bệnh nhân nhập viện trong ngày, v.v...

Trên đây có đề cập tới 3 loại phân bố: Phân bố chuẩn, Phân bố Nhị thức và Phân bố Poisson. Chúng ta chưa đi sâu vào công thức toán của nó – nói chung là “phức tạp”. Ngoài ra còn nhiều phân bố khác nữa. Ở đây muốn nhấn mạnh là bạn cần hiểu tùy hiện tượng, tùy vấn đề chúng ta quan sát thì có nhiều loại phân bố phù hợp để mô tả xác suất của sự kiện. Chúng ta sẽ áp dụng các phân bố này trong các bài toán cụ thể với code Python sau.

Bài 15: Mô hình kiểm định giả thuyết

Phương pháp sau đây là tổng hợp từ hai phương pháp **Kiểm định ý nghĩa thống kê** (Test of Significance) và **Kiểm định giả thuyết** (Test of hypothesis) của Fisher và Neyman Pearson bởi những nhà khoa học đời sau. Để đánh giá nghiên cứu thì người ta phát biểu hai giả thuyết phủ định nhau.

Ví dụ một số câu hỏi đặt ra trong vài lĩnh vực như:

- Chiến lược kinh doanh này có hiệu quả / không hiệu quả.
- Thuốc này có hiệu quả giảm tử vong / không có hiệu quả giảm tử vong.
- Phương pháp đào tạo này có hiệu quả / không có hiệu quả.
- Phương pháp quản trị này có hiệu quả / không có hiệu quả.

Phương pháp chung như sau:

Bước ①: Đưa ra giả thuyết phủ định và giả thuyết khẳng định. Hoặc giả thuyết vô hiệu và giả thuyết chính.

Giả thuyết vô hiệu (giả thuyết phủ định) kí hiệu là H_0 .

Giả thuyết chính (giả thuyết khẳng định) kí hiệu là H_1 .

Bước ②: Xác định xác suất để bác bỏ H_0 (gọi là xác suất α), và xác suất để bác bỏ H_1 (gọi là xác suất β). Đồng thời xác định đối tượng dự kiến cần nghiên cứu (ước tính cỡ mẫu – sample size). Cần xác định α và β sao cho chấp nhận được.

Bước ③: Tiến hành thí nghiệm, thu thập số liệu, điều tra để tổng hợp số liệu liên quan đến giả thuyết. Gọi dữ liệu là D .

Bước ④: Ước tính quan sát dữ liệu D nếu H_0 đúng. Kí hiệu xác suất $P(D | H_0)$. Đây chính là giá trị P (P-value). Nếu P-value thấp chứng tỏ xác suất trong dữ liệu D không đủ chứng cứ để tin H_0 đúng. Thấp bao nhiêu thì trong kế hoạch đã định trước (α trong bước ②)

Ví dụ khi $P=0.01$ thì có thể phát biểu như sau: với dữ liệu ta có thì xác suất để H_0 đúng chỉ là 1%. Tức là với dữ liệu quan sát được thì chỉ có 1% trường hợp là H_0 đúng. Như vậy có cơ sở là bác bỏ H_0 .

Bước ⑤: Nếu $P < \alpha$, tức là xác suất quan sát dữ liệu để tin H_0 là đúng có giá trị dưới ngưỡng đã định trong bước ②. Như vậy bác bỏ giả thuyết H_0 . Chú ý bác bỏ giả thuyết H_0 thì không có nghĩa là chấp nhận giả thuyết H_1 .

Bài 16: Ứng dụng minh họa kiểm định giả thuyết

Bài này tôi trình bày một ứng dụng tương tự để áp dụng vài lý thuyết đã được đề cập. Ứng dụng này cũng không hẳn là nằm trong phạm vi Phân tích mô tả. Đây chỉ là một tình huống áp dụng các kỹ thuật đã học để đi tìm một câu trả lời trong một ngữ cảnh cụ thể.

Tình huống:

Trong một công ty cung cấp dịch vụ và sản phẩm liên quan đến phần mềm gồm có hai bộ phận phát triển sản phẩm. Để đơn giản tình huống và minh họa rõ cho phương pháp kiểm định giả thuyết ở trên thì tôi đơn giản hóa bằng các giả định sau:

- Sản phẩm cuối cùng của bốn bộ phận này đều gồm có: mã nguồn và tài liệu.
- Khối lượng đầu ra sản phẩm được tính bằng số dòng mà nguồn (LOC: Line of code) và số trang tài liệu (Page A4). Hệ số, hoặc trọng số của các sản phẩm nói chung là như nhau.
- Số LOC và Page A4 được thu thập vào cuối mỗi tháng sau khi đã được thực hiện đầy đủ các qui trình kiểm tra (review) kiểm thử (testing) và được bộ phận đảm bảo chất lượng (QC) xác nhận.
- Năng suất trung bình của các công ty trong cùng lĩnh vực, cùng loại sản phẩm thì năng suất là: 2000 LOCs/man.month (man.month: tháng công). Giả định trình độ, kinh nghiệm của nhân viên trong công là tương đương với các công ty cùng lĩnh vực.

Câu hỏi đặt ra là: **năng lực tiềm ẩn của các nhân viên trong các bộ phận đã được phát huy hết chưa?** Câu hỏi này được Giám đốc Tiềm Năng Và Phát Triển Nhân Sự (HRIPD - Human Resource and Inside Power Director) đưa ra. Câu hỏi này được chuyển tới Data Coordinator¹⁴ (DC). DC triển khai câu hỏi này rõ hơn một chút:

¹⁴ Người tương đương, có phần cao hơn một chút các vị trí Data Science/Data Analytics được mô tả ngoài thị trường lao động. Data Coordinator làm công việc “Người liên kết dữ liệu” để phục vụ cho mục tiêu kinh doanh. Data Coordinator có thể bao gồm các việc của Data Analytics, Data Scientist tùy từng thời điểm. Data Coordinator hướng tới mang lại giá trị dài hạn cho tổ chức bằng cách kết hợp công việc của Data Engineering, Data Analytics, Data Scientist.

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

- Giả định toàn bộ kết quả của nhân viên được nộp đầy đủ lên máy chủ GIT¹⁵ của công ty mỗi ngày, mỗi tuần. Theo quy định của công ty thì ngày cuối tuần toàn bộ sản phẩm (mã nguồn, tài liệu) phải được nhân viên nộp (Push) lên máy chủ theo đúng quy trình để các đội liên quan có thể theo dõi và đánh giá. Đặc biệt là bộ phận Đánh Giá Tài Sản có thể ước lượng tài sản mềm của công ty.
- Với dữ liệu thu thập được hàng tuần/tháng/quý/năm thì câu hỏi đặt ra là **Có sự khác biệt lớn về năng suất giữa hai bộ phận này không?**

Dữ liệu mẫu tại https://thachln.github.io/datasets/sample_dev_prod.csv.

Code Python sau đọc file dữ liệu và vẽ biểu đồ boxplot về năng suất lập trình (biên prod) giữa hai nhóm (biên team).

Đọc dữ liệu vào DataFrame:

```
import pandas as pd
fp = 'https://thachln.github.io/datasets/sample_dev_prod.csv'
df = pd.read_csv(fp)
```

Chuyển kiểu dữ liệu trong cột team từ dạng số sang dạng danh mục

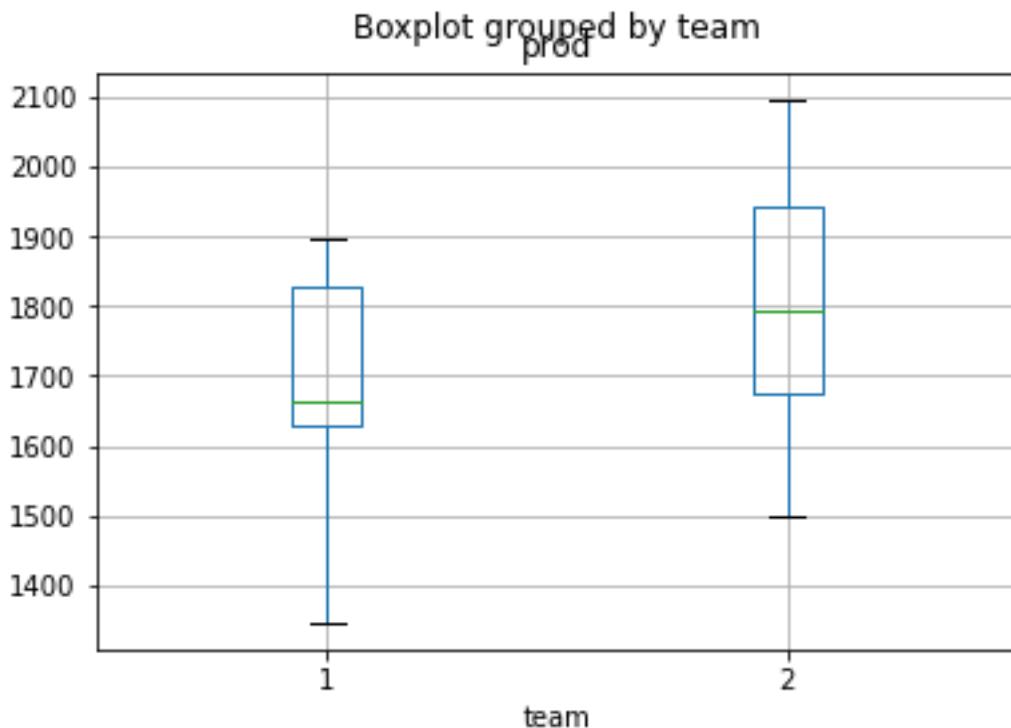
```
df['team'] = df['team'].astype('category')
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   team      37 non-null    category
 1   prod      37 non-null    int64   
dtypes: category(1), int64(1)
memory usage: 557.0 bytes
```

Vẽ boxplot với dữ liệu prod theo team để có cảm nhận về năng suất giữa 2 đội:

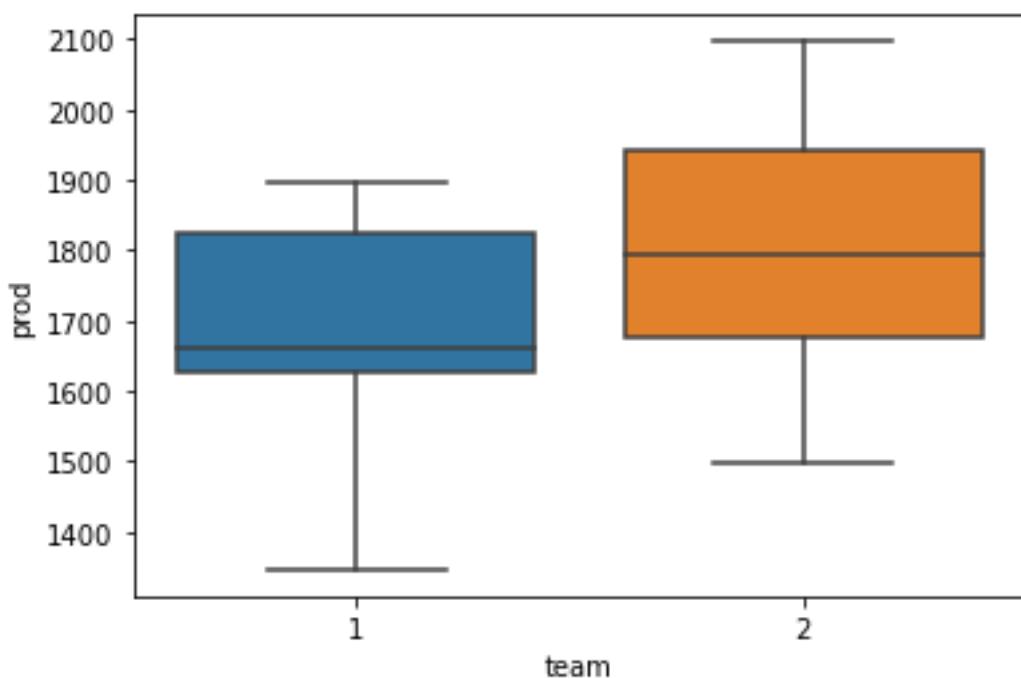
```
df.boxplot(column='prod', by='team')
```

¹⁵ GIT là phần mềm thường dùng cho các nhà phát triển phần mềm để lưu phiên bản của mã nguồn hoặc tài liệu. Trang web nổi tiếng là nơi chia sẻ và là nơi cộng tác của nhiều nhà phát triển phần mềm khắp thế giới <https://github.com>.



Có thể dùng boxplot của thư viện seaborn:

```
import seaborn as sns  
sns.boxplot(x=df['team'], y=df['prod'])
```



Phần tiếp theo chúng ta sẽ áp dụng phương pháp Phân tích Phương sai để đi tìm câu trả lời. Đồng thời minh họa rõ hơn cho phương pháp Hỗn hợp gồm **Kiểm định ý**

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

nghĩa thống kê (Test of Significance) và **Kiểm định giả thuyết** (Test of hypothesis) đã nêu ở trên.

Phân tích phương sai

Sử dụng hàm `anova_lm` trong thư viện `statsmodels.api` của Python để phân tích phương sai như sau:

```
import statsmodels.api as sm
from statsmodels.formula.api import ols

# Ordinary Least Squares (OLS) model
model = ols('prod ~ C(team)', data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
anova_table
```

Kết quả:

	sum_sq	df	F	PR(>F)
C(team)	1.543561e+05	1.0	4.923154	0.033079
Residual	1.097359e+06	35.0	NaN	NaN

Một cách khác là dùng thư viện `bioinfokit` bằng cách cài đặt với lệnh sau:

```
pip install bioinfokit
```

Phân tích anova với thư viện `bioinfokit`:

```
from bioinfokit.analys import stat
res = stat()
res.anova_stat(anova_model='prod ~ C(team)', df=df, res_var='prod')
res.anova_summary
```

	df	sum_sq	mean_sq	F	PR(>F)
C(team)	1.0	1.543561e+05	154356.148157	4.923154	0.033079
Residual	35.0	1.097359e+06	31353.101558	NaN	NaN

Kết quả cho thấy mức độ biến thiên giữa hai nhóm cao hơn mức độ biến thiên trong mỗi nhóm là **31353** (Giá trị Mean Square ở dòng Residual). Với kiểm định F = **4.923154** và trị số P = **0.033079**, chúng ta kết luận rằng có sự khác biệt giữa hai nhóm về năng suất lập trình.

Ghi chú: bạn thấy thư viện ở trên trong Python dùng hàm với tham số `'prod ~ C(team)'`, dấu ngã ‘~’ có nghĩa là cần phân tích biến `prod` trong các `team`. Tức là nếu có nhiều nhóm thì hàm này vẫn sử dụng bình thường. Thật ra là phương pháp Phân tích phương sai có điểm mạnh là để so sánh **một biến liên tục** giữa **nhiều nhóm** (lớn hơn 2). Phương pháp này có tên gọi là ANOVA.

Tham khảo thêm: <https://www.reneshbedre.com/blog/anova.html>

Phân tích t-test

Với phương pháp ANOVA ở trên đã ra kết quả. Tuy nhiên bài toán minh họa ở đây chỉ có 2 nhóm nên có một cách đơn giản hơn là dùng kiểm định t (t-test).

Trong Python có thể dùng thư viện bioinfokit.analys như sau:

```
import pandas as pd
fp = 'https://thachln.github.io/datasets/sample_dev_prod.csv'
df = pd.read_csv(fp)

from scipy import stats as st
from bioinfokit.analys import get_data, stat

# t test using bioinfokit
res = stat()
res.ttest(df=df, test_type=2, xfac='team', res='prod', mu=2,
evar=True)
print(res.summary)
```

Two sample t-test with equal variance

Mean diff	-131.555
t	-2.21882
Std Error	59.2904
df	35
P-value (one-tail)	0.0165395
P-value (two-tail)	0.033079
Lower 95.0%	-251.92
Upper 95.0%	-11.1887

Parameter estimates

Level	Number	Mean	Std Dev	Std Error	Lower 95.0%	Upper 95.0%
1	15	1685.4	169.25	43.7	1591.67	1779.13
2	22	1816.95	182.094	38.8226	1736.22	1897.69

Năng suất trung bình của team 1 là **1685.4 LOCs/man.month** và của team 2 là **1816.95 LOCs/man.month**. Do đó mức độ khác biệt giữa hai team là **-131.555** (Team 1 có năng suất trung bình **thấp** hơn team 2 là 131.555 LOCs/man.month). Khoảng tin cậy của khác biệt là **-251.92** đến **-11.1887**. Nói cách khác nếu nghiên cứu lặp lại 100 lần (tức là cả 2 team có cơ hội làm 100 dự án tương tự) thì sẽ có 95 dự án với năng suất trung bình của team 1 thấp hơn team 2 từ 11.2 đến 252 LOCs/man.month. Trị số P < 0.05 nên chúng ta kết luận rằng sự khác biệt này có ý nghĩa thống kê.

Kiểm định giả thuyết

Ở trên đã sử dụng phân tích ANOVA và t-test để kết luận là năng suất lập trình của hai team là có sự khác biệt (có ý nghĩa thống kê).

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

Một câu hỏi mới được đưa ra như sau: Năng suất lập trình của team 1 và team 2 có đạt chuẩn hay không?

Theo giả định ở trên thì chuẩn là một lập trình viên đạt 2000 LOCs/man.month.

Ở đây chúng ta có 2 team nên sẽ có hai phần đánh giá năng suất cho 2 team. Về mặt phương pháp thì có thể sử dụng ngay phương pháp t.test ở trên để so sánh từng nhóm như sau:

```
df1 = df[df['team'] == 1]
res.ttest(df=df1, test_type=1, res='prod', mu=2000)
```

```
df2 = df[df['team'] == 2]
res.ttest(df=df2, test_type=1, res='prod', mu=2000))
```

Tham số mu=2000 là số trung bình tiêu chuẩn cần được đối chiếu.

Tuy nhiên để minh họa cho phương pháp kiểm định giả thuyết ở trên thì cần trình bày các bước cho rõ ràng một chút.

Câu hỏi thứ nhất: Team 1 không đạt chuẩn năng suất phải không?

Bước 1:

Phát biểu giả thuyết vô hiệu H_0 : Năng suất ≥ 2000 LOCs/man.month

Giải thuyết chính H_1 : Năng suất < 2000 /man.month

Bước 2:

Xác định $\alpha = 0.05$: là xác suất có thể chấp nhận dữ liệu có Năng suất ≥ 2000 LOCs/man.month; $\beta = 0.80$.

Bước 3:

Ước tính cỡ mẫu và thu thập số liệu.

Bước 4:

Sử dụng phương pháp t.test:

```
import pandas as pd
fp = 'https://thachln.github.io/datasets/sample_dev_prod.csv'
df = pd.read_csv(fp)

from scipy import stats as st
from bioinfokit.analys import get_data, stat

# t test using bioinfokit
```

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

```
res = stat()  
  
df1 = df[df['team'] == 1]  
res.ttest(df=df1, test_type=1, res='prod', mu=2000)  
print(res.summary)
```

One Sample t-test

Sample size	15
Mean	1685.4
t	-7.19908
Df	14
P-value (one-tail)	2.28571e-06
P-value (two-tail)	4.57143e-06
Lower 95.0%	1591.67
Upper 95.0%	1779.13

Kết quả cho thấy trị số $P = 4.57143e-06$, thấp hơn mức $\alpha (0.05)$ đã xác định.

Và với Năng suất trung bình của team 1 là 1685.4 LOCs/month đương nhiên là thấp hơn mức tiêu chuẩn 2000 LOCs/man.month. Đồng thời KTC95% giao động từ **1591.67** đến **1779.13** (đều thấp hơn 2000) cho thấy **giả thuyết Năng suất của team 1 đạt chuẩn có thể bác bỏ ở mức độ $\alpha = 5\%$.**

Câu hỏi thứ hai: Team 2 có đạt chuẩn năng suất hay không?

Bạn có thể thực hiện tương tự với kết quả trong Python như sau:

```
import pandas as pd
fp = 'https://thachln.github.io/datasets/sample_dev_prod.csv'
df = pd.read_csv(fp)

from bioinfokit.analys import stat

# t test using bioinfokit
res = stat()

df2 = df[df['team'] == 2]
res.ttest(df=df2, test_type=1, res='prod', mu=2000)
print(res.summary)
```

One Sample t-test

Sample size	22
Mean	1816.95
t	-4.71492
Df	21
P-value (one-tail)	5.89042e-05
P-value (two-tail)	0.000117808
Lower 95.0%	1736.22
Upper 95.0%	1897.69

Phần phân tích và rút ra kết luận xem như bài tập nhỏ cho bạn.

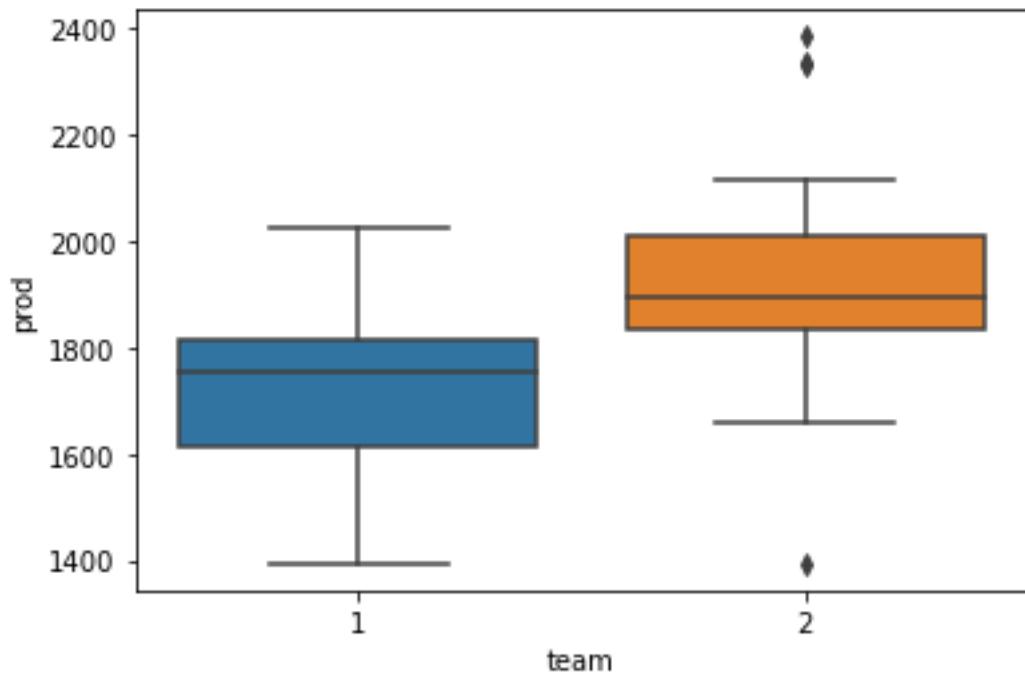
Một tình huống mở rộng cho câu hỏi này là nếu KTC95% không phải đều nhỏ hơn 2000 mà nó phủ lên giá trị 2000 thì sao?

Cụ thể là hãy phân tích dữ liệu thu thập được ở đây:

https://thachln.github.io/datasets/sample_dev_prod_2.csv.

```
import pandas as pd
fp = 'https://thachln.github.io/datasets/sample_dev_prod_2.csv'
df = pd.read_csv(fp)

import seaborn as sns
sns.boxplot(x=df['team'], y=df['prod'])
```



Sử dụng t.test trong Python để so sánh năng suất của team 2 với năng suất tiêu chuẩn 2000 LOCs/man.month:

```
from bioinfokit.analys import stat

# t test using bioinfokit
res = stat()

df2 = df[df['team'] == 2]
res.ttest(df=df2, test_type=1, res='prod', mu=2000)
print(res.summary)
```

Kết quả như sau:

One Sample t-test	
Sample size	22
Mean	1930.41
t	-1.41886
Df	21
P-value (one-tail)	0.0853064
P-value (two-tail)	0.170613
Lower 95.0%	1828.41
Upper 95.0%	2032.41

Năng suất trung bình của team 2 là 1930.41 LOCs/man.month rõ ràng là nhỏ hơn tiêu chuẩn 2000 LOCs/man.month. Tuy nhiên KTC95% dao động từ 1828.41 đến 2032.41, không phải đều nhỏ hơn 2000. Đồng thời P value 0.170613 lớn hơn trị số α (0.05) đã thiết lập.

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

Vì vậy dù Năng suất trung bình < 2000 LOCs/man.month nhưng dữ liệu cho thấy H_0 (Năng suất dưới tiêu chuẩn) bị bác bỏ với $\alpha = 5\%$. Chú ý bác bỏ H_0 không có nghĩa là chấp nhận H_1 (Năng suất đạt hoặc trên tiêu chuẩn).

Bài 17: Phân tích mối tương quan

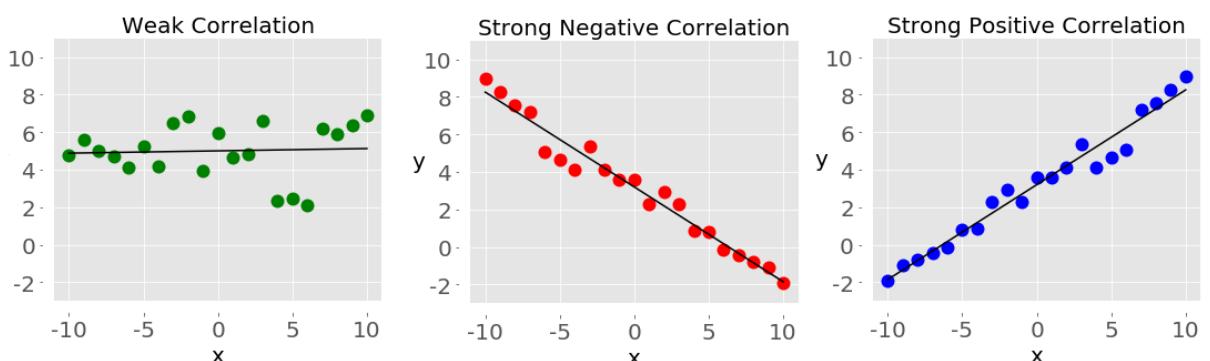
Khi cần khảo sát mối quan hệ (relationships) giữa hai hay nhiều biến (còn gọi là đặc trưng – features) thì chúng ta cần đến khái niệm **correlation** – mối tương quan. Khảo sát ví dụ sau:

```
import pandas as pd
df =
pd.read_csv('https://thachln.github.io/datasets/sample_health_vn.csv')
# tì lệ vòng eo với chiều cao
df['whtr'] = df['waist'] / df['height']
df.columns
df[['age', 'weight', 'whtr']]
```

	age	weight	whtr
0	23	38	0.466216
1	26	57	0.479532
2	66	77	0.524390
3	55	73	0.523529
4	30	59	0.493506
..
3955	48	39	0.448052
3956	25	50	0.528302
3957	18	51	0.415663
3958	38	63	0.503311
3959	48	59	0.500000
[3960 rows x 3 columns]			

Câu hỏi đặt ra là có mối liên quan nào giữa 3 thông tin tuổi, cân nặng và tỉ số vòng eo cho với chiều cao (tương ứng 3 biến age, weight, và whtr) không?

Nếu khảo sát số liệu liệu giữa 2 biến (x và y) thì có khả năng xảy ra 3 trường hợp sau:



- Hình bên trái minh họa biến x và y không có mối liên quan một cách rõ ràng.
- Hình ở giữa minh họa khi x càng tăng thì y càng giảm, hoặc ngược lại y càng tăng thì x càng giảm. Mối tương quan này gọi là tương quan nghịch (**negative correlation**). Đường thẳng minh họa đi xuyên qua dữ liệu cho

thấy các chấm đỏ nằm rất gần đường thẳng. Điều này cho thấy sự phụ thuộc của x vào y hoặc y vào x là rất lớn. Tức là nếu chỉ biết x thì có thể suy ra y và ngược lại. Thuật ngữ gọi là mối tương quan mạnh (**strong**).

- Hình bên phải minh họa ngược lại hình ở giữa, khi x tăng thì y tăng, và ngược lại y tăng thì x tăng. Mối tương quan này gọi là tương quan thuận (positive correlation). Các điểm tròn màu xanh bao quanh rất sát đường kẻ cho thấy đây là mối tương quan mạnh. Gọi đây là mối tương quan thuận mạnh (strong positive correlation)

Để đo mức độ tương quan giữa 2 biến thì chúng ta dùng khái niệm hệ số tương quan (**correlation coefficient**). Có 3 hệ số tương quan phổ biến:

- Hệ số tương quan Pearson (Pearson's r)
- Hệ số tương quan Spearman (Spearman's rho)
- Hệ số tương quan Kendall (Kendall's tau)

Hệ số tương quan Pearson đo mức độ tương quan tuyến tính, trong khi đó hai hệ số còn lại (Spearman và Kendall) so sánh thứ bậc của dữ liệu. Thứ bậc được hiểu là các dữ liệu có giá trị dạng danh mục (categorical) bao gồm danh mục không quan tâm đến thứ tự (nominal) và danh mục có thứ tự (ordinal).

Minh họa hệ số tương quan Pearson

Với dữ liệu minh họa ở trên, đoạn code sau tính toán hệ số tương quan Pearson giữa 2 biến:

- Tuổi và Cân nặng
- Tuổi và Tỉ số vòng eo và chiều cao (gọi tắt là Tỉ số eo cao)

```
import scipy.stats
scipy.stats.pearsonr(df['age'], df['weight'])
scipy.stats.pearsonr(df['age'], df['whtr'])

scipy.stats.pearsonr(df['age'], df['weight'])
Out[1]: (0.36696368575726984, 1.6910450238144636e-126)

scipy.stats.pearsonr(df['age'], df['whtr'])
Out[2]: (0.4525146650843153, 3.3327035799565613e-199)
```

Kết quả của hàm `scipy.stats.pearsonr` cho ra 2 thông tin (a, b):

- Hệ số tương quan: giá trị a
- Trị số p (gọi là p-value): giá trị b

Có thể gán 2 giá trị này cho 2 biến tương ứng khi gọi hàm như sau:

```
r, p = scipy.stats.pearsonr(df['age'], df['weight'])  
print('Hệ số tương quan giữa Tuổi và Cân nặng là: %.2f' % r)  
print('Trị số p trong mối tương quan giữa Tuổi và Cân nặng là: %s' % p)
```

Trị số p trong mối tương quan giữa Tuổi và Cân nặng là: 0.000000

Trị số p trong mối tương quan giữa Tuổi và Cân nặng là: 1.6910450238144636e-126

Điễn giải trị số P

Ý tưởng về trị số P ở đây là sử dụng phương pháp Kiểm định giả thuyết (Testing a hypothesis). **Giả thuyết vô hiệu** là hai biến cần khảo sát **không có mối tương quan** với nhau (hệ số tương quan = zero). Nguồn giá trị p thông thường được dùng là 0.05. Nếu p-value < 0.05 có nghĩa là: Nếu lặp lại thí nghiệm nhiều lần thì với dữ liệu thu thập được có ít hơn 5% trường hợp cho thấy Giải thuyết vô hiệu là đúng. Con số 0.05 ý nói xác suất để dữ liệu phản ánh dữ liệu đúng theo Giải thuyết vô hiệu là rất nhỏ. Vì vậy cần phải bác bỏ Giải thuyết vô hiệu. Tức là dữ liệu cho thấy hai biến cần khảo sát **có mối tương quan với nhau**. Tóm lại khi sử dụng hàm `scipy.stats.pearsonr` mà thấy p-value nhỏ hơn 0.05 thì có thể kết luận là 2 biến đang khảo sát có mối tương quan với nhau.

Trong ví dụ trên thì p-value giữa Tuổi và Cân nặng, giữa Tuổi và Tỉ số eo cao đều rất nhỏ (e-126, e-199 có nghĩa là 10^{-126} và 10^{-199}). Chứng tỏ Tuổi và Cân nặng có tương quan với nhau; Tuổi và Tỉ số eo cao có tương quan với nhau.

Độ mạnh

Câu hỏi tiếp theo là cái nào tương quan mạnh hơn cái nào? Xét hệ số tương quan giữa Tuổi và Tỉ số eo cao là 0.45. Trong khi hệ số tương quan giữa Tuổi và Cân nặng là 0.37. Số này càng lớn thì cho biết độ mạnh của mối tương quan càng cao.

Hệ số tương quan Spearman

Gọi hàm tương tự như trên. Sử dụng hàm `scipy.stats.spearmanr` như sau:

```
scipy.stats.spearmanr(df['age'], df['weight'])  
scipy.stats.spearmanr(df['age'], df['whtr'])  
  
scipy.stats.spearmanr(df['age'], df['weight'])  
Out[1]: SpearmanResult(correlation=0.37155082342065937, pvalue=7.1036986560963  
61e-130)  
  
scipy.stats.spearmanr(df['age'], df['whtr'])  
Out[2]: SpearmanResult(correlation=0.451305999357554, pvalue=5.056000605690559  
6e-198)
```

Hệ số tương quan kendall

Tương tự như trên, gọi hàm `scipy.stats.kendalltau` như sau:

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

```
scipy.stats.kendalltau(df['age'], df['weight'])  
scipy.stats.kendalltau(df['age'], df['whtr'])  
  
scipy.stats.kendalltau(df['age'], df['weight'])  
Out[1]: KendalltauResult(correlation=0.25695403548521434, pvalue=1.305086742981  
229e-124)  
  
scipy.stats.kendalltau(df['age'], df['whtr'])  
Out[2]: KendalltauResult(correlation=0.3109251610720995, pvalue=3.4532852226509  
24e-185)
```

Gọi hàm .corr() trực tiếp trên cột của DataFrame

Ngoài cách sử dụng thư viện `scipy.stats` để tính các hệ số tương quan giữa hai cột dữ liệu như minh họa ở trên thì thư viện `pandas` có cung cấp sẵn hàm `.corr()` để tính trực tiếp trên cột dữ liệu bằng cách sau:

```
cột_x.corr(cột_y, method='method_name')
```

Nếu method không được chỉ định thì phương pháp Pearson được sử dụng.

Minh họa

```
df['age'].corr(df['weight'])  
df['age'].corr(df['weight'], method='pearson')  
df['age'].corr(df['weight'], method='spearman')  
df['age'].corr(df['weight'], method='kendall')  
  
df['age'].corr(df['weight'])  
Out[1]: 0.36696368575726995  
  
df['age'].corr(df['weight'], method='pearson')  
Out[2]: 0.36696368575726995  
  
df['age'].corr(df['weight'], method='spearman')  
Out[3]: 0.37155082342065937  
  
df['age'].corr(df['weight'], method='kendall')  
Out[4]: 0.25695403548521434
```

Mối tương quan tuyến tính

Tiếp tục với bộ dữ liệu ở trên, bạn nhận thấy có thể có mối quan hệ tuyến tính giữa Tuổi và Cân nặng.

Để khảo sát mối tương quan tuyến tính thì có thể dùng hàm `linregress` trong thư viện `scipy.stats` như sau:

```
scipy.stats.linregress(x, y)
```

Minh họa

Khảo sát mối tương quan tuyến tính giữa Tuổi và Cân nặng như sau:

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

```
result = scipy.stats.linregress(df['age'], df['weight'])
print(result)

LinregressResult(slope=0.2934621776596799, intercept=46.14591481573929, rvalue=0.3669636857572699, pvalue=1.691045023813897e-126, stderr=0.011824531773224177)
```

Kết quả (làm tròn) cho thấy:

- Hệ số gốc (slope) là: 0.29
- Intercept là: 46.15
- Hệ số tương quan rvalue: 0.37
- pvalue < 0.05
- stderr (sai số chuẩn): 0.01

Từ đó có thể viết công thức mối quan hệ tuyến tính giữa Tuổi và Cân nặng trong tập dữ liệu đang khảo sát như sau:

$$\text{Cân nặng} = 46.15 + 0.29 \times \text{Tuổi}$$

Điễn giải:

- Cứ một tuổi được tăng thêm thì Cân nặng tăng thêm 0.29 cm.

Phản diễn giải chi tiết thuật ngữ và cách tính sẽ được trình bày trong Chuyên đề “Hồi qui tuyến tính”.

Khảo sát mối tương quan từ DataFrame

Chọn 2 cột trong DataFrame để chuẩn bị phân tích mối tương quan như sau:

```
df1 = df[['age', 'weight']]
df1.head()
```

	age	weight
0	23	38
1	26	57
2	66	77
3	55	73
4	30	59

Gọi hàm .corr() cho DataFrame gồm 2 cột

Gọi hàm .corr() trên DataFrame df1 gồm có 2 cột và hiển thị ma trận tương quan như sau:

```
corr_matrix = df1.corr()
```

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

```
print(corr_matrix)
```

	age	weight
age	1.000000	0.366964
weight	0.366964	1.000000

Ma trận tương quan cho thấy Tuổi và Cân nặng có tương quan với nhau với hệ số là 0.37.

Gọi hàm .corr() cho DataFrame gồm nhiều cột

```
corr_matrix = df.corr()  
print(corr_matrix)
```

corr_matrix - DataFrame									
Index	id	age	sex	height	waist	risk	weight	hit	life
id	1	-0.0103087	0.00474435	-0.0109541	-0.00837234	0.00561986	-0.0219398	-0.0099051	0.0177065
age	-0.0103087	1	-0.00688092	0.0124624	0.412266	-0.0196104	0.366964	0.109705	-0.373624
sex	0.00474435	-0.00688092	1	0.698404	0.610193	-0.00587649	0.43606	0.738347	-0.181128
height	-0.0109541	0.0124624	0.698404	1	0.444762	-0.0133147	0.307404	0.516824	0.0792731
waist	-0.00837234	0.412266	0.610193	0.444762	1	-0.0135808	0.444098	0.502042	-0.286429
risk	0.00561986	-0.0196104	-0.00587649	-0.0133147	-0.0135808	1	0.0102234	0.00373656	-0.0173373
weight	-0.0219398	0.366964	0.43606	0.307404	0.444098	0.0102234	1	0.367555	-0.921325
hit	-0.0099051	0.109705	0.738347	0.516824	0.502042	0.00373656	0.367555	1	-0.183323
life	0.0177065	-0.373624	-0.181128	0.0792731	-0.286429	-0.0173373	-0.921325	-0.183323	1

Điễn giải:

- Vì giới tính (sex) về ý nghĩa thực tế là biến phân loại nên bỏ qua, không xem xét (hàm .corr ở trên chỉ tính biến có kiểu là số).
- Hệ số tương quan giữa Tuổi và Chiều cao là: 0.01; giữa Tuổi và Vòng eo là 0.41, v.v...
- Hệ số tương quan giữa Cân nặng và Vòng eo là: 0.44, v.v...
- Hệ số tương quan giữa Cân nặng và Tuổi thọ (life) là -0.92. Đây là mối tương quan nghịch. Tức là Cân nặng càng lớn thì Tuổi thọ càng kém (theo bộ dữ liệu mẫu minh họa).

Đọc thêm

<https://realpython.com/numpy-scipy-pandas-correlation-python>

Bài tập

① Cho trước DataFrame, viết đoạn chương trình để lọc các cặp biến có tương quan lớn nhất với trị số p nhỏ hơn 0.05.

- Viết hàm có dạng sau:

```
[col_x1, col_x2,...], [col_y1, col_y2,...], [r1, r2,...], [p1, p2,...] =  
    find_corellation(df)
```

Hàm sẽ được gọi như sau:

```
df = pd.read_csv(file)  
X, Y, R, P = find_corellation(df)
```

Trong đó:

- X, Y, R, P: là các array.
- X: là array chứa các tên cột
- Y: là array chứa các tên cột mà tại vị trí i thì X[i] có tương quan với Y[i] với hệ số là R[i] và p-value là P[i]

Nâng cao – Viết hàm tự tìm mối tương quan trong DataFrame

Phân tích đoạn chương trình sau:

```
import math  
import pandas as pd  
from scipy.stats import linregress  
  
df = pd.read_csv('data.csv')  
  
df.columns  
df.info()  
  
df.describe()  
  
def find_corellation(df):  
    corr_matrix = df.corr()  
    attr_names = corr_matrix.columns  
  
    arr_key = []  
    arr_col1 = []  
    arr_col2 = []  
    arr_r = []  
    arr_rlin = []  
    arr_pvalue = []  
    for attr_name1 in attr_names:  
        # Get row of correlation matrix by index (attribute name)  
        # row_matrix: Series with index is attribute name  
        row_matrix = corr_matrix.loc[attr_name1]  
  
        # Scan each value of row_matrix  
        for attr_name2 in attr_names:
```

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

```
# Skip some column: .._id
if attr_name1.endswith('_id') or attr_name2.endswith('_id'):
    continue

if attr_name1 != attr_name2:
    r = row_matrix[attr_name2]
    if not math.isnan(r):
        # print('%s <-> %s: %s' % (attr_name1, attr_name2, r))

        # Check linear regression
        corr_lin = linregress(df[attr_name1], df[attr_name2])
        if (corr_lin.pvalue < 0.05):
            print('Correlation between "%s" and "%s", r=%s' %
(attrib_name1, attrib_name2, corr_lin.rvalue))

        if (attrib_name2 + attrib_name1) not in arr_key:
            arr_key.append(attrib_name1 + attrib_name2)

            arr_col1.append(attrib_name1)
            arr_col2.append(attrib_name2)
            arr_rlin.append(corr_lin.rvalue)
            arr_r.append(r)
            arr_pvalue.append(corr_lin.pvalue)

# Make datafram of correlation
df_corr = pd.DataFrame(list(zip(arr_col1, arr_col2, arr_rlin, arr_r,
arr_pvalue)), columns=['attr1', 'attr2', 'rlin', 'r', 'pvalue'])

return df_corr

df_co = find_correlation(df)
```

Ngày 4 – Chủ đề: Dữ liệu lớn

Ngày hôm nay sẽ bàn tùng huống là bạn có quá nhiều dữ liệu thì sao? Bạn biết là cái máy tính bạn đang dùng thì có giới hạn của nó. Trong lĩnh vực IT nói chung là mọi thứ đều có giới hạn như máy bao nhiêu RAM, đĩa cứng bao nhiêu GB, TB, tốc độ CPU bao nhiêu MHz đều có giới hạn hết. Cụ thể một file CSV được lưu trên ổ cứng có giới hạn tối đa là dung lượng còn trống của ổ cứng hoặc giới hạn về kích thước file mà hệ điều hành (Windows, MacOS, Linux) hỗ trợ.

Vì thế các bài trong ngày này sẽ giúp chúng ta một vài giải pháp để lưu trữ dữ liệu lớn và làm quen với vài thao tác truy cập dữ liệu bằng R và Python.

Ngày thứ tư này sẽ gồm 4 bài:

Bài 18: Giới thiệu cách lưu trữ dữ liệu trong các phần mềm quản lý dữ liệu chuyên nghiệp, minh họa bằng phần mềm mã nguồn mở¹⁶ MySQL. Sau đó mở rộng vấn đề lưu trữ dữ liệu lớn bằng phần mềm Hadoop, cũng là phần mềm mở. Đặc biệt gợi mở cho các bạn không phải là người lập trình có thể trải nghiệm thêm bằng cách sử dụng máy ảo trên máy thật. Ngoài ra trải nghiệm một hệ điều hành mới, Ubuntu, rất được nhiều người trong giới làm về Data Science, AI nói chung, Machine Learning nói riêng sử dụng.

Bài 19: Hướng dẫn sử dụng Ubuntu cơ bản cho người chưa biết.

Bài 20: Hướng dẫn cài đặt Hadoop để có thể làm quen, trải nghiệm.

Bài 21: Hướng dẫn dùng R và Python để khai thác dữ liệu trên Hadoop.

Ngày này mang tính kỹ thuật hơi nhiều. Sẽ hơi khó khăn cho các bạn không phải là người học chuyên sâu về Công Nghệ Thông Tin. Tuy nhiên tôi cho rằng các bạn xứng đáng để trải nghiệm một chút về kỹ thuật. Nó cũng không quá khó đâu. Đặc biệt có trải nghiệm một chút để dễ “nói chuyện” với các đồng nghiệp đảm trách về IT, hệ thống.

¹⁶ Phần mềm mở (open-source) được hiểu là miễn phí, các nhân được quyền sử dụng và phát triển tiếp. Đối với tổ chức thương mại thì tùy nhu cầu sẽ đọc kỹ giấy phép sử dụng mã nguồn của tác giả, nhóm tác giả của phần mềm.

Bài 18: Cách xử lý tập hợp dữ liệu lớn

Đến thời điểm này bạn đã làm quen với việc đọc dữ liệu từ file CSV, một dạng file văn bản rất đơn giản lưu trữ dữ liệu thành các cột, cách nhau bởi dấu phẩy, hoặc file TXT tương tự trong đó các cột cách nhau bởi dấu chấm phẩy. Bài này sẽ giúp các bạn hình dung ra cách lưu trữ dữ liệu lớn hơn. Phần này liên quan nhiều đến kỹ thuật và công nghệ một chút nhưng tôi sẽ cố gắng trình bày một cách đơn giản để bạn hình dung. Đặc biệt các bạn làm việc trong môi trường đội nhóm, hoặc có đội ngũ dưới quyền thì cũng biết đồng nghiệp, nhân viên mình tổ chức dữ liệu hỗ trợ mình như thế nào!

Lưu trữ dữ liệu trong các phần mềm chuyên dụng

Đối với file CSV hoặc TXT mà bạn được làm quen và thực hành thì bạn dễ dàng xem dữ liệu bằng các phần mềm như Notepad, hoặc Nodepad++ mà tôi giới thiệu trong Bài 5. Xin xò hơn một chút thì mở bằng phần mềm dạng bảng tính (Spreadsheet) như Microsoft Excel, Open Office Spreadsheet.

Trong các hệ thống xử lý thông tin thì sẽ có các phần mềm chuyên dụng để lưu trữ, quản lý dữ liệu chặt chẽ hơn. Thông thường dữ liệu mà các bạn xử lý là dữ liệu dạng bảng (Table), trong đó gồm các cột (Column, hoặc Field – trường dữ liệu). Loại dữ liệu này rất thích hợp để lưu trữ trong các phần mềm quản trị cơ sở dữ liệu (DBMS – Database Management System). Các dạng phần mềm DBMS phổ biến và nổi tiếng như Oracle, Microsoft SQL Server, IBM DB2. Nhìn chung thì các phần mềm này có bản quyền. Nếu tiết kiệm chi phí thì có thể dùng MySQL, PostgreSQL. Phần tiếp theo tôi sẽ minh họa cho các bạn cách sử dụng MySQL để lưu trữ dữ liệu và thực hiện thao tác đọc dữ liệu để phân tích. Trong quá trình trình bày, tôi sẽ cô đọng các ý tưởng chính và bạn có thể áp dụng cho các phần mềm tương tự.

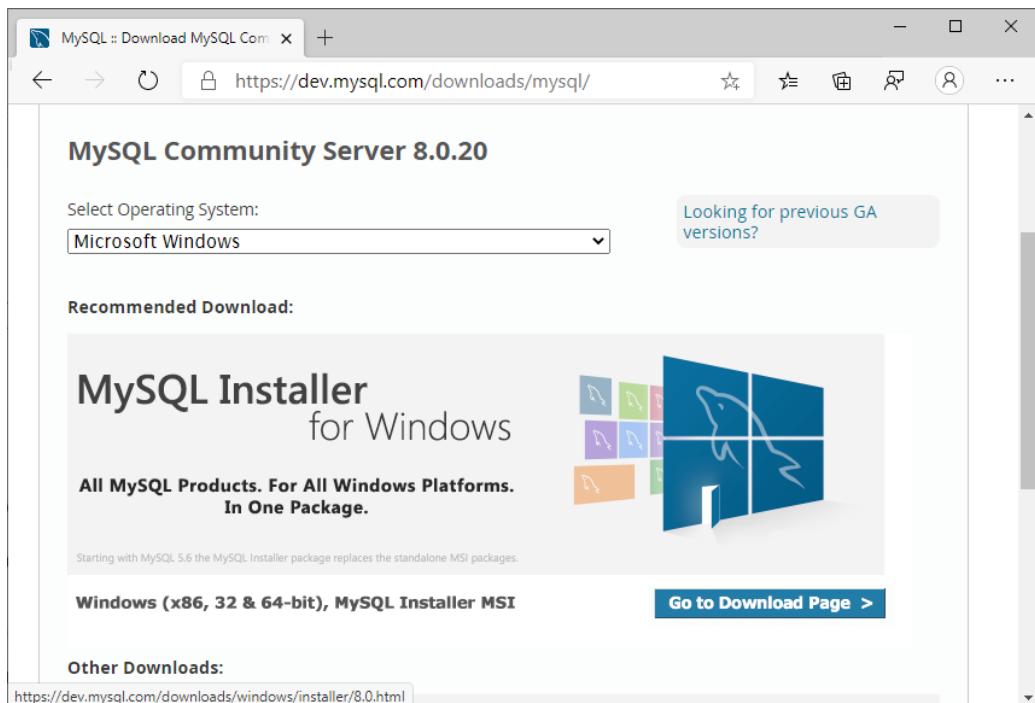
Sử dụng MySQL

Để các bạn có trải nghiệm nhanh thì chúng ta đi vào cài đặt và sử dụng luôn.

Cài đặt

Bạn vào trang web <https://dev.mysql.com/downloads/mysql/> để tải phiên bản MySQL Community Server mới nhất (hiện tại là bản 8.0.20)

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



MySQL :: Download MySQL Com x +

← → ⏪ https://dev.mysql.com/downloads/mysql/ ⏩ ...

MySQL Community Server 8.0.20

Select Operating System:

Microsoft Windows

Looking for previous GA versions?

Recommended Download:

MySQL Installer for Windows

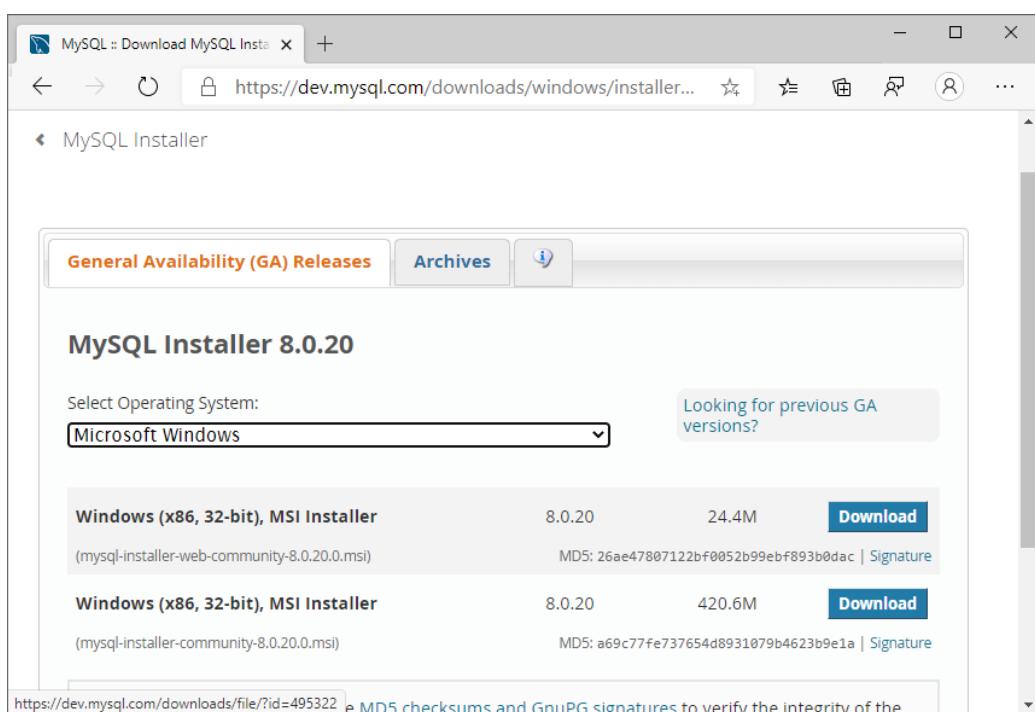
All MySQL Products. For All Windows Platforms.
In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

Go to Download Page >

Other Downloads:
<https://dev.mysql.com/downloads/windows/installer/8.0.html>



MySQL :: Download MySQL Insta x +

← → ⏪ https://dev.mysql.com/downloads/windows/installer... ⏩ ...

MySQL Installer

General Availability (GA) Releases Archives ⓘ

MySQL Installer 8.0.20

Select Operating System:

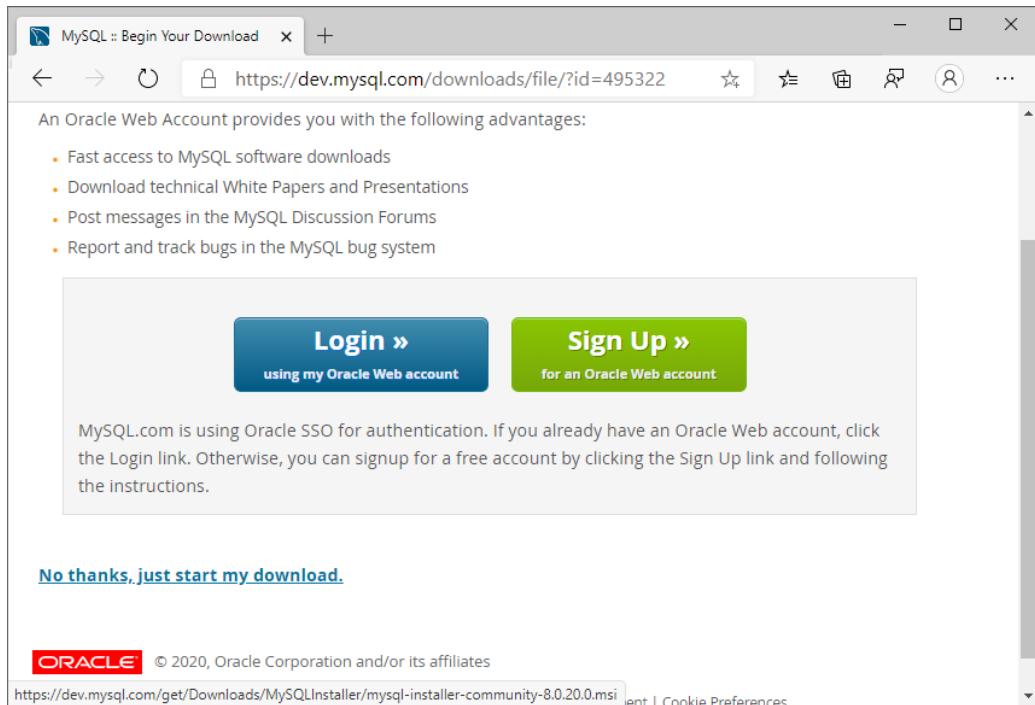
Microsoft Windows

Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer	8.0.20	24.4M	Download
(mysql-installer-web-community-8.0.20.0.msi)			MD5: 26ae47807122bf0052b99ebf893b0dac Signature
Windows (x86, 32-bit), MSI Installer	8.0.20	420.6M	Download
(mysql-installer-community-8.0.20.0.msi)			MD5: a69c77fe737654d8931079b4623b9e1a Signature

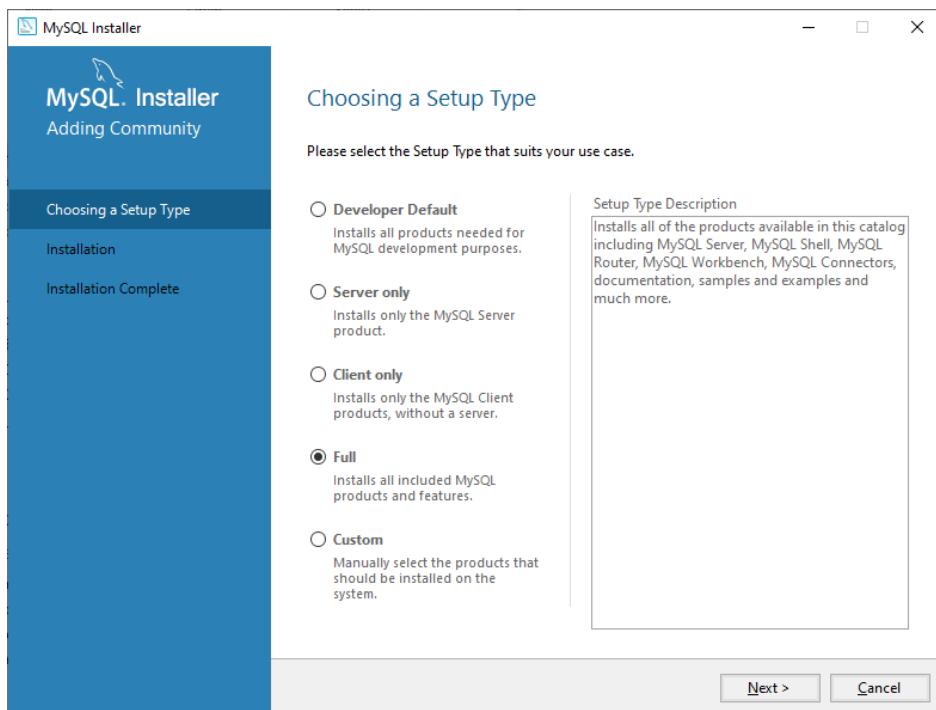
https://dev.mysql.com/downloads/file/?id=495322 e [MD5 checksums and GnuPG signatures](#) to verify the integrity of the

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



Sau khi tải file “mysql-installer-community-8.0.20.0.msi” về máy, bạn double-click vào nó để cài đặt.

Trong màn hình đầu tiên, bạn cho chế độ cài Full (đầy đủ) để làm quen.



Các màn hình tiếp theo bạn nhấn Next và Yes.

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

The screenshot shows the MySQL Installer interface. On the left, a sidebar lists steps: 'Choosing a Setup Type' (selected), 'Check Requirements' (highlighted in blue), 'Installation', 'Product Configuration', and 'Installation Complete'. The main panel title is 'Check Requirements'. It contains a message: 'The following products have failing requirements. MySQL Installer will attempt to resolve them automatically. Requirements marked as manual cannot be resolved automatically. Click on each item to try and resolve it manually.' Below this is a table:

For Product	Requirement	Status
MySQL For Excel 1.3.8	Visual Studio 2010 Tools for Office R...	

At the bottom are buttons: '< Back', 'Execute', 'Next >', and 'Cancel'.

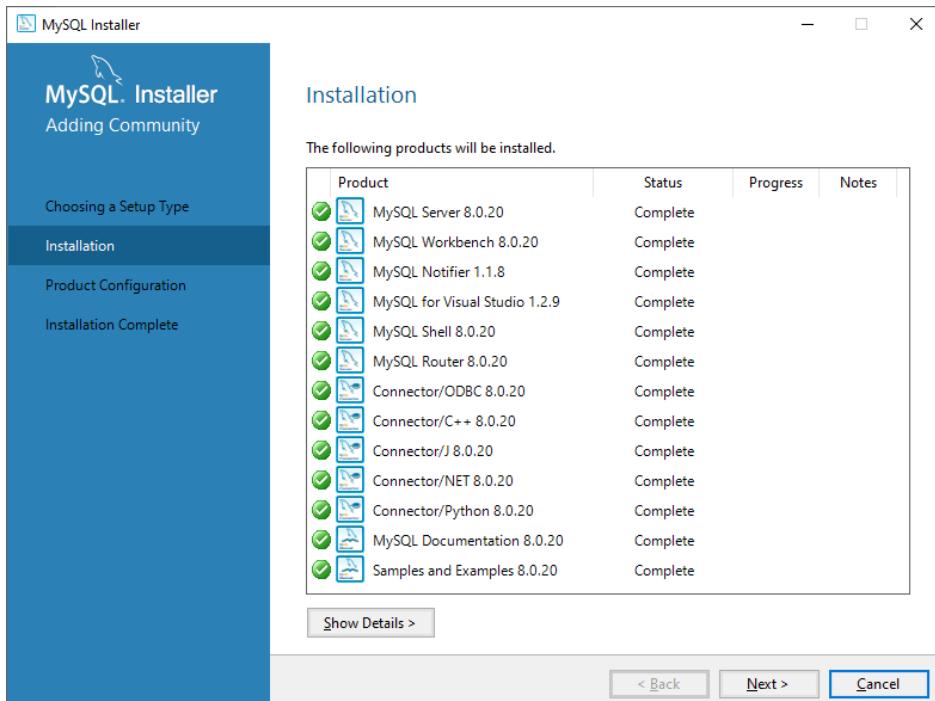
A modal dialog box titled 'MySQL Installer' displays a warning: 'One or more product requirements have not been satisfied'. It says: 'Those products with missing requirements will not be installed or upgraded. Do you want to continue?'. There are 'Yes' and 'No' buttons at the bottom.

The screenshot shows the MySQL Installer interface again. The sidebar is identical. The main panel title is 'Installation'. It contains a message: 'The following products will be installed.' Below this is a table:

Product	Status	Progress	Notes
MySQL Server 8.0.20	Ready to Install		
MySQL Workbench 8.0.20	Ready to Install		
MySQL Notifier 1.1.8	Ready to Install		
MySQL for Visual Studio 1.2.9	Ready to Install		
MySQL Shell 8.0.20	Ready to Install		
MySQL Router 8.0.20	Ready to Install		
Connector/ODBC 8.0.20	Ready to Install		
Connector/C++ 8.0.20	Ready to Install		
Connector/J 8.0.20	Ready to Install		
Connector/.NET 8.0.20	Ready to Install		
Connector/Python 8.0.20	Ready to Install		
MySQL Documentation 8.0.20	Ready to Install		
Samples and Examples 8.0.20	Ready to Install		

At the bottom are buttons: '< Back', 'Execute', and 'Cancel'.

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

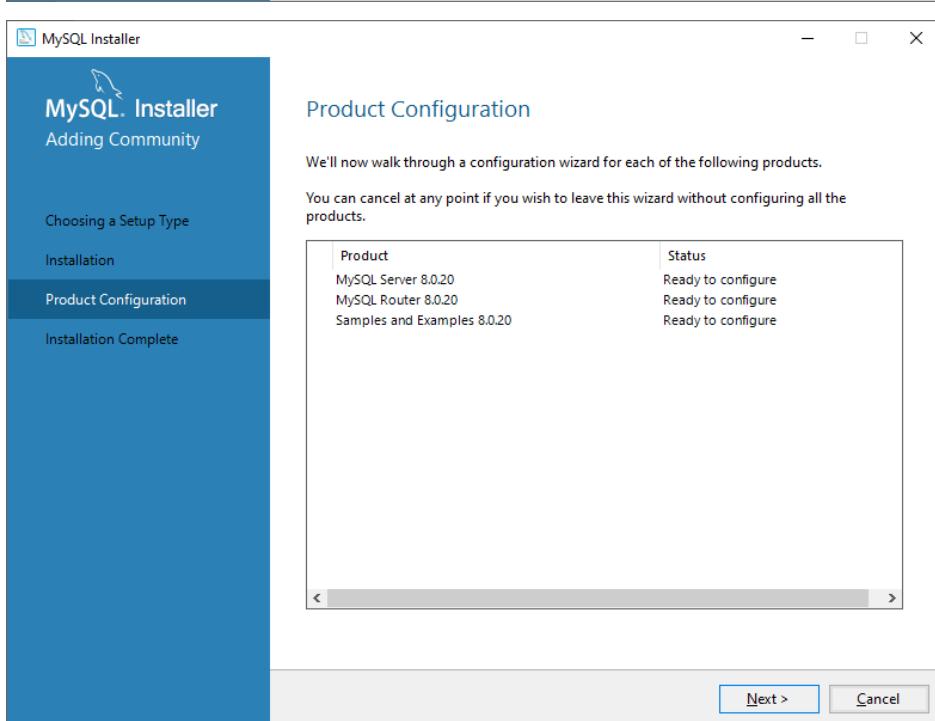


The screenshot shows the MySQL Installer window titled "Adding Community". The left sidebar lists "Choosing a Setup Type", "Installation" (which is selected), "Product Configuration", and "Installation Complete". The main area is titled "Installation" and displays a table of installed products:

Product	Status	Progress	Notes
MySQL Server 8.0.20	Complete		
MySQL Workbench 8.0.20	Complete		
MySQL Notifier 1.1.8	Complete		
MySQL for Visual Studio 1.2.9	Complete		
MySQL Shell 8.0.20	Complete		
MySQL Router 8.0.20	Complete		
Connector/ODBC 8.0.20	Complete		
Connector/C++ 8.0.20	Complete		
Connector/J 8.0.20	Complete		
Connector/.NET 8.0.20	Complete		
Connector/Python 8.0.20	Complete		
MySQL Documentation 8.0.20	Complete		
Samples and Examples 8.0.20	Complete		

[Show Details >](#)

< Back [Next >](#) Cancel



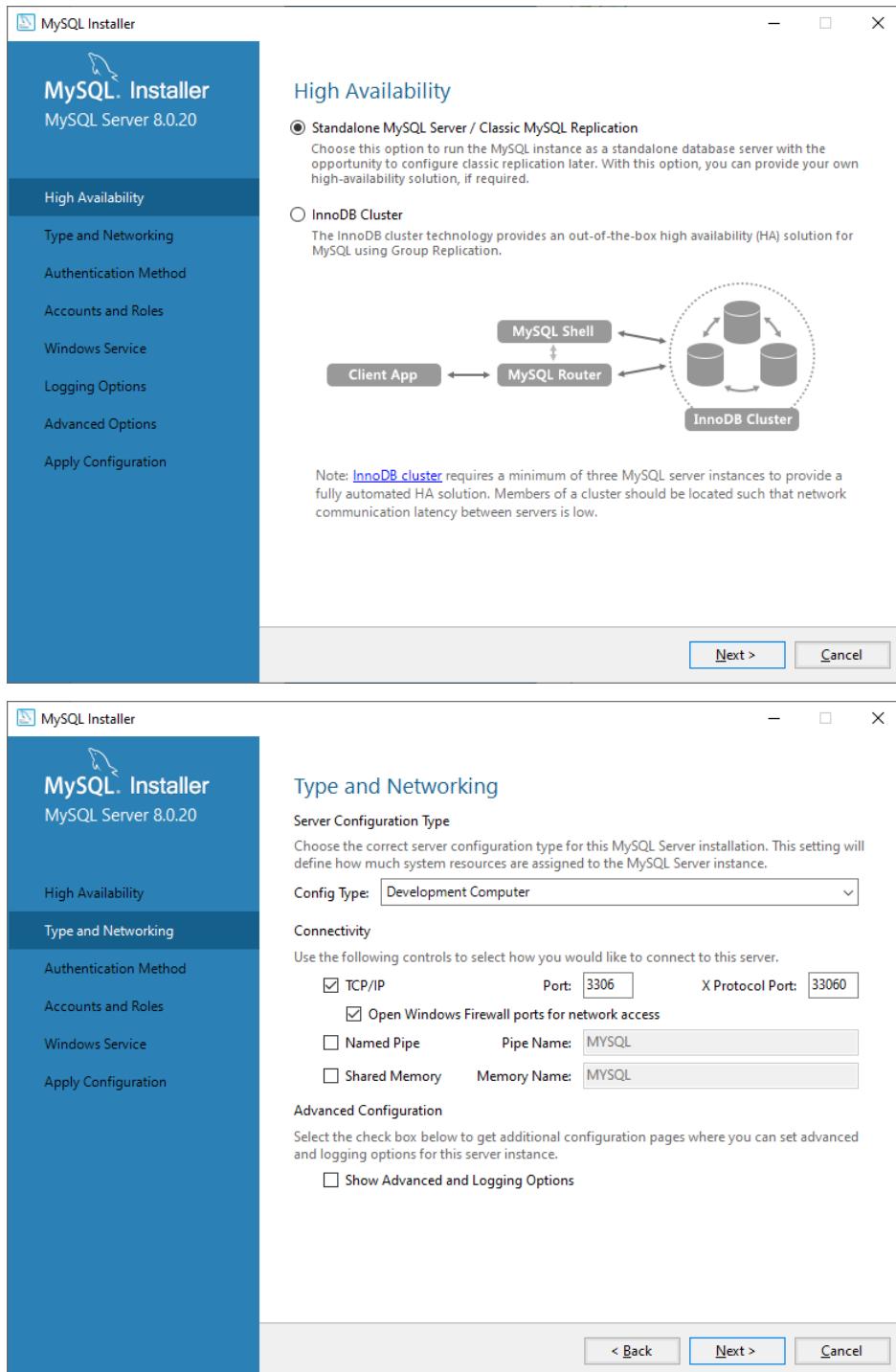
The screenshot shows the MySQL Installer window titled "Adding Community". The left sidebar lists "Choosing a Setup Type", "Installation" (which is selected), "Product Configuration" (which is selected), and "Installation Complete". The main area is titled "Product Configuration" and displays the following text:
We'll now walk through a configuration wizard for each of the following products.
You can cancel at any point if you wish to leave this wizard without configuring all the products.

Product	Status
MySQL Server 8.0.20	Ready to configure
MySQL Router 8.0.20	Ready to configure
Samples and Examples 8.0.20	Ready to configure

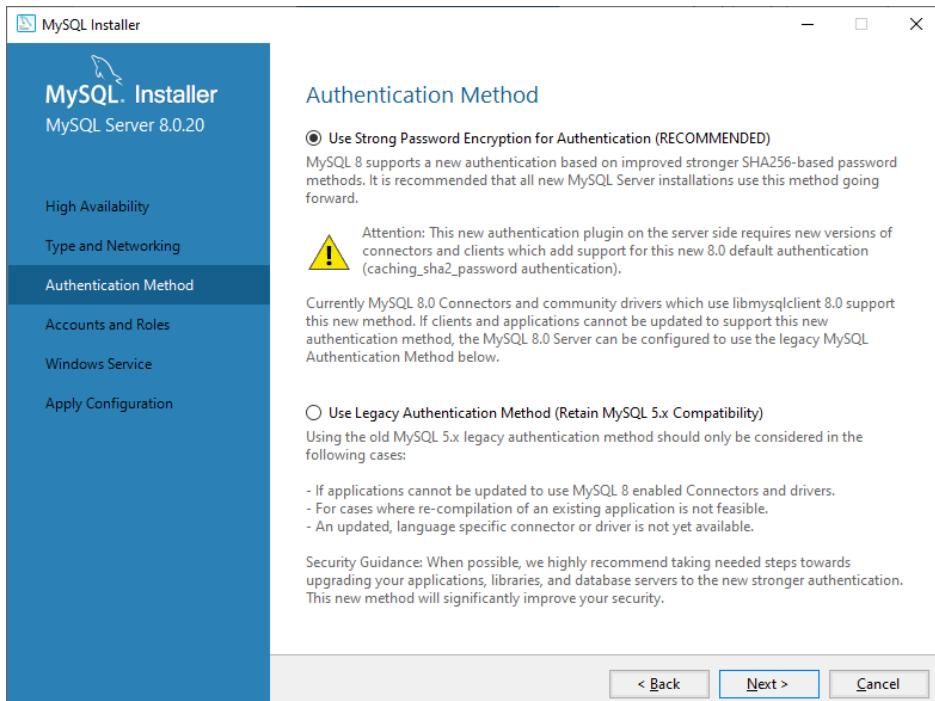
< >

[Next >](#) Cancel

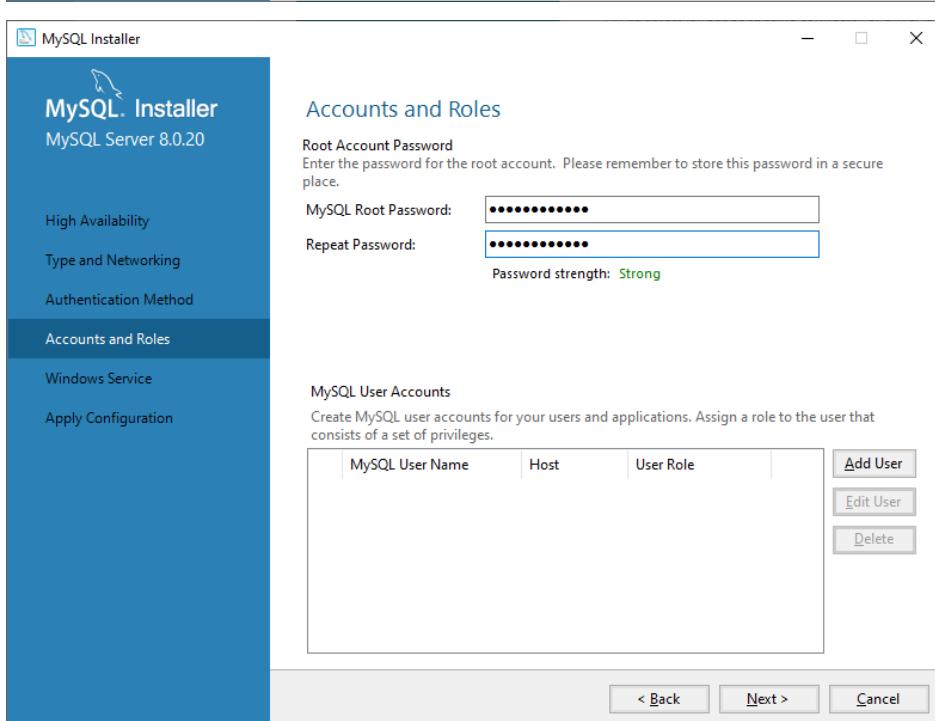
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



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

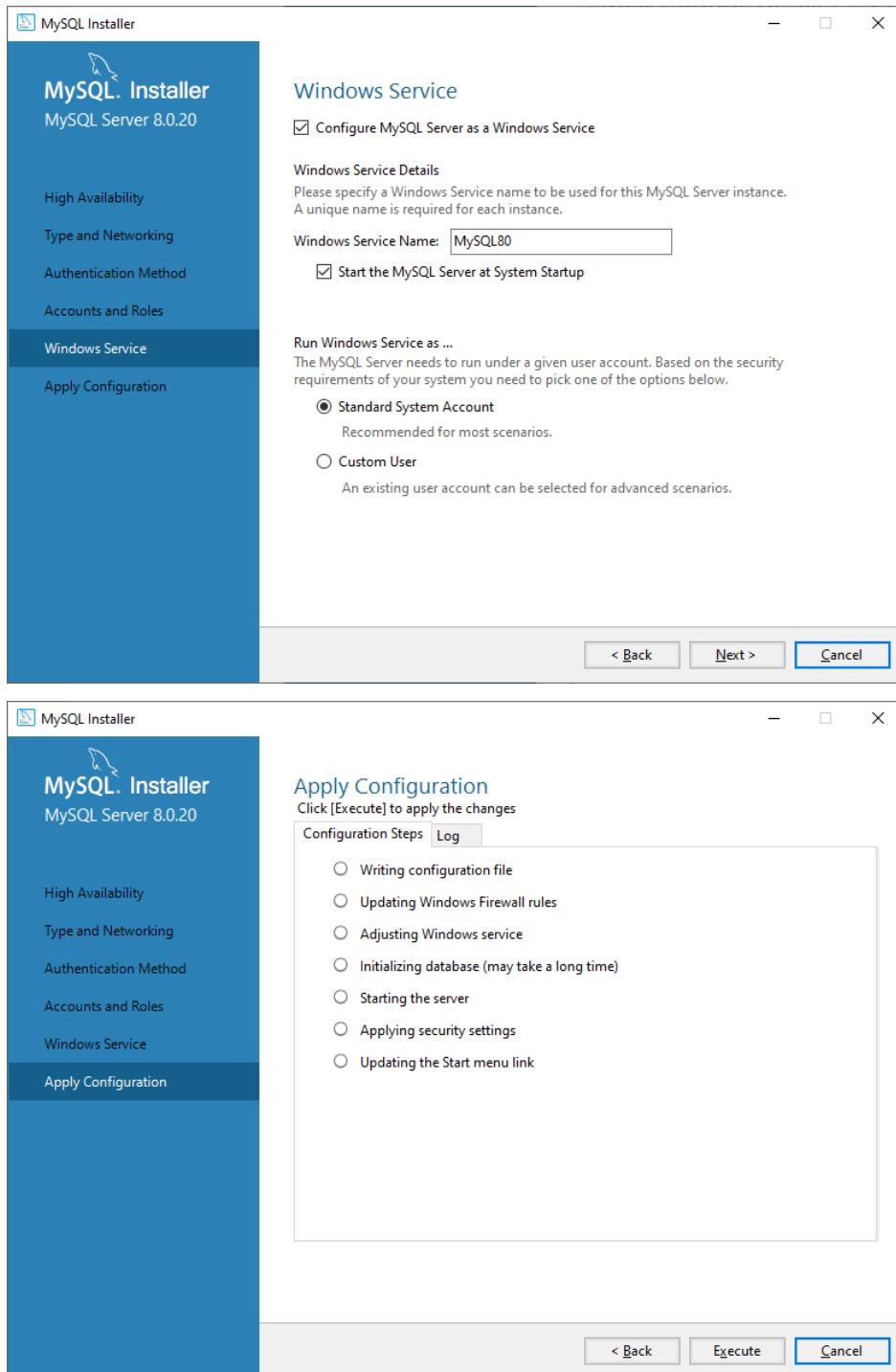


The screenshot shows the MySQL Installer for MySQL Server 8.0.20. The left sidebar has tabs: High Availability, Type and Networking, Authentication Method (which is selected), Accounts and Roles, Windows Service, and Apply Configuration. The main panel title is "Authentication Method". It contains two radio button options: "Use Strong Password Encryption for Authentication (RECOMMENDED)" (selected) and "Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)". A note states: "MySQL 8 supports a new authentication based on improved stronger SHA256-based password methods. It is recommended that all new MySQL Server installations use this method going forward." A warning icon indicates: "Attention: This new authentication plugin on the server side requires new versions of connectors and clients which add support for this new 8.0 default authentication (caching_sha2_password authentication)." Another note says: "Currently MySQL 8.0 Connectors and community drivers which use libmysqlclient 8.0 support this new method. If clients and applications cannot be updated to support this new authentication method, the MySQL 8.0 Server can be configured to use the legacy MySQL Authentication Method below." Below these are "Security Guidance" notes and "Security Best Practices". At the bottom are buttons: < Back, Next >, and Cancel.

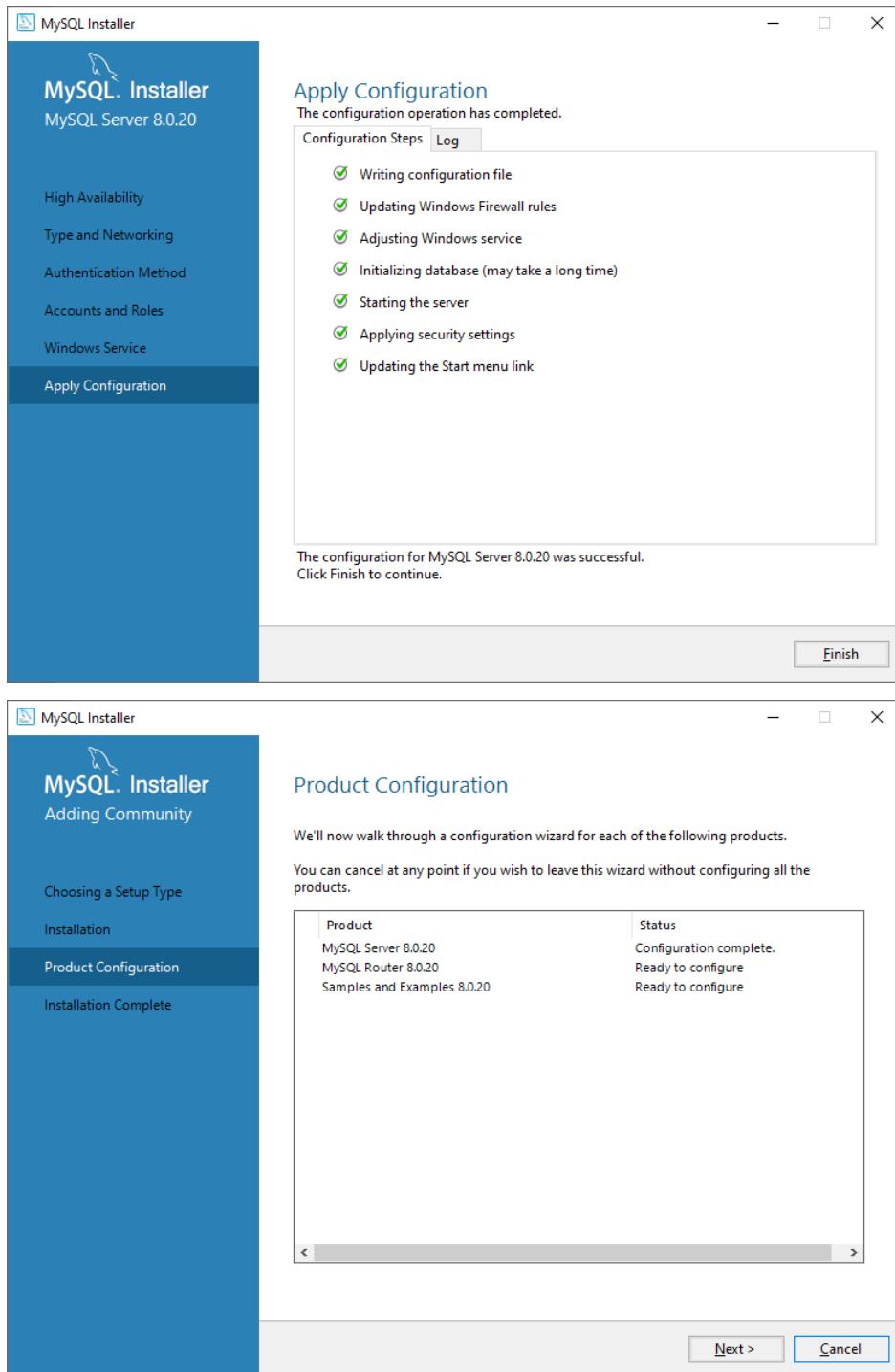


The screenshot shows the MySQL Installer for MySQL Server 8.0.20. The left sidebar has tabs: High Availability, Type and Networking, Authentication Method (selected), Accounts and Roles, Windows Service, and Apply Configuration. The main panel title is "Accounts and Roles". It includes sections for "Root Account Password" (with fields for MySQL Root Password and Repeat Password, both masked) and "MySQL User Accounts" (with a table for adding users and buttons for Add User, Edit User, and Delete). At the bottom are buttons: < Back, Next >, and Cancel.

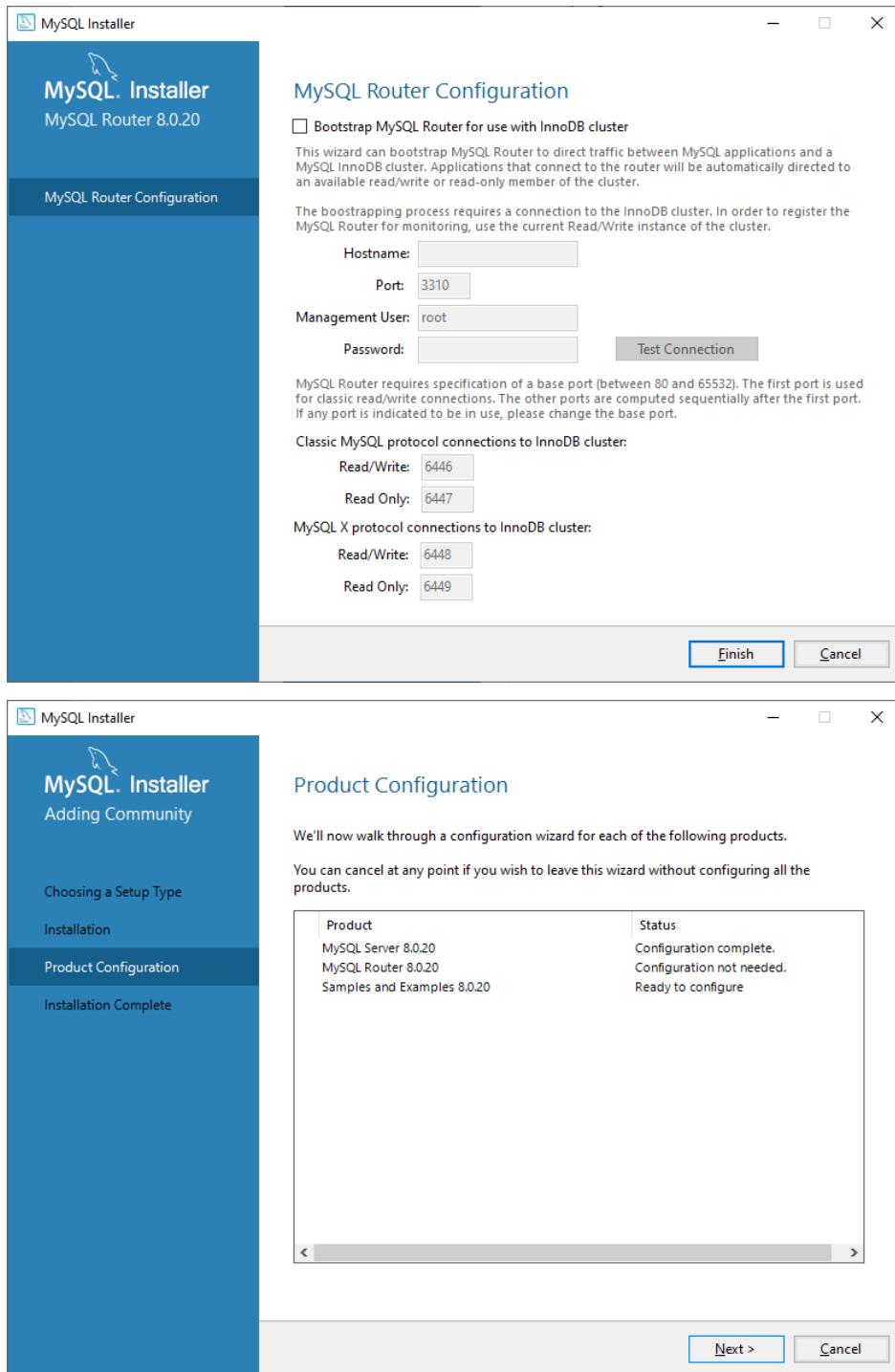
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



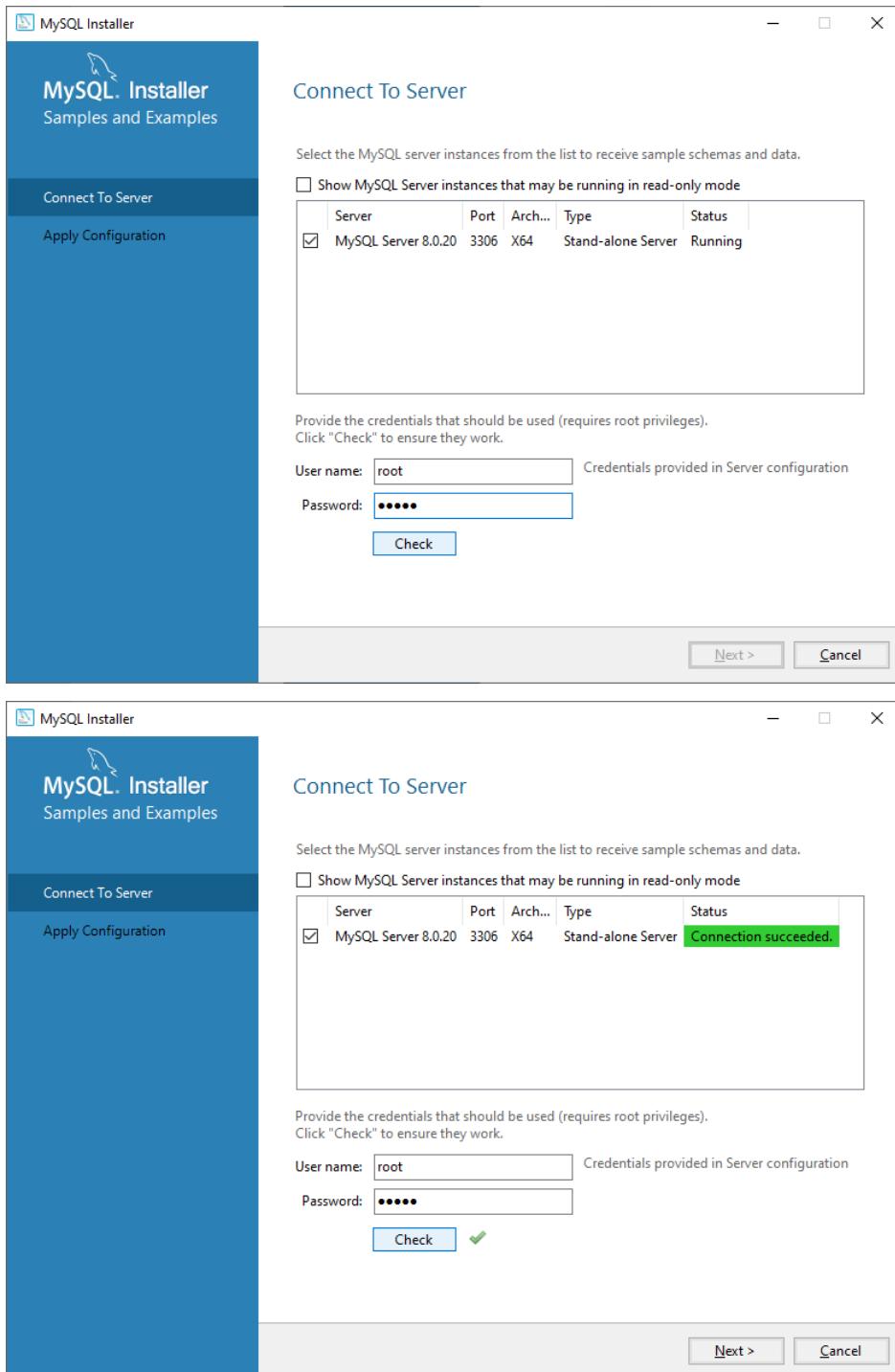
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



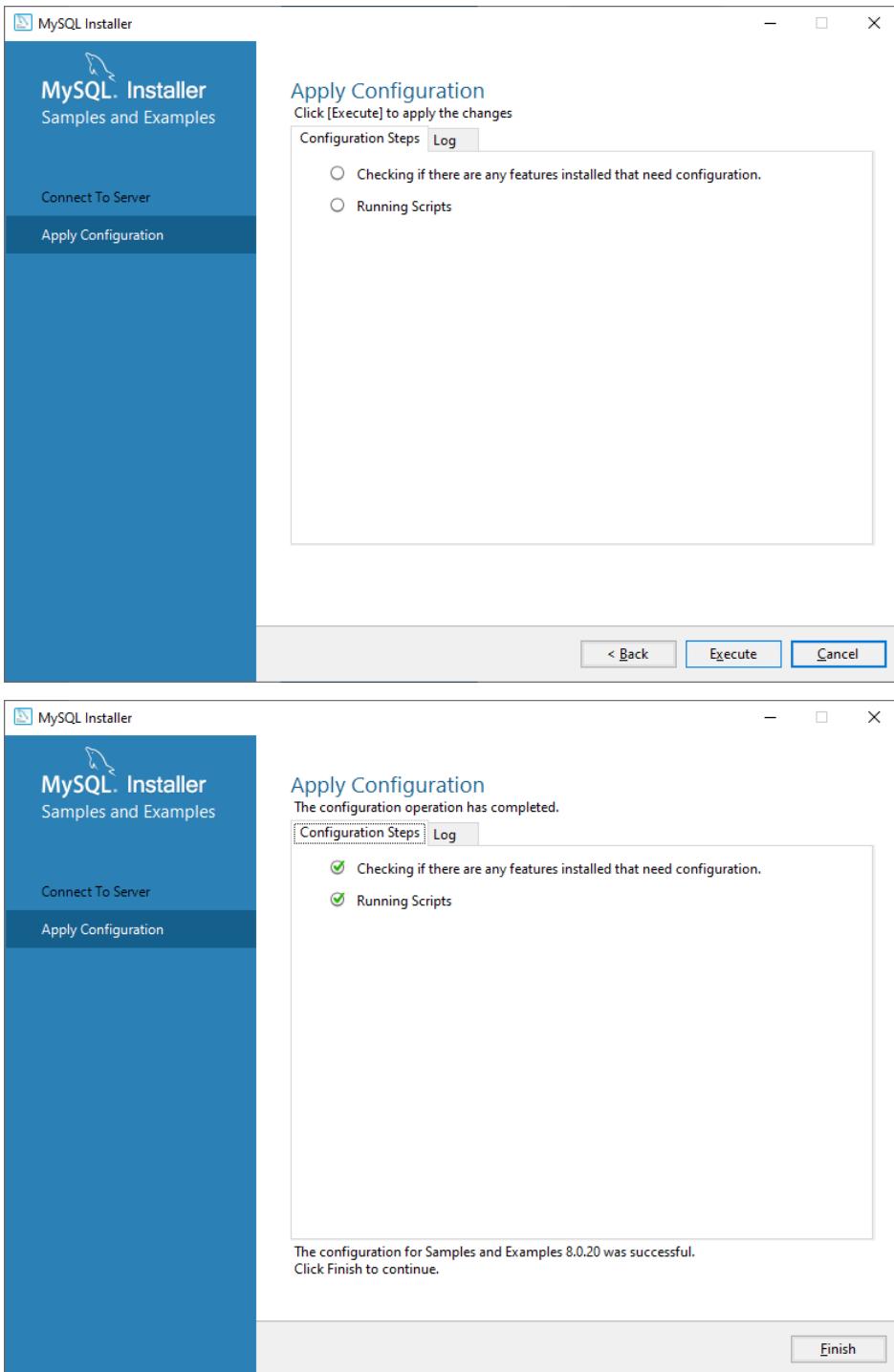
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



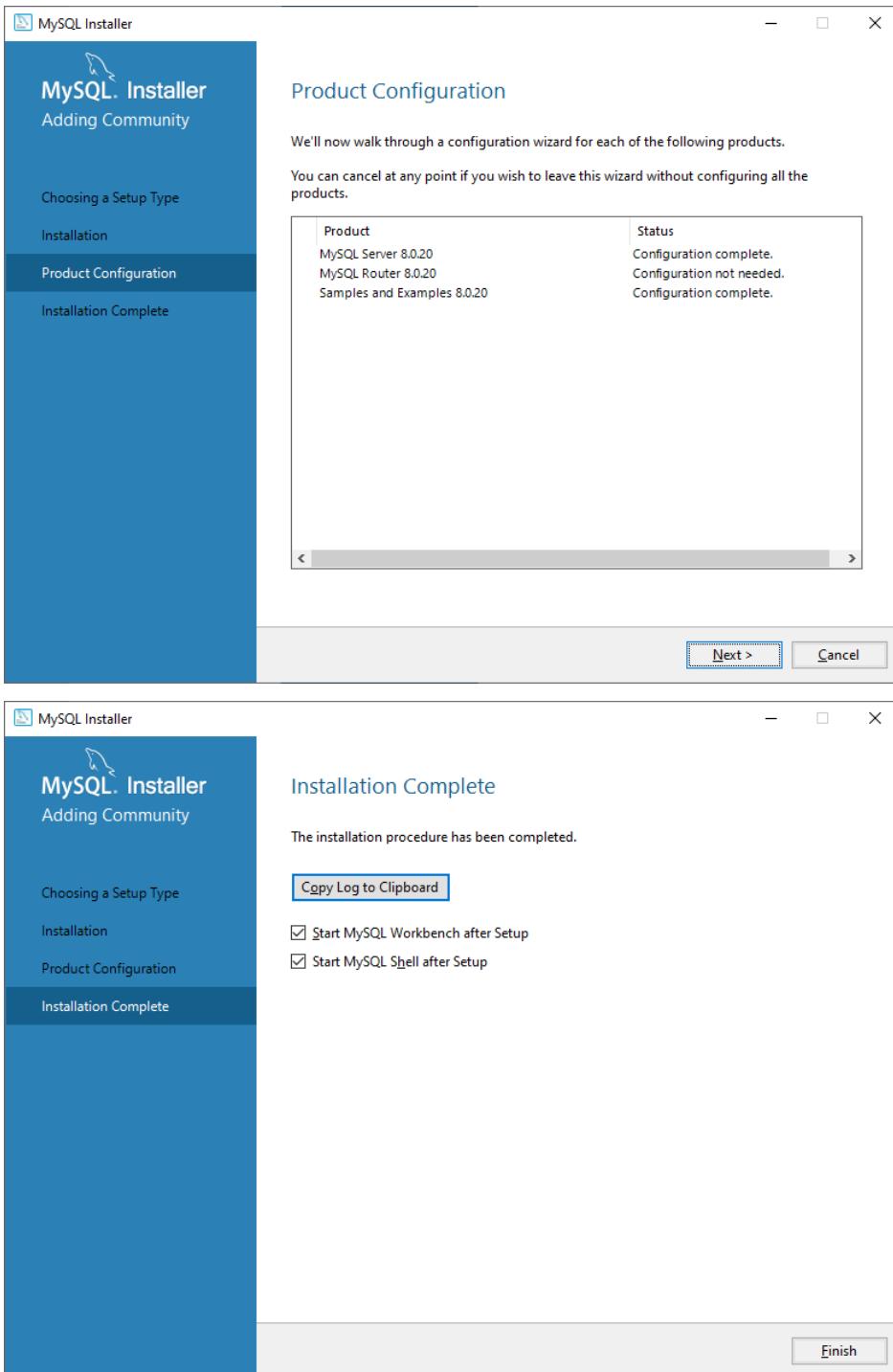
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



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

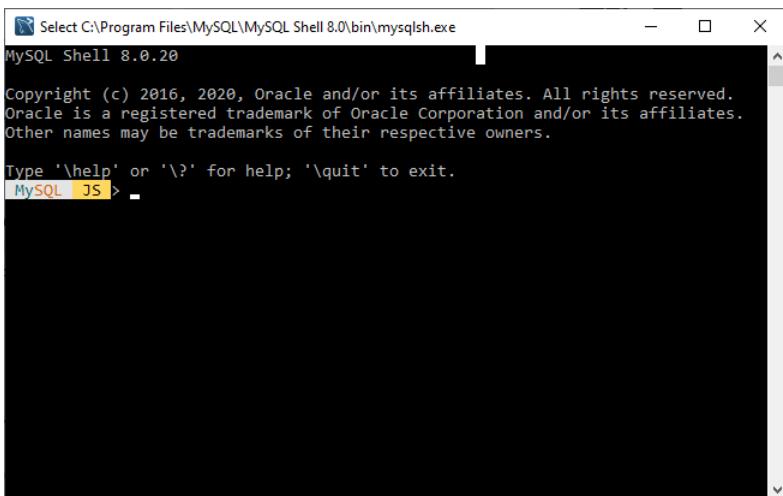


Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

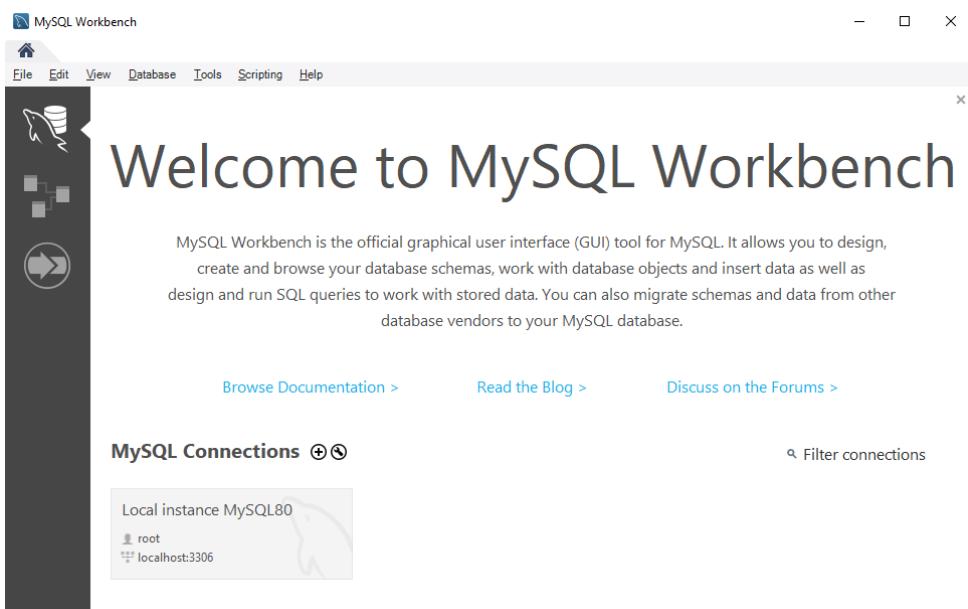


Sau khi bấm Finished thì 2 cửa sổ tương ứng của 2 chức năng trong MySQL sẽ hiển thị ra như sau:

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



Gõ \quit và nhấn Enter để thoát khỏi cửa sổ màu đen này. Chúng ta sẽ sử dụng sau.

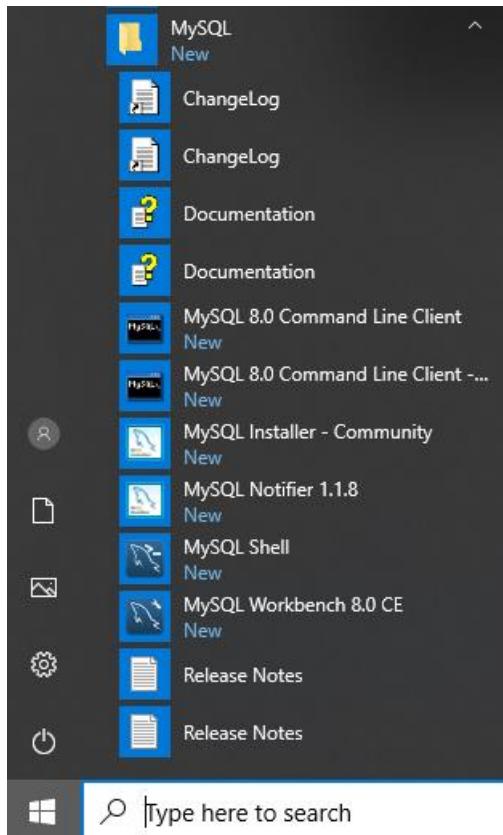


Bạn bấm nút x ở góc phải trên để thoát khỏi công cụ MySQL Workbench này.

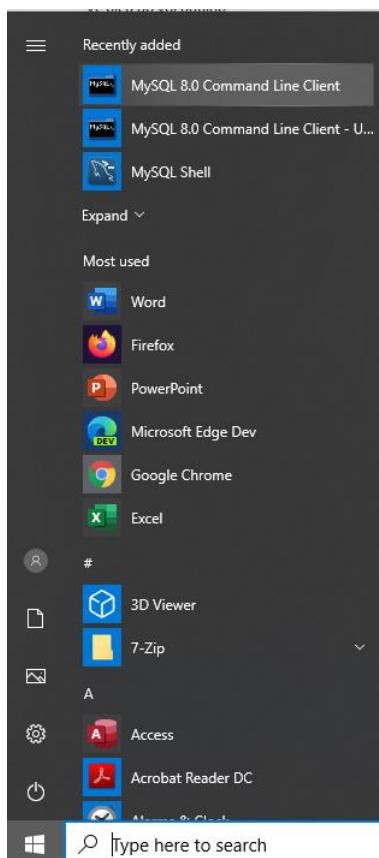
Khởi động MySQL Command Line

Sau khi cài đặt xong thì trong nút Start của Windows, bạn sẽ thấy các menu của phần mềm MySQL như sau.

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



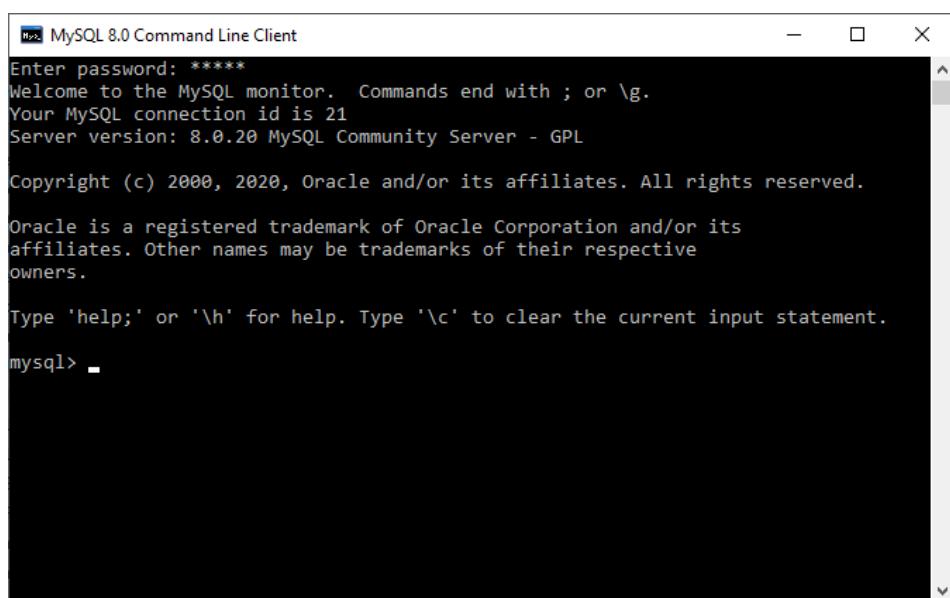
Hoặc trong nút Start có mục Recently added như bên dưới.



Tạo cơ sở dữ liệu với MySQL Command Line

Để tương tác với hệ quản trị CSDL MySQL thì có rất nhiều cách, ở đây tôi sẽ hướng dẫn bạn cách gõ lệnh để trải nghiệm một chút. Thông thường gõ lệnh sẽ là một cực hình đối với các bạn không quen. Tuy nhiên ở đây tôi sẽ làm sẵn các lệnh mà bạn sau này có thể chỉnh chỉnh sửa, copy & paste cho nhu cầu tương tự của mình sau này.

Mở cửa sổ gõ lệnh MySQL bằng cách vào nút Start của Windows, tìm shorcut “MySQL 8.0 Command Line Client” hoặc “MySQL 8.0 Command Line Client Unicode” nếu có làm việc liên quan đến tiếng Việt (không phải tiếng Anh nói chung). Cửa sổ hiện ra yêu cầu bạn gõ mật khẩu. Đây chính là mật khẩu tài khoản root mà bạn đã nhập trong lúc cài đặt. Sau khi gõ mật khẩu và nhấn Enter, cửa sổ lệnh của mysql hiện ra như sau:



The screenshot shows the MySQL 8.0 Command Line Client window. The title bar says "MySQL 8.0 Command Line Client". The main area displays the following text:
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.20 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> -

Có vài lệnh được gợi ý bạn có thể làm quen:

- **help;** hoặc **\h** để hiện ra hướng dẫn.
- **exit** để thoát của sổ lệnh này.

Thông thường thì để kết thúc một lệnh gõ dấu chấm phẩy ;

Nhấn Enter để thực hiện lệnh.

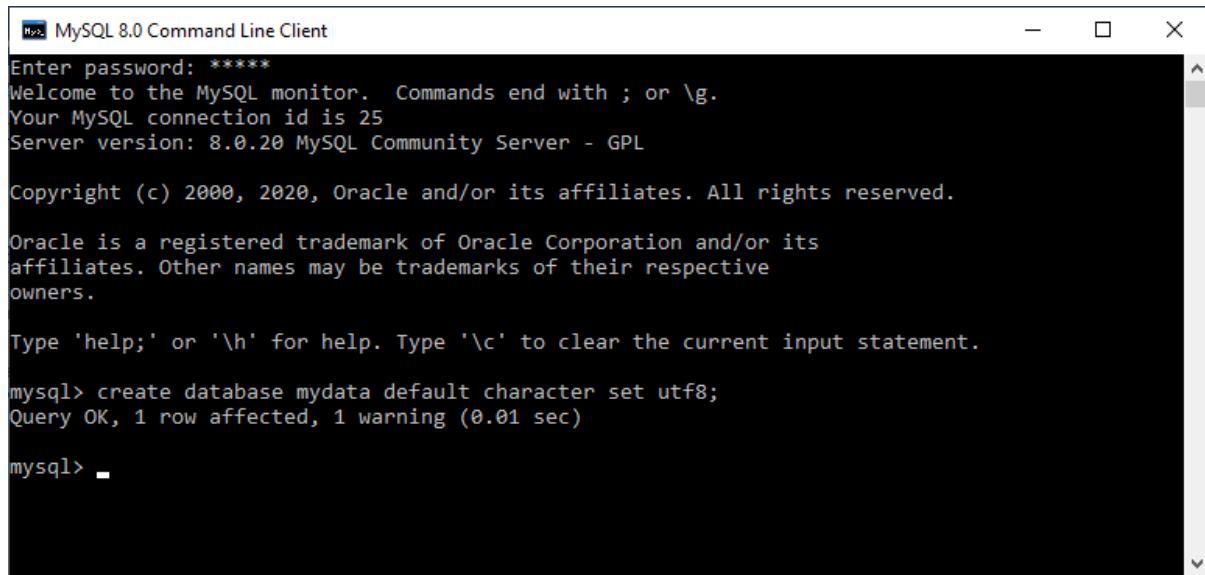
Tạo cơ sở dữ liệu

Lệnh thứ nhất bạn cần làm quen là tạo ra một cơ sở dữ liệu.

```
create database mydata default character set utf8;
```

Bạn có thể copy dòng lệnh ở trên. Sau đó dán (paste) vào cửa sổ lệnh mysql bằng cách nhấp phải chuột bên trong cửa sổ màu đen. Sau đó nhấn Enter để thực hiện lệnh.

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



```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.20 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

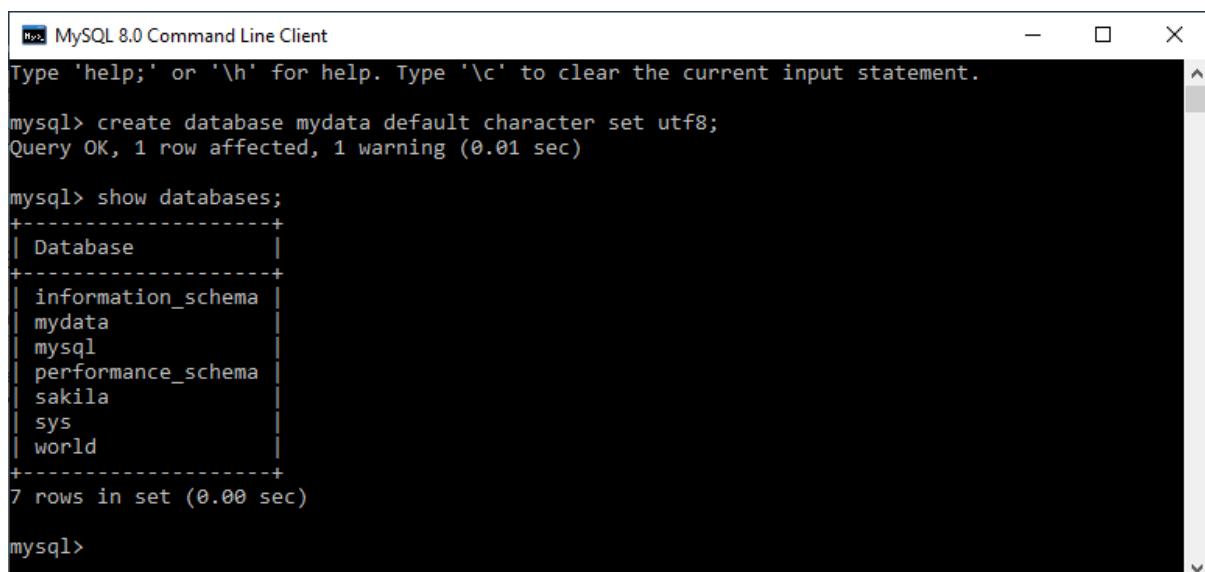
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database mydata default character set utf8;
Query OK, 1 row affected (0.01 sec)

mysql>
```

Lệnh tiếp theo là “show databases” để xem các cơ sở dữ liệu đang có trong máy.

show databases;



```
MySQL 8.0 Command Line Client
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database mydata default character set utf8;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mydata        |
| mysql          |
| performance_schema |
| sakila         |
| sys            |
| world          |
+-----+
7 rows in set (0.00 sec)

mysql>
```

Ngoài database “mydata” là do bạn tạo thì các database còn lại của hệ thống MySQL. Tạm thời bạn không cần quan tâm, và không được dùng vào các database này.

Lệnh tiếp theo là “drop database mydata” để xóa database “mydata” mà bạn vừa mới tạo:

drop database mydata;

Hãy gõ lại lệnh show databases và create database... để quen tay.

Khởi tạo lại cơ sở dữ liệu:

create database mydata default character set utf8;

Tạo tài khoản

Khởi tạo tài khoản cho cơ sở dữ liệu:

```
CREATE USER user_mysql@'localhost' IDENTIFIED WITH  
mysql_native_password BY 'mysqlpass';
```

Gán quyền cho tài khoản

Gán quyền cho user để truy cập cơ sở dữ liệu:

```
grant all on mydata.* to user_mysql@localhost;
```

Thiết lập MySQL cho phép nạp dữ liệu

```
SET GLOBAL local_infile=1;
```

Truy cập MySQL bằng Python

Một trong các thư viện hỗ trợ kết nối Python với MySQL là

<https://pythontic.com/database/mysql>

Cài đặt thư viện:

```
pip install pymysql
```

Code Python để khai báo thư viện và mở kết nối

```
from sqlalchemy import create_engine  
connection = 'mysql+pymysql://username:password@server/dbname'  
sqlEngine = create_engine(connection)  
dbConnection = sqlEngine.connect()
```

- `connection` chứa thông tin để kết nối với máy chủ chạy MySQL. Trong trường hợp bạn không phải là người kỹ thuật thì hãy hỏi bộ phận quản lý máy chủ MySQL. Cụ thể các thông tin bao gồm:
 - ✓ `server`: là tên máy chủ hoặc địa chỉ IP chạy phần mềm MySQL. Ví dụ máy của bạn cài MySQL thì tên là localhost hoặc IP là 127.0.0.1
 - ✓ `dbname`: là tên của cơ sở dữ liệu (CSDL)
 - ✓ `username`: là tên của tài khoản có quyền truy cập vào CSDL.
 - ✓ `password`: là mật khẩu của tài khoản

Ví dụ minh họa

Giả định bạn đã có database tên là “stock” và đã thực hiện lệnh MySQL để tạo tài khoản với username là **user_stock** và password là **Stock@123**; và thiết lập quyền truy cập **SELECT** như sau:

```
CREATE USER user_stock@'localhost' IDENTIFIED WITH  
mysql_native_password BY 'Stock@123';  
GRANT SELECT ON stock.* to user_stock@localhost;
```

Bước 1: Mở kết nối

Lệnh mở kết nối tới database bằng Python như sau:

```
from sqlalchemy import create_engine  
  
sqlEngine =  
create_engine('mysql+pymysql://user_stock:Stock@123@localhost/stock')  
  
dbConnection = sqlEngine.connect()
```

Bước 2: Chuẩn bị câu lệnh truy vấn

Bạn cần phải làm quen và biết cơ bản về câu lệnh quy vấn (query) trong cơ sở dữ liệu. Cụ thể câu lệnh sau sẽ SELECT bốn cột dữ liệu symbol, time, price, volume từ bảng dailystock với điều kiện là mã cổ phiếu (symbol) bằng ‘VNM’ và kết quả trả về xếp theo thứ tự giảm dần (decrease) theo cột time.

```
query = "SELECT symbol,time,price,volume FROM dailystock where symbol  
= 'VNM' order by time desc"
```

Bước 3: Thực hiện truy vấn lấy dữ liệu và dataframe

Sử dụng thư viện pandas để thực hiện truy vấn:

```
import pandas as pd  
df = pd.read_sql(query, con=dbConnection)
```

Bước 4: Đóng kết nối

```
dbConnection.close()
```

Bước 5: Xem thông tin của dataframe

```
df.describe()
```

	price	volume
count	615432.000000	6.154320e+05
mean	138.228321	1.619740e+03

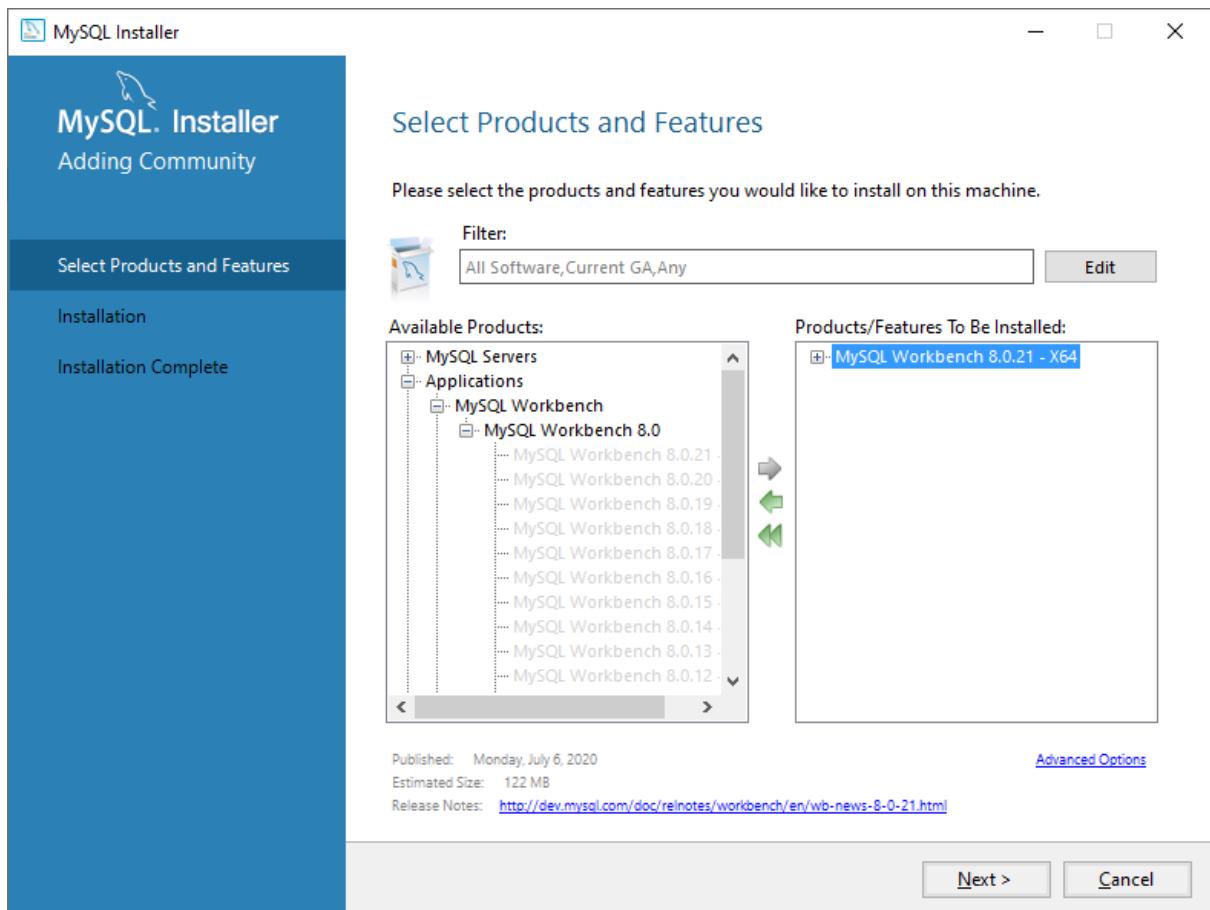
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

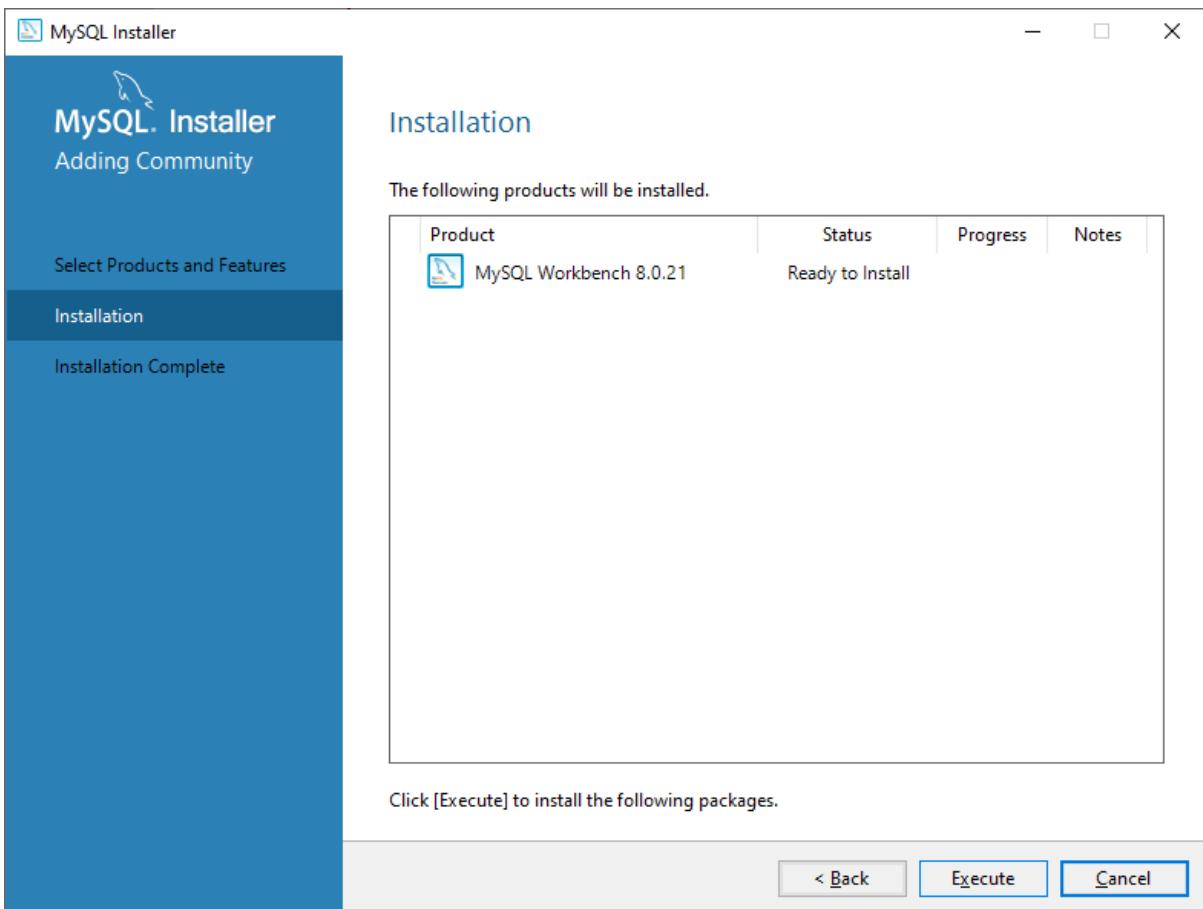
std	29.979673	2.658865e+04
min	-135.500000	0.000000e+00
25%	116.900000	8.000000e+01
50%	134.000000	3.400000e+02
75%	153.300000	1.100000e+03
max	215.000000	1.887654e+07

Sử dụng phần mềm MySQL Workbench

Tải và cài đặt phần mềm tại:

<https://dev.mysql.com/downloads/workbench/>





Sử dụng Hadoop

Trong bối cảnh dữ liệu ngày càng nhiều, đặc biệt là các hệ thống IoT (Internet of Things) thì dữ liệu phát sinh được tính bằng giây, thậm chí là mili giây. Các hệ thống lưu trữ truyền thống sẽ không còn phù hợp. Từ đó khái niệm Big Data (dữ liệu lớn) ra đời và có nhiều phần mềm giúp triển khai Big Data. Ngoài ra việc sử dụng R hoặc Python để đọc dữ liệu từ file CSV, hoặc từ các phần mềm quản lý dữ liệu chuyên dụng thì cơ chế chung là dữ liệu được nạp vào bộ nhớ máy tính (RAM – Random Access Memory). Vì vậy nếu bạn có dữ liệu lớn hơn dung lượng RAM mà máy tính của bạn đang có thì chắc là sẽ không phân tích được theo cách truyền thống.

Việc sử dụng Hadoop kết hợp với R hoặc Python có thể giải quyết được vấn đề dữ liệu lớn (hơn RAM) ở trên.

Trong phần này tôi sẽ giúp bạn làm quen với hệ thống Hadoop. Chỉ hy vọng là dừng ở mức độ làm quen và có chút trải nghiệm ban đầu. Từ đó nếu dự án lớn hơn thì có ý tưởng để nghiên cứu và triển khai cụ thể.

Bạn có thể hình dung phần mềm Hadoop sẽ giúp kết nối các máy tính thành một mạng lưới để lưu trữ dữ liệu. Nếu bạn có một cái laptop hoặc PC thì ổ cứng 1TB (Terabyte) đã là khủng. Nếu đơn vị bạn có hệ thống máy chủ thì có thể lưu trữ vài chục, vài trăm TB. Nếu cần lưu trữ lớn hơn thì sao? Một hệ thống đơn lẻ rất khó đáp ứng được. Câu hỏi đặt ra là liệu có thể kết nối các máy tính đơn lẻ trong tổ chức của bạn để trở thành một hệ thống khủng mà việc truy cập vào nó như là truy cập vào một máy tính

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

có được không? Câu trả lời là được. Phần mềm Hadoop sẽ giúp chúng ta làm được việc này. Mạng lưới các máy tính trong hệ thống Hadoop gọi là Hadoop Cluster. Phần mềm R sẽ hỗ trợ phân tích dữ liệu lớn được lưu trữ trên Hadoop.

Cài đặt Hadoop

Để làm quen với Hadoop nếu bạn rành Công nghệ thông tin thì có thể dùng hệ điều hành Linux (phổ biến là Ubuntu Linux và CentOS linux). Nếu bạn dùng Windows thì có thể dùng VMware để tạo máy ảo và cài Linux. Sau đó sẽ cài Hadoop vào Linux. Phần tiếp theo tôi sẽ giúp bạn hình dung và có thể tự cài mọi thứ lên máy tính của mình gồm:

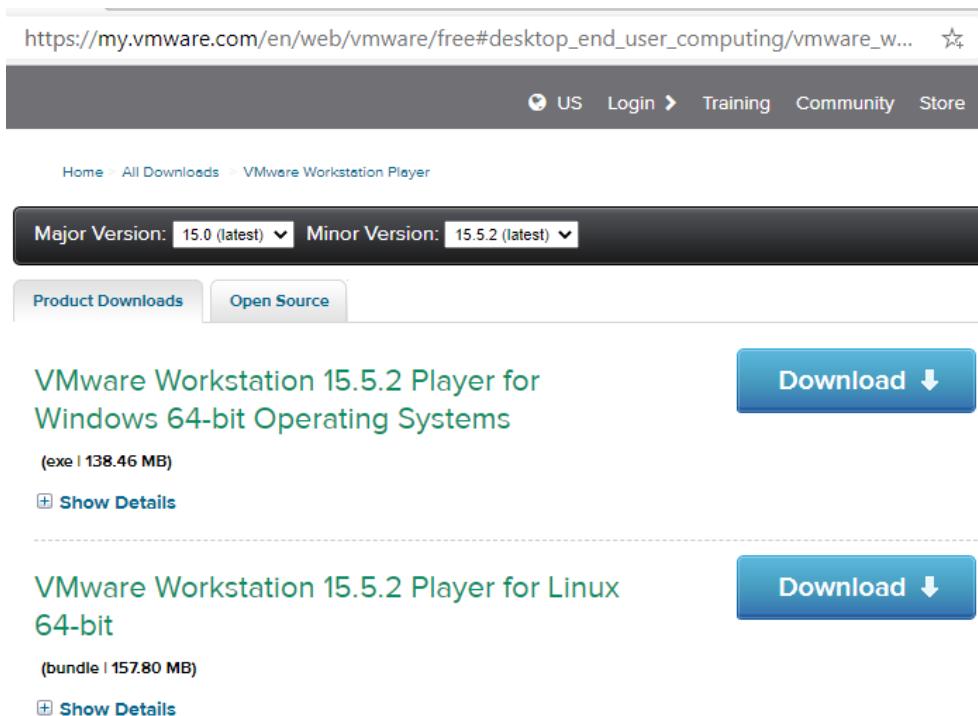
- ① Tải và file .ISO của Ubuntu Desktop.
- ② Tải và cài đặt VMware, phiên bản hiện tại là 15.5.2
- ③ Cài đặt Ubuntu Desktop 20.0 lên VMware

Tải Ubuntu

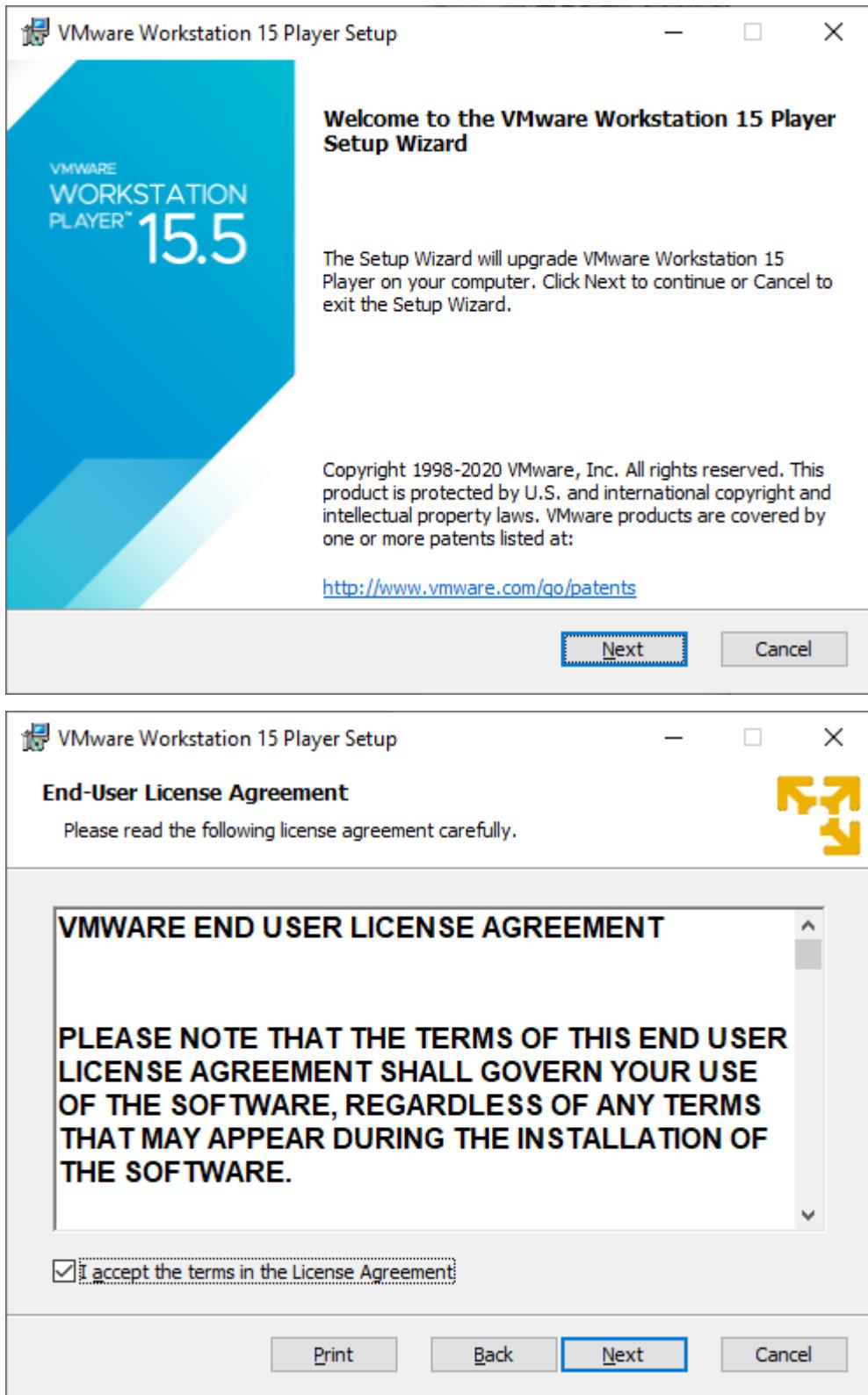
Vào trang web <https://ubuntu.com/download/desktop> để tải Ubuntu Desktop với phiên bản mới nhất – hiện tại là Ubuntu 20.04.2. Để làm quen thôi thì bạn dùng phiên bản Desktop là được.

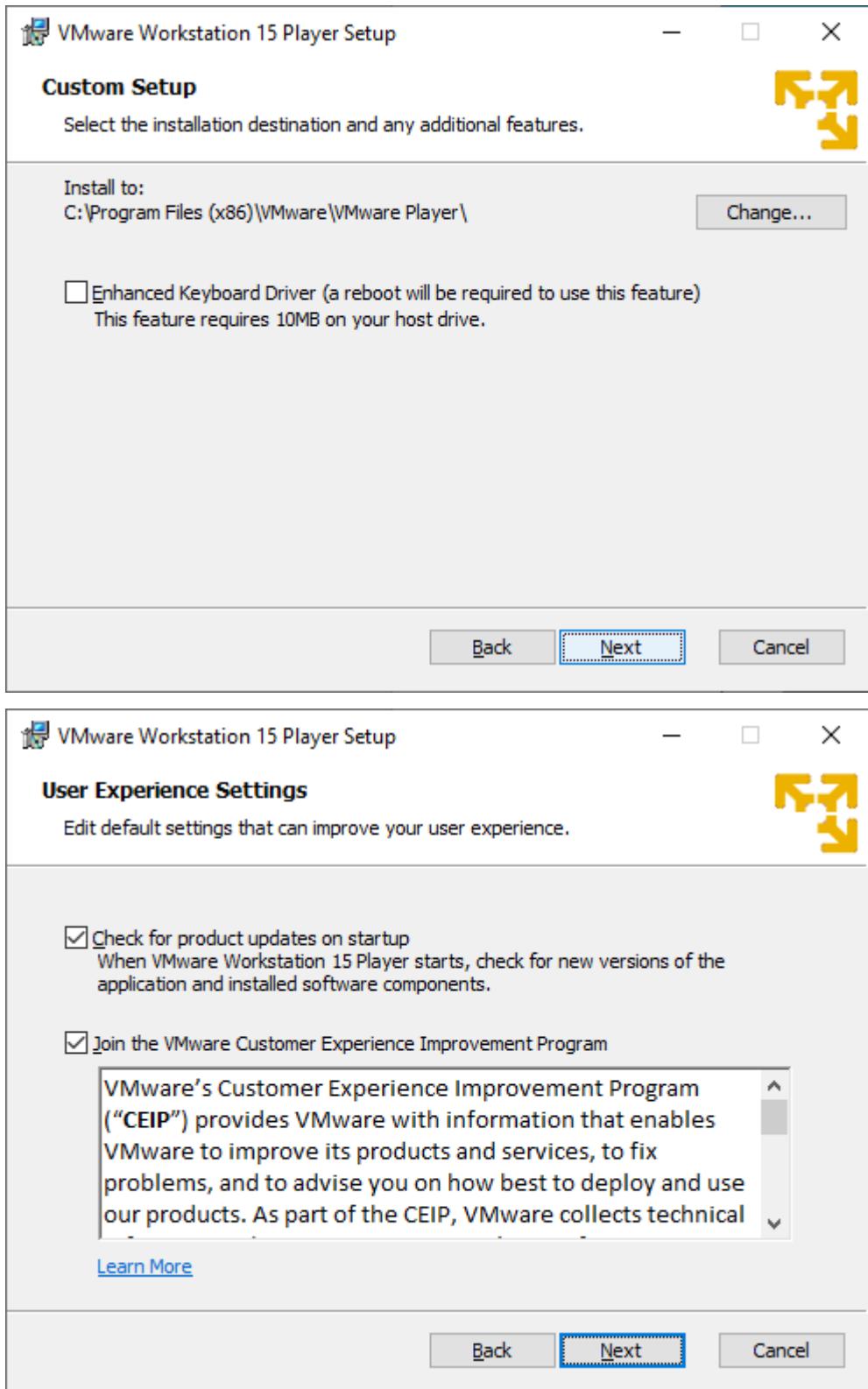
Sử dụng VMware

Vào trang web <https://www.vmware.com/go/downloadworkstationplayer> để tải và cài đặt VMware Workstation Player (gọi tắt là VMware).

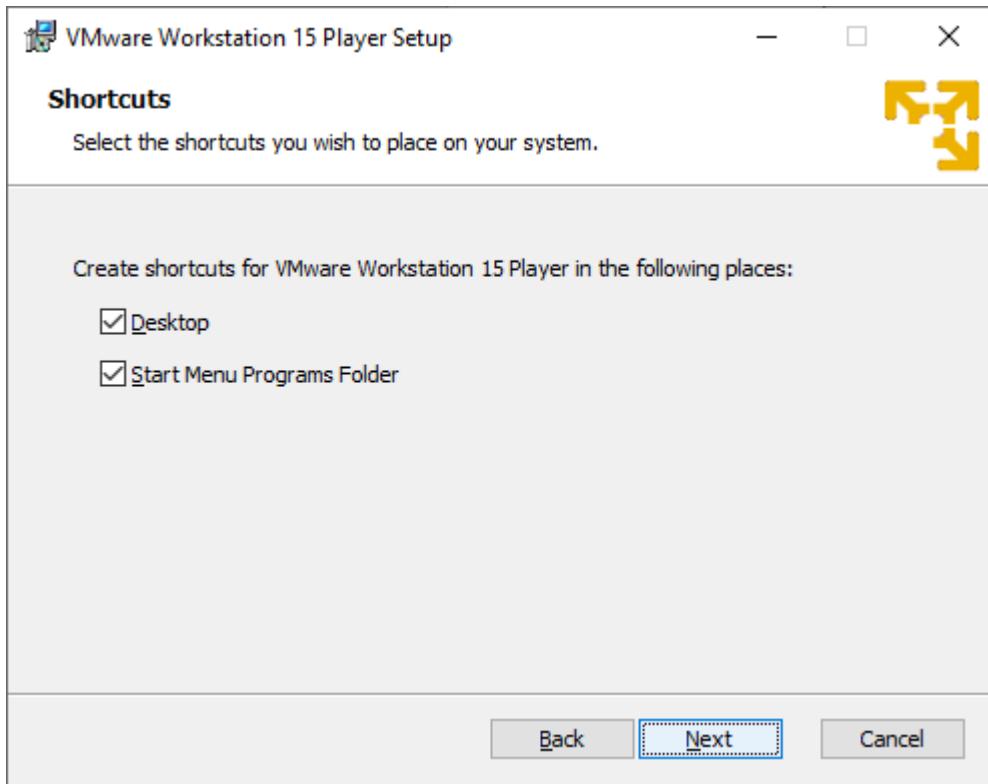


Quá trình cài đặt tương đối dễ dàng, bạn cứ làm theo hướng dẫn, cơ bản là chọn Yes và Next:

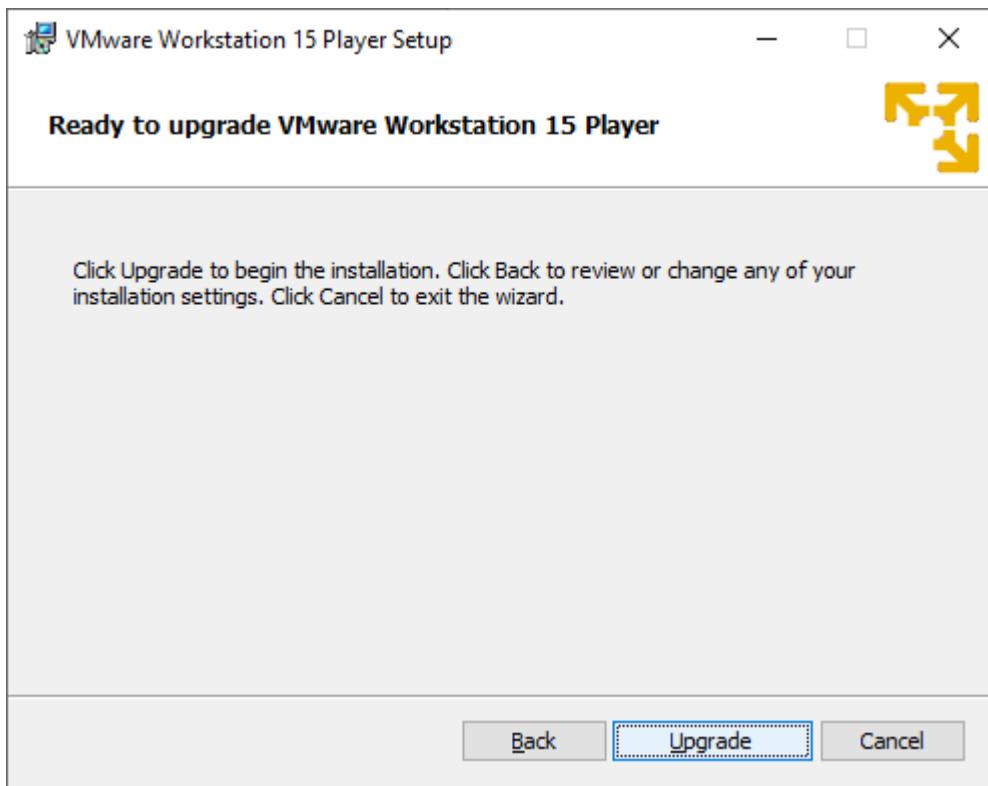


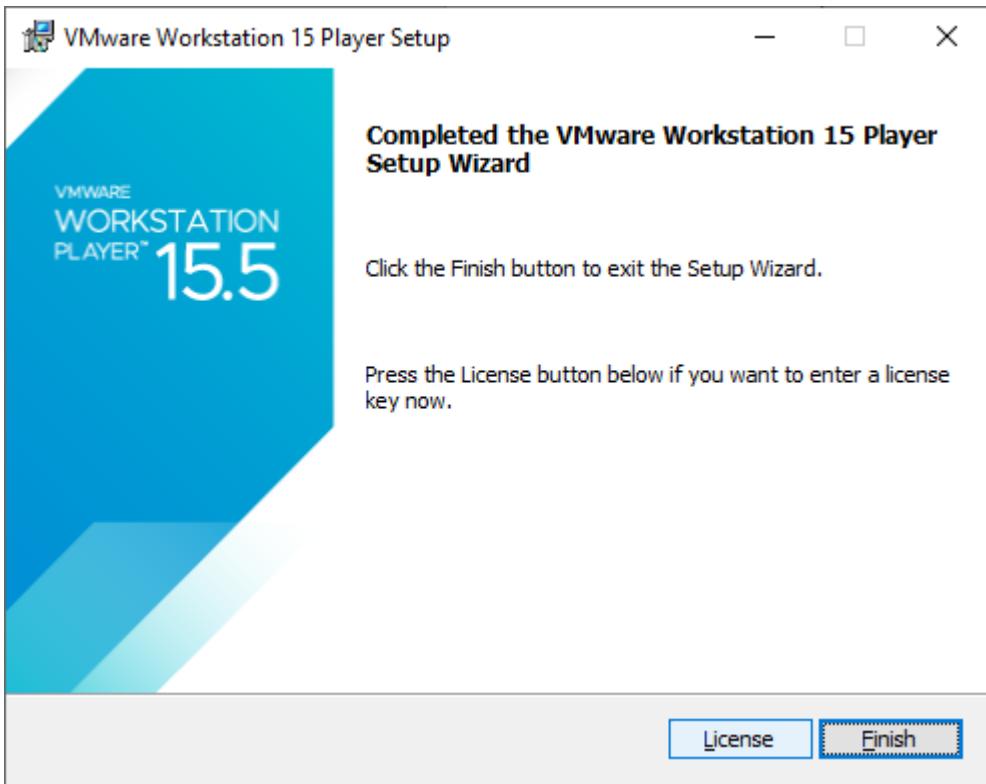


Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



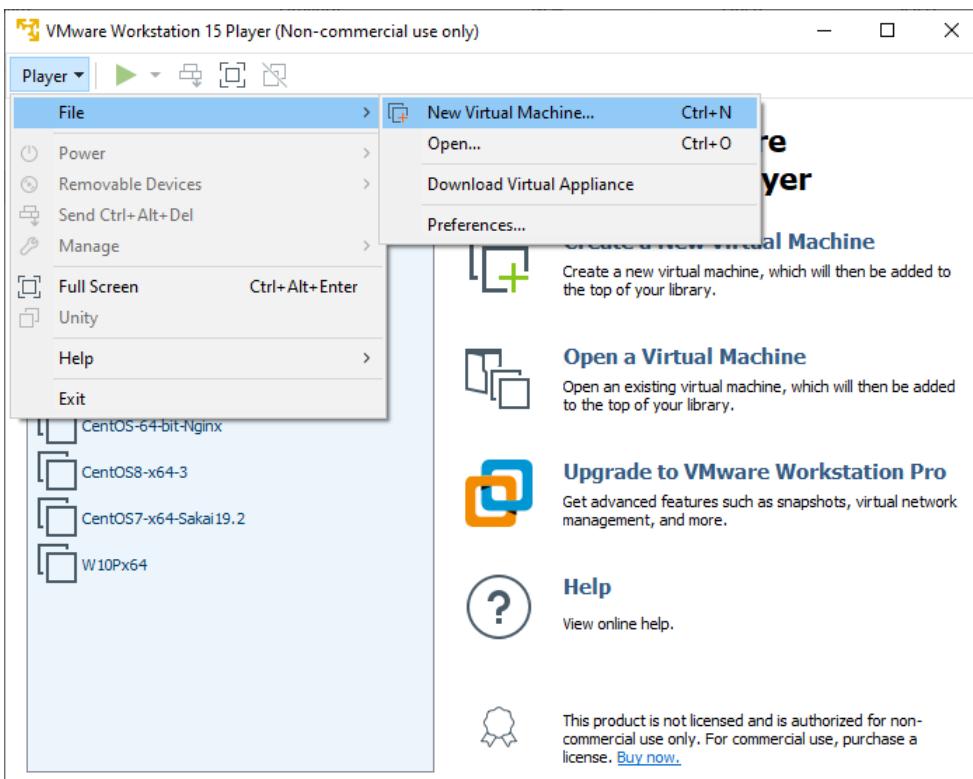
Nếu máy tính của bạn đã có VMware thì có thể nâng cấp bằng cách bấm nút Upgrade.

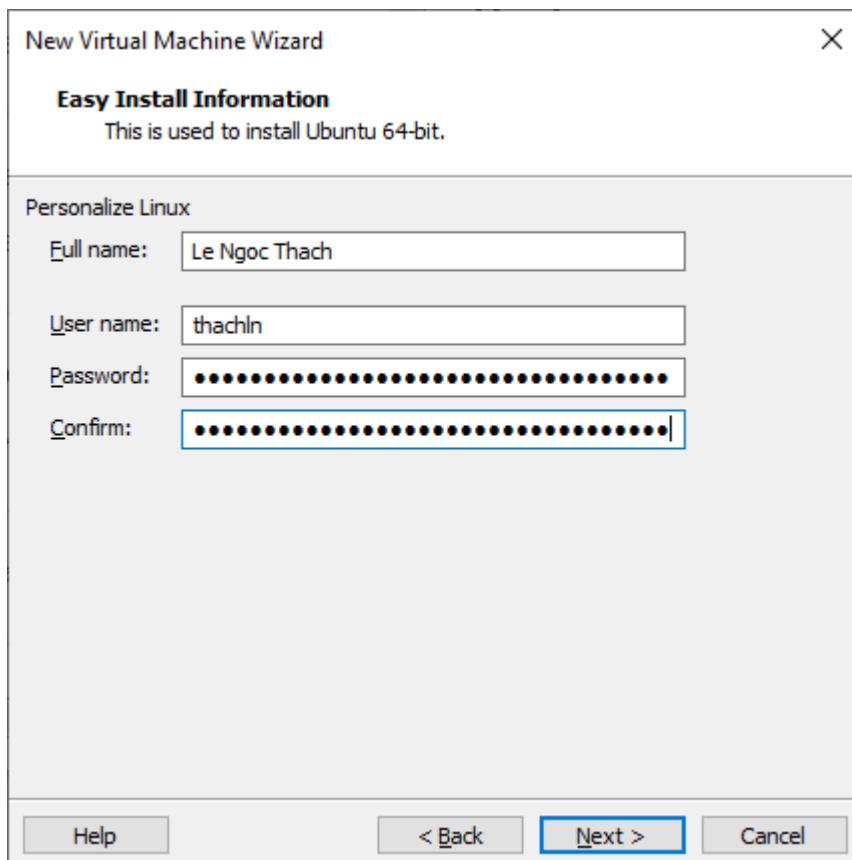
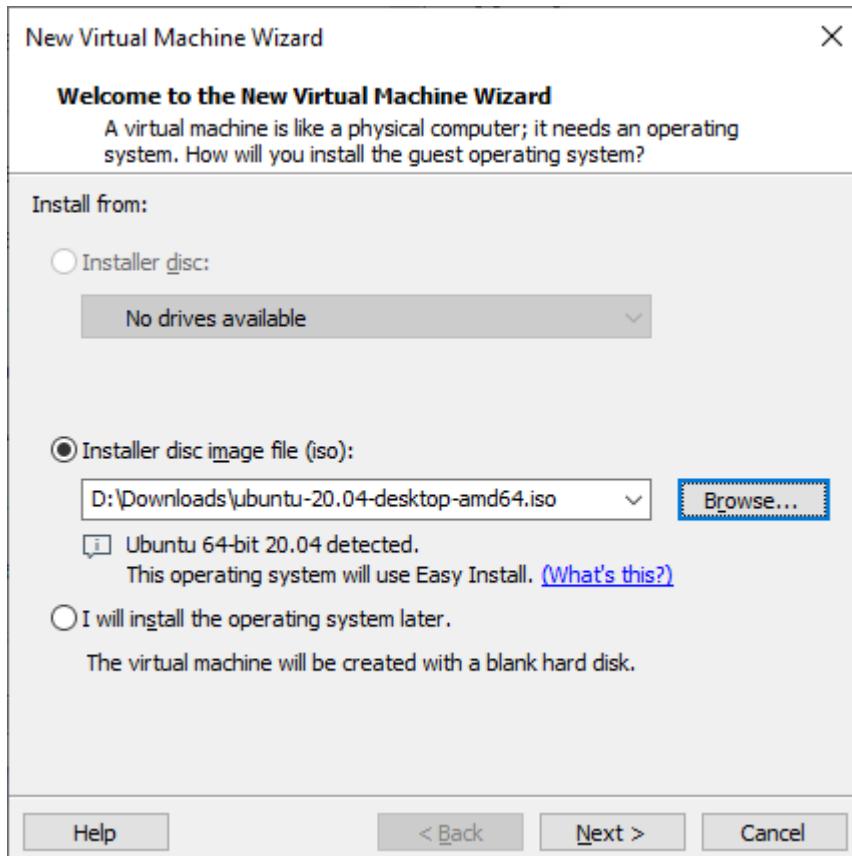


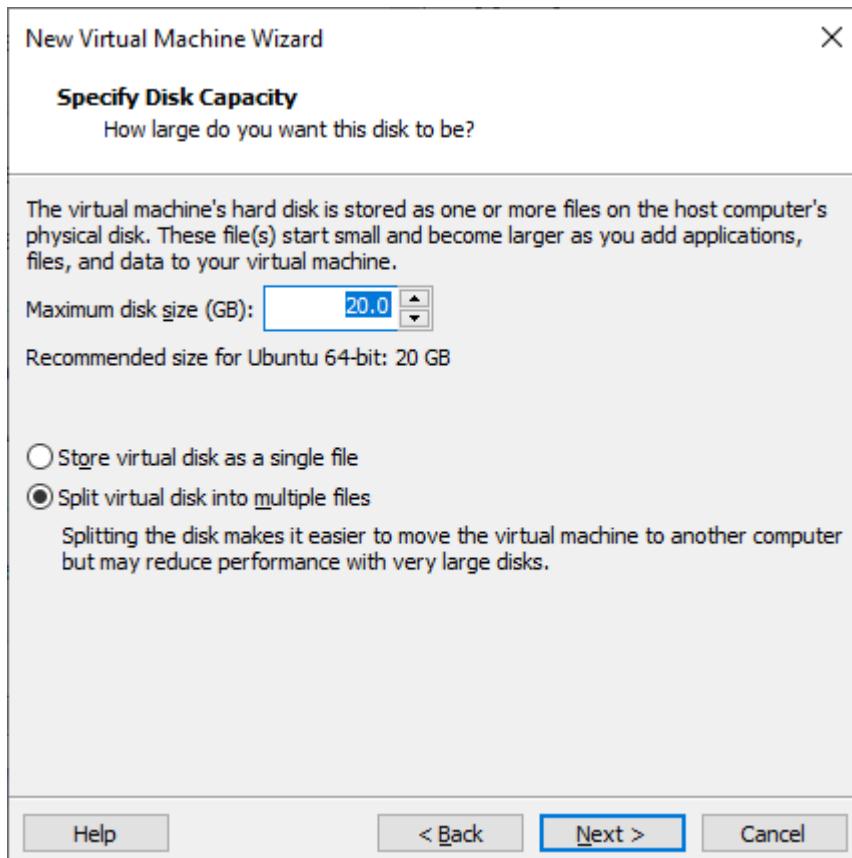
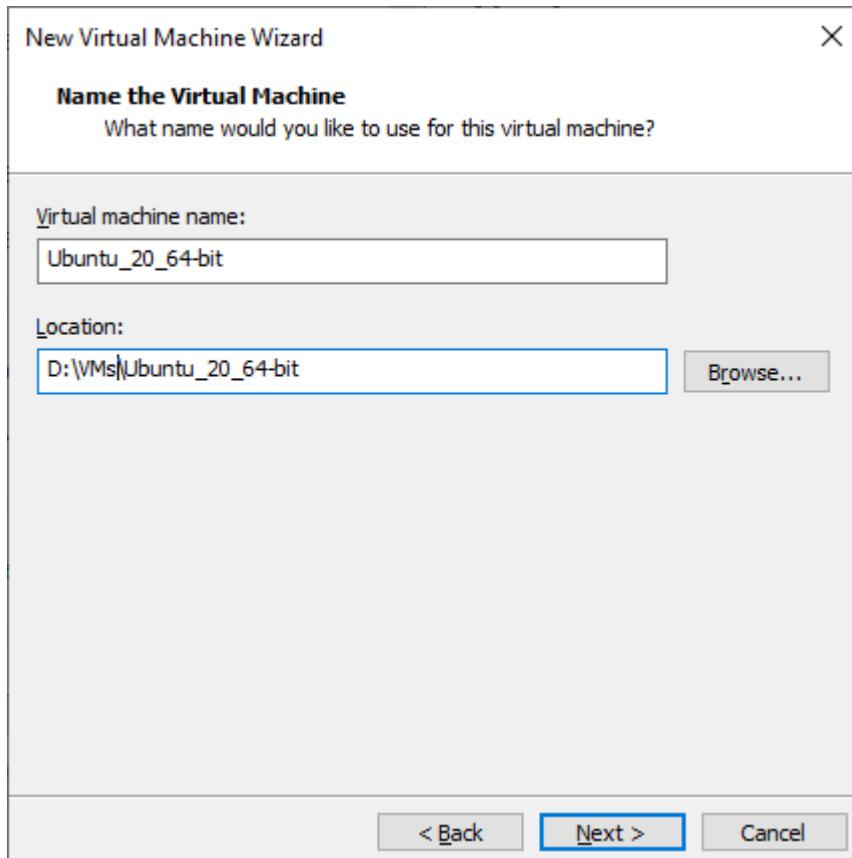


Tạo máy ảo Ubuntu

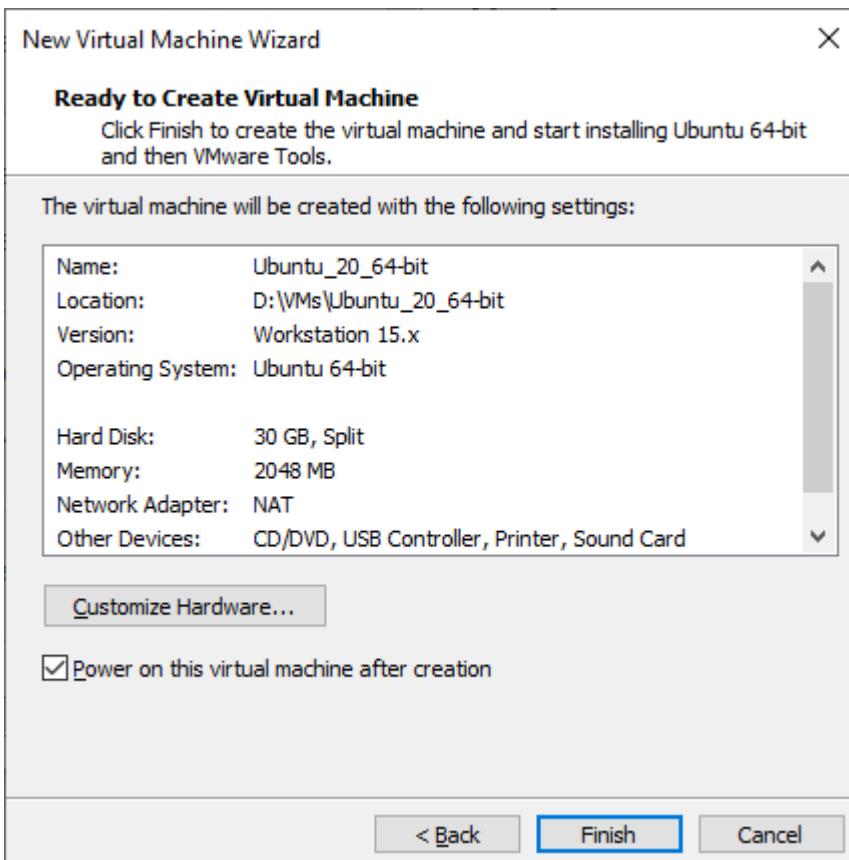
Sau đó khởi động VMware và khởi tạo máy ảo theo hướng dẫn sau đây





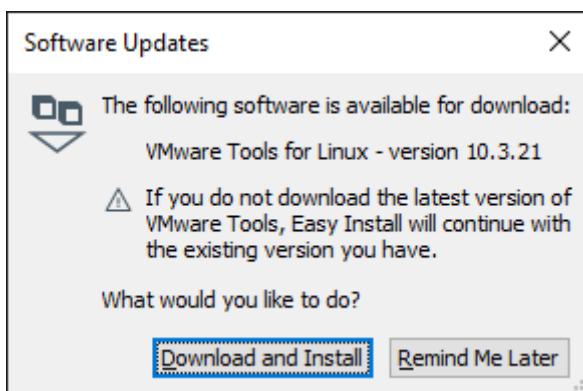


Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



Bấm nút Finish để hoàn thành việc tạo máy tính ảo bên trong phần mềm VMWare.

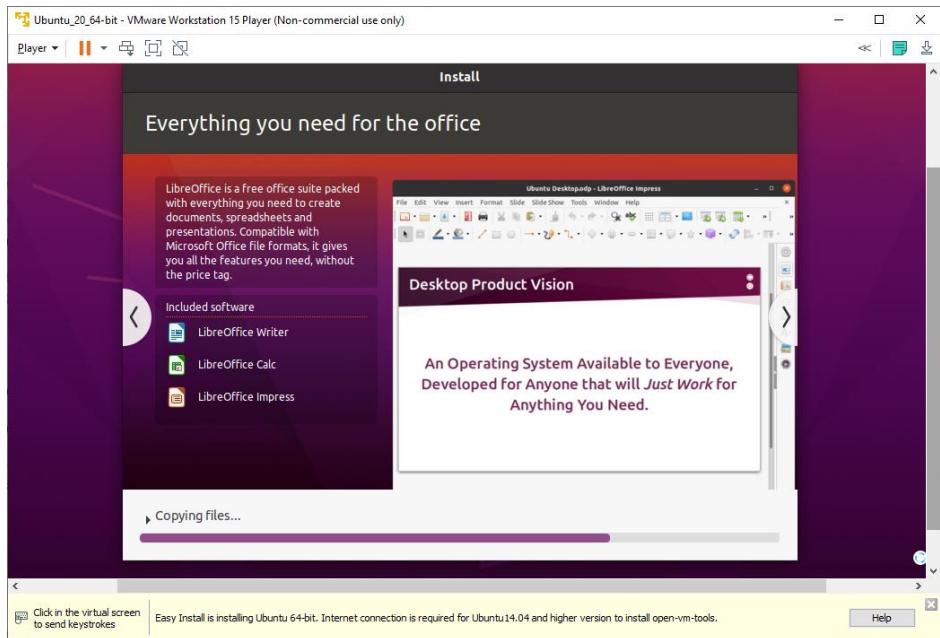
VMware sẽ tự động bắt đầu quá trình cài đặt Ubuntu cho bạn. Trong lúc cài đặt thì có thể hiển thị hộp thoại bên dưới. Bạn cứ chọn Download and Install.



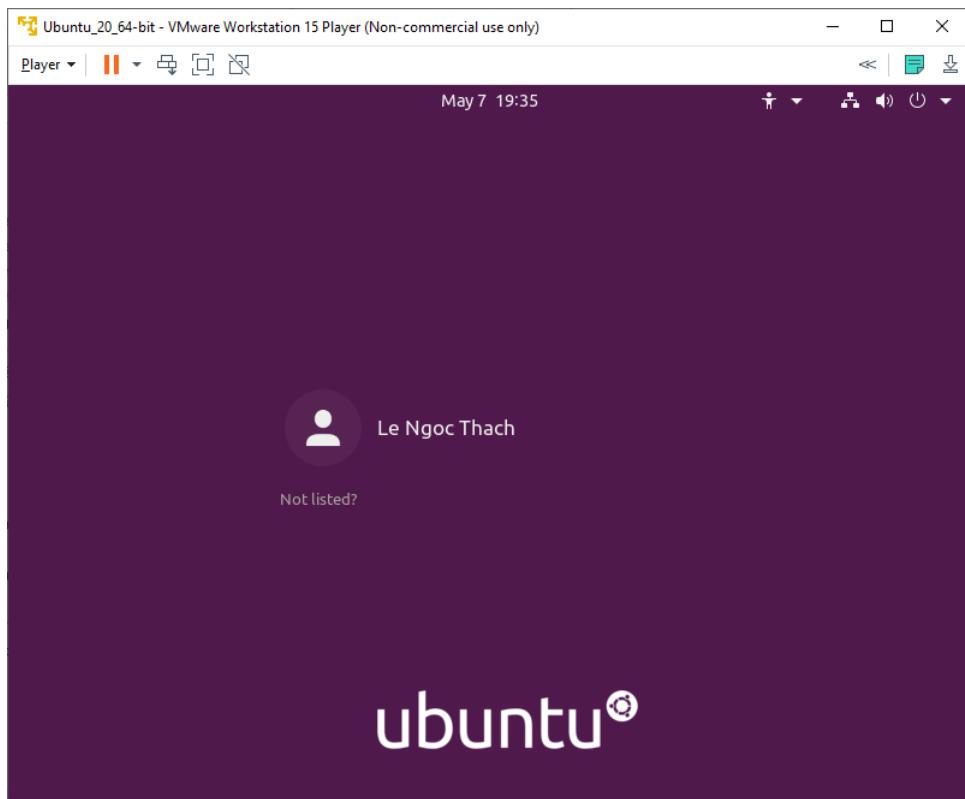
Như vậy lúc này trên máy tính Windows 10 của bạn có phần mềm VMware. Bên trong phần mềm VMware lại có một cái máy tính Ubuntu 20.0. Thật là tuyệt vời phải không?

Phải nói lời cảm ơn đến hãng VMware đã có một phần mềm tuyệt vời để giúp chúng ta trải nghiệm nhiều cái máy tính ảo bên trong chỉ một cái máy tính thật.

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

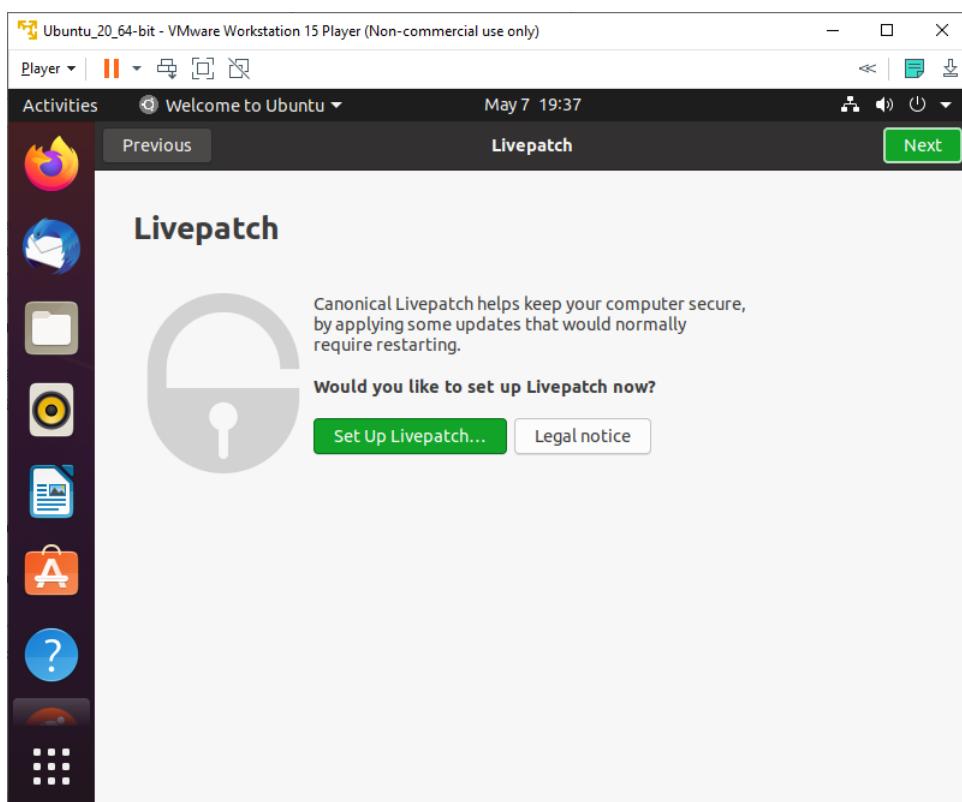
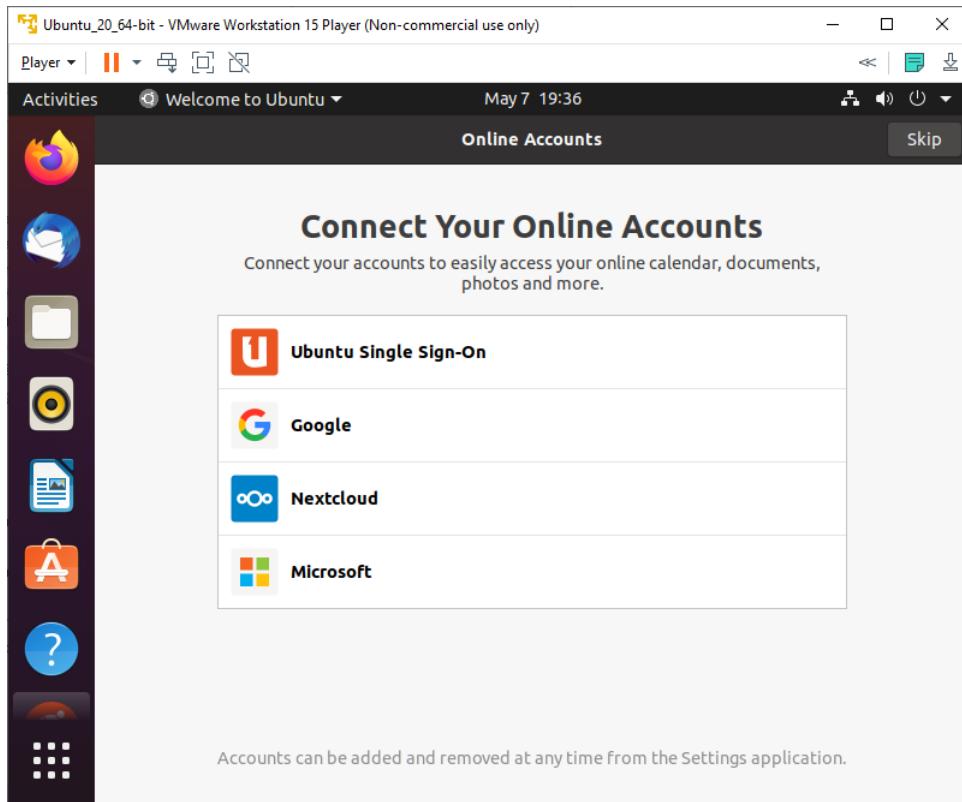


Sau khi cài đặt xong thì bấm vào tên mà bạn đã khai báo lúc nãy. Sau đó gõ password để đăng nhập

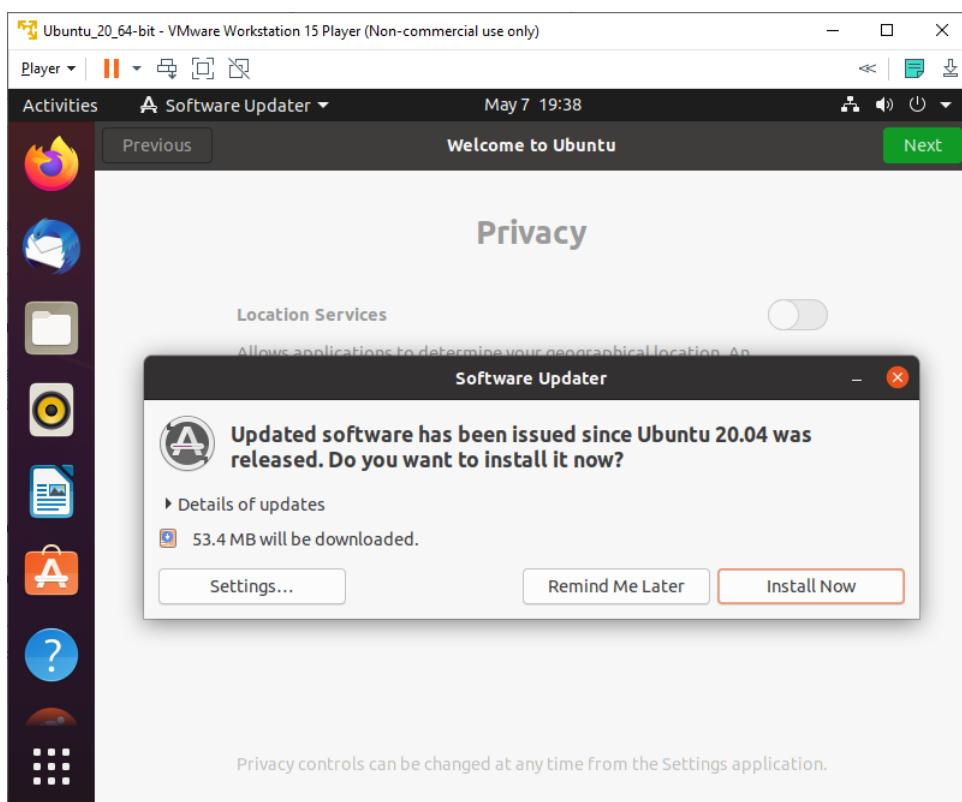
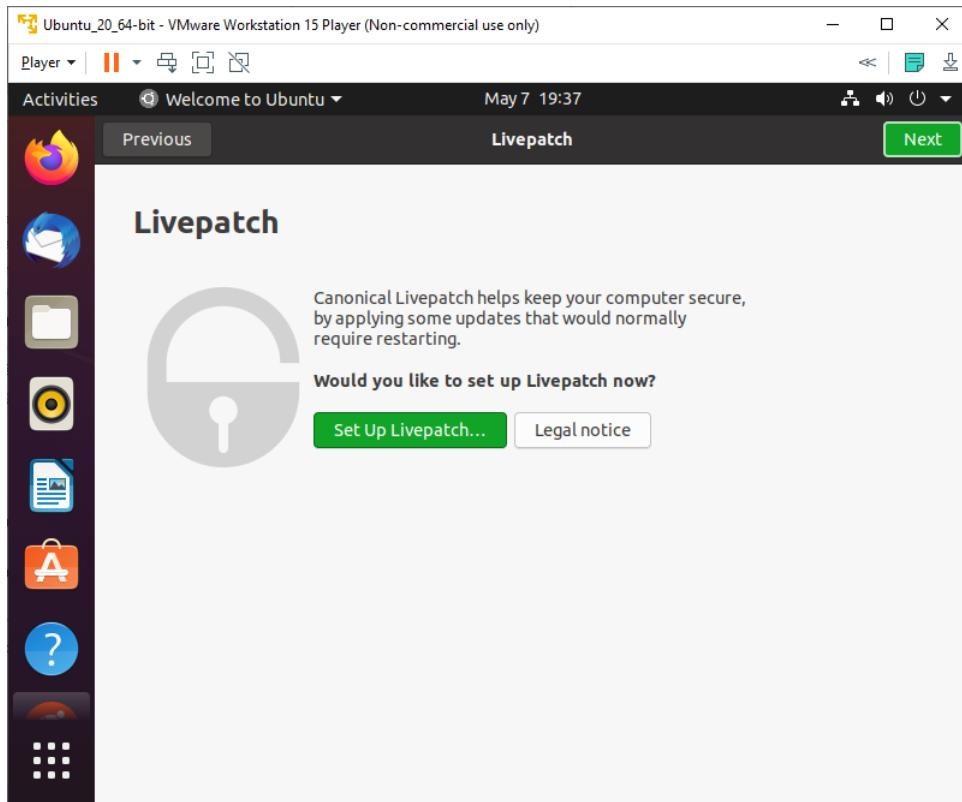


Sau khi đăng nhập, bạn theo hướng dẫn bấm nút ở góc phải trên Skip, Next, và Done.

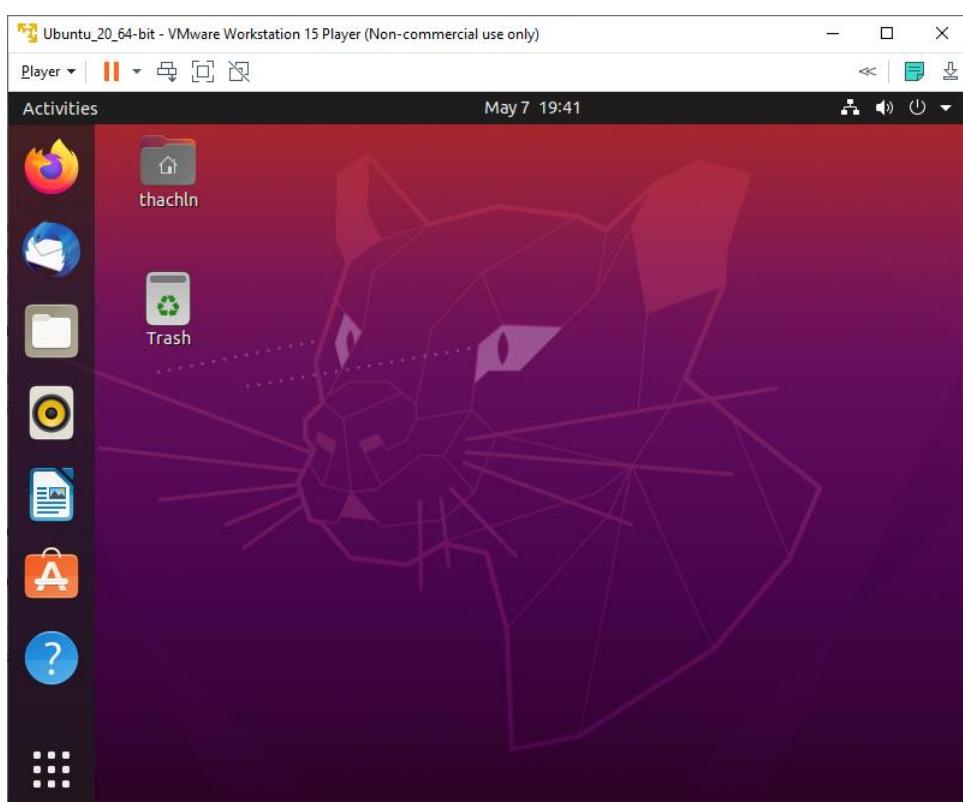
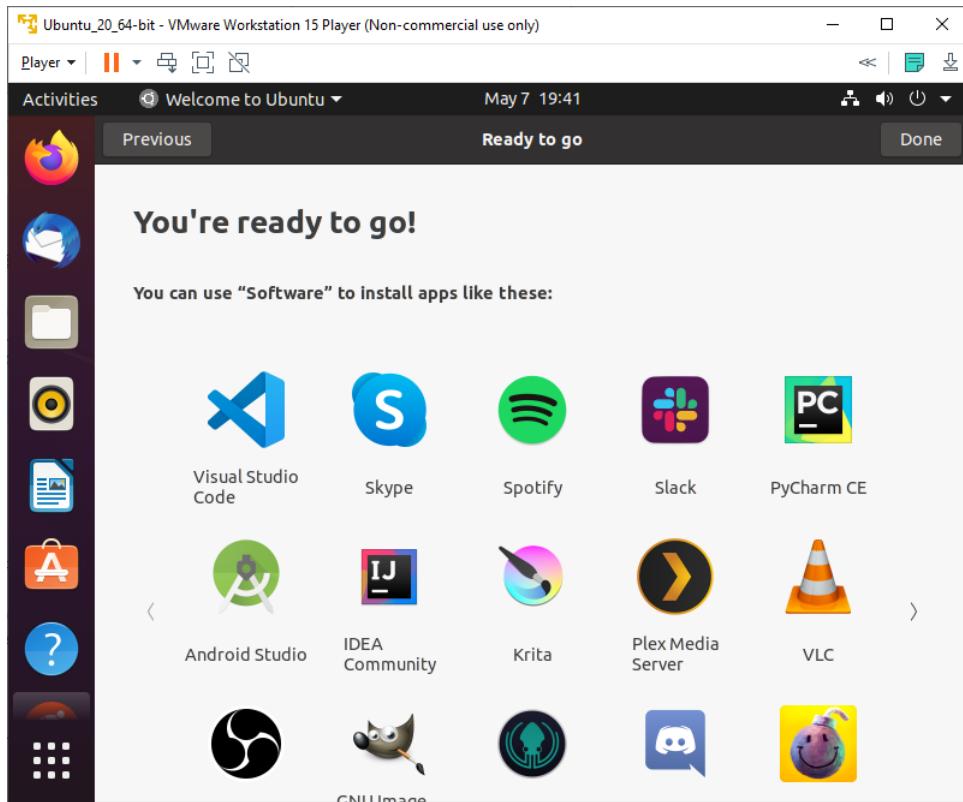
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python



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



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



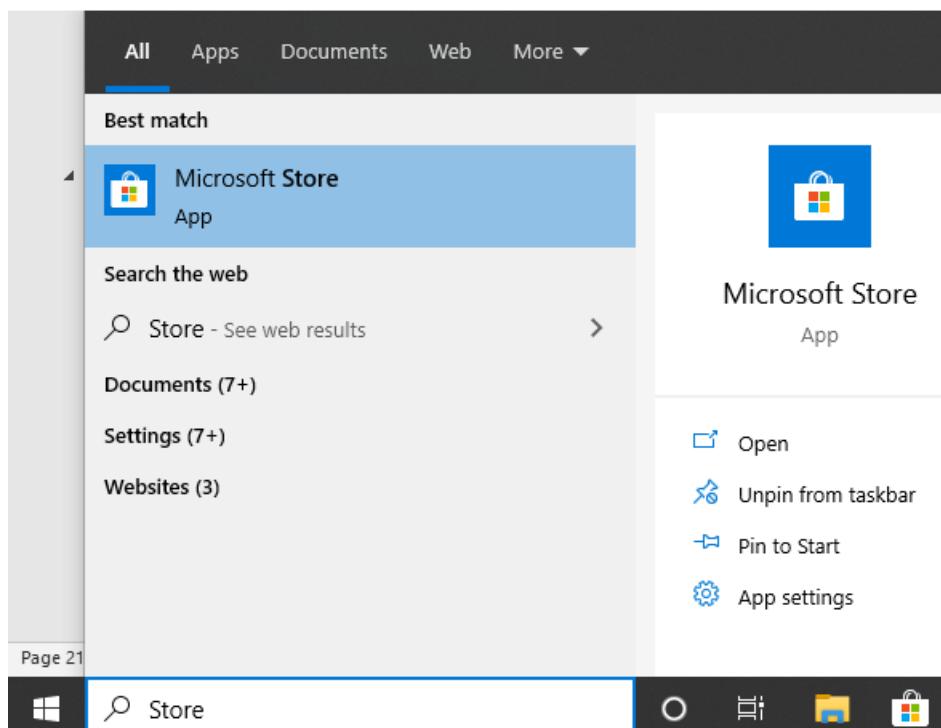
Như vậy bạn đã có máy ảo Ubuntu 20.04,

Để phóng to máy ảo đầy màn hình để làm việc thì bấm vào biểu tượng trong thanh công cụ của VMware.

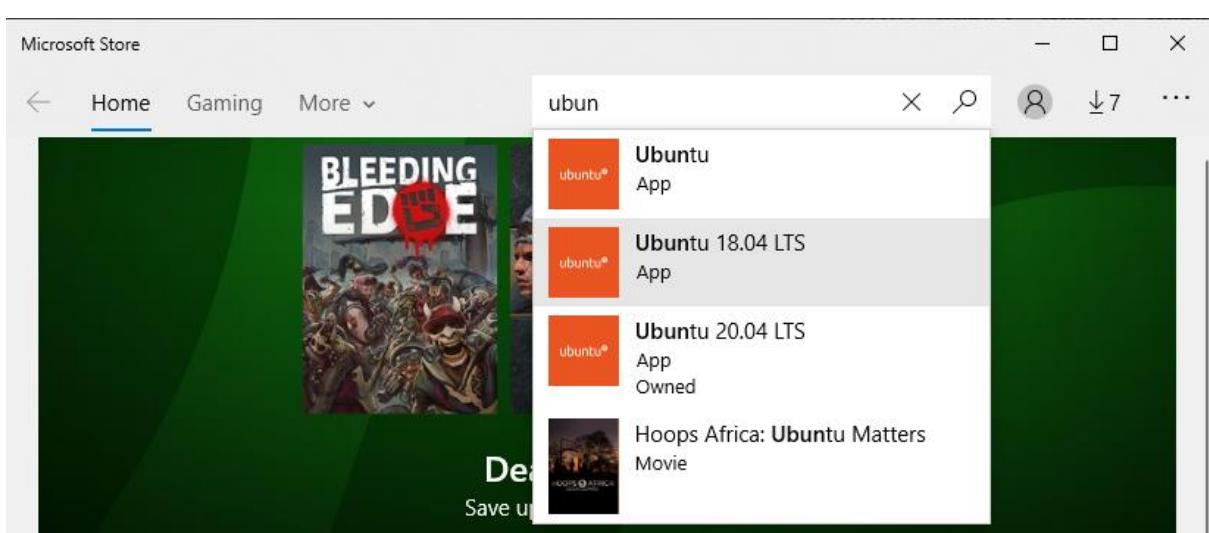
Cài đặt Ubuntu nhúng trong Windows

Một cách khác để dùng Ubuntu trong Windows 10 mà không phải cài máy ảo VMWare là chức năng Windows Subsystem for Linux, gọi tắt là WSL. WSL được Microsoft tích hợp hỗ trợ chạy Ubuntu trong Windows như là một phần mềm.

Cài đặt WSL bằng cách mở Microsoft Store bằng cách nhấn phím Ctrl + Esc, hoặc bấm phím Windows hoặc bấm chuột vào nút Start. Sau đó gõ chữ Store ra menu bên dưới:



Bấm vào biểu tượng hoặc mục Microsoft Store. Sau đó gõ Ubuntu trên ô Search. Tôi cài Ubuntu 18.04 LTS để thực hiện một số phần ví dụ trong eBook này.

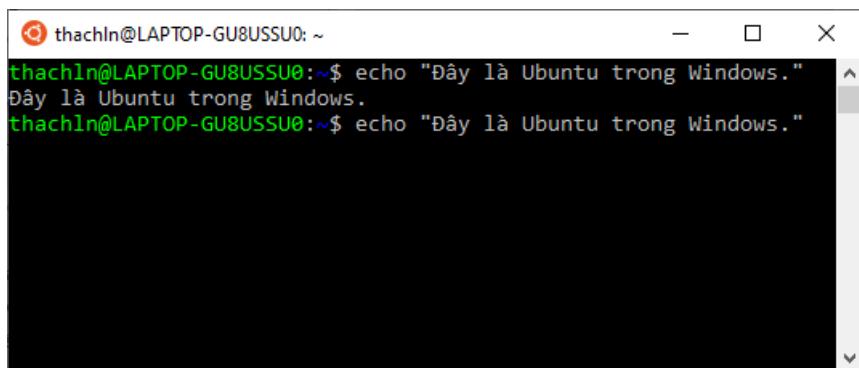


Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

Chú ý hiện tại đã có phiên bản Ubuntu mới 20.04 nhưng khi thực hành theo tài liệu trong eBook này sẽ gặp một số trục trặc vì thế nếu các bạn dùng thì hãy tìm cách xử lý nếu gặp lỗi nhé! Hãy xem qua trang web sau nếu muốn cài đặt Ubuntu 20.04:

<https://docs.microsoft.com/en-us/windows/wsl/install-win10>

Sau khi cài xong khởi động Ubuntu như là một phần mềm bình thường. Cửa sổ hiện lên có chức năng giống như là Terminal (cửa sổ gõ lệnh) của Linux.



Từ trong cửa sổ này, bạn có thể truy cập ra các ổ đĩa của Windows bằng cách vào thư mục /mnt. Ví dụ thử lệnh:

```
cd /mnt/d  
ls
```

Cập nhật thư viện:

```
sudo apt-get update
```

Mặc định python3 được cài, hãy kiểm tra bằng lệnh:

```
python3 -v
```

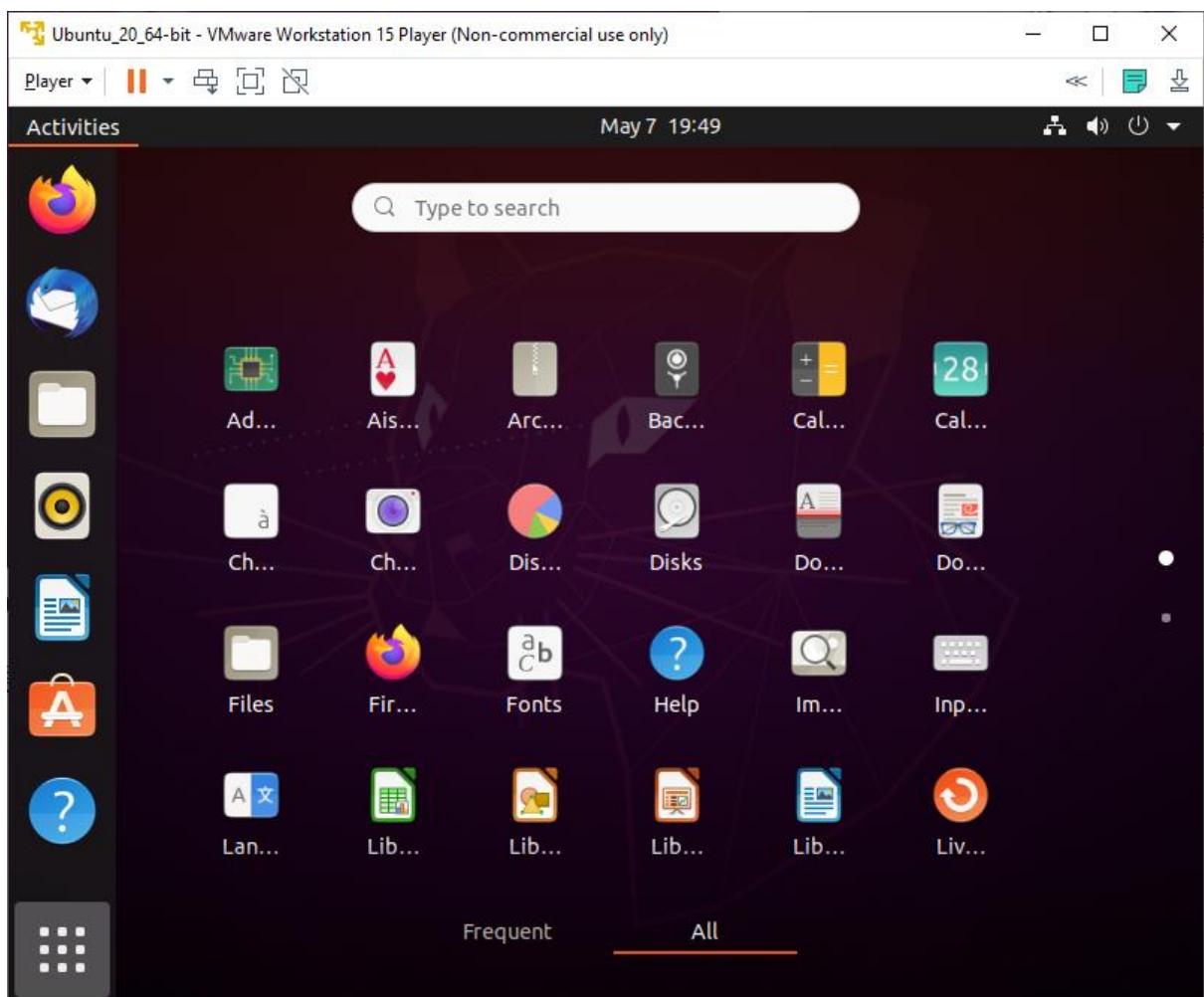
Hy vọng bạn có trải nghiệm sử dụng Ubuntu ngay trên Windows 10 để hỗ trợ học hành, cho công việc tốt!

Cám ơn Microsoft! Cám ơn Ubuntu!

Bài 19: Sử dụng Ubuntu

Mở cửa sổ lệnh

Bấm vào biểu tượng ở góc trái bên dưới màn hình. Biểu tượng này có tên là Show Applications. Sau đó gõ chữ terminal vào ô tìm kiếm. Kết quả sẽ ra công cụ Terminal. Bấm chuột vào Terminal để mở cửa sổ lệnh. Từ đây gọi tắt là mở Terminal.



Dán nội dung vào máy ảo

Sử dụng phím tắt Ctrl + Shift + V.

Để các bạn trải nghiệm nhanh thì có thể copy & paste các lệnh vào trong Terminal của Ubuntu rồi sửa lại nếu cần.

Thực hiện lệnh với quyền root

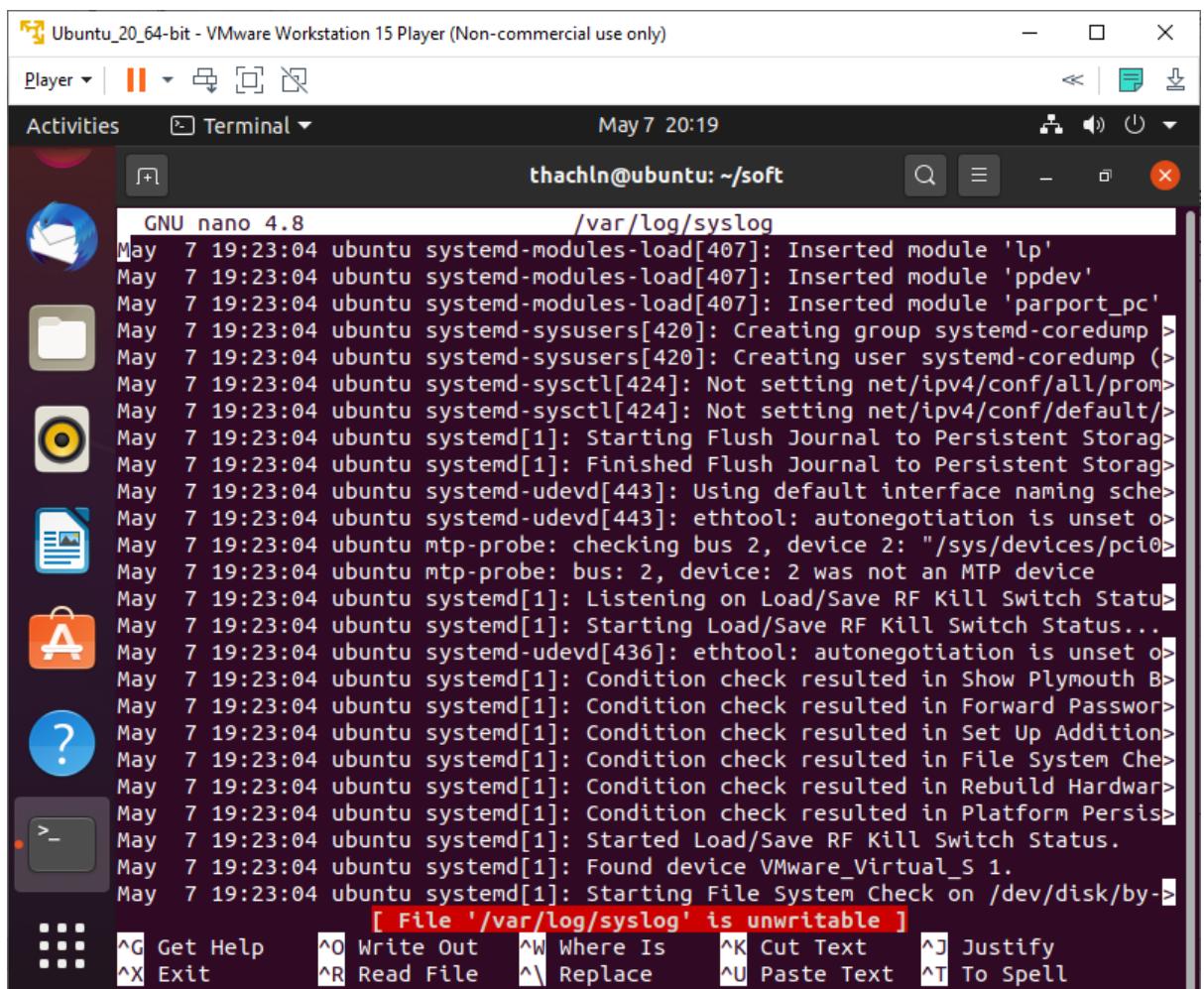
Thông thường bạn đăng nhập vào Ubuntu với một tài khoản được cấp. Tài khoản này đôi khi bị hạn chế một số quyền trên máy Ubuntu. Trong quá trình cài đặt và thực hiện lệnh nói chung khi nào cần thực thi với quyền cao hơn thì thêm chữ sudo ở đầu lệnh.

Soạn thảo tài liệu bằng lệnh nano

Trong Ubuntu hay Linux nói chung, lệnh nano rất hữu dụng cho các bạn chỉnh sửa các file cấu hình.

Ví dụ lệnh sau sẽ mở file system cho bạn chỉnh sửa:

```
nano /var/log/syslog
```



```
GNU nano 4.8 /var/log/syslog
May  7 19:23:04 ubuntu systemd-modules-load[407]: Inserted module 'lp'
May  7 19:23:04 ubuntu systemd-modules-load[407]: Inserted module 'ppdev'
May  7 19:23:04 ubuntu systemd-modules-load[407]: Inserted module 'parport_pc'
May  7 19:23:04 ubuntu systemd-sysusers[420]: Creating group systemd-coredump >
May  7 19:23:04 ubuntu systemd-sysusers[420]: Creating user systemd-coredump (>
May  7 19:23:04 ubuntu systemd-sysctl[424]: Not setting net/ipv4/conf/all/prom>
May  7 19:23:04 ubuntu systemd-sysctl[424]: Not setting net/ipv4/conf/default/>
May  7 19:23:04 ubuntu systemd[1]: Starting Flush Journal to Persistent Storage
May  7 19:23:04 ubuntu systemd[1]: Finished Flush Journal to Persistent Storage
May  7 19:23:04 ubuntu systemd-udevd[443]: Using default interface naming sche...
May  7 19:23:04 ubuntu systemd-udevd[443]: ethtool: autonegotiation is unset o...
May  7 19:23:04 ubuntu mtp-probe: checking bus 2, device 2: "/sys/devices/pci0...
May  7 19:23:04 ubuntu mtp-probe: bus: 2, device: 2 was not an MTP device
May  7 19:23:04 ubuntu systemd[1]: Listening on Load/Save RF Kill Switch Status...
May  7 19:23:04 ubuntu systemd[1]: Starting Load/Save RF Kill Switch Status...
May  7 19:23:04 ubuntu systemd-udevd[436]: ethtool: autonegotiation is unset o...
May  7 19:23:04 ubuntu systemd[1]: Condition check resulted in Show Plymouth Br...
May  7 19:23:04 ubuntu systemd[1]: Condition check resulted in Forward Passwor...
May  7 19:23:04 ubuntu systemd[1]: Condition check resulted in Set Up Additional...
May  7 19:23:04 ubuntu systemd[1]: Condition check resulted in File System Che...
May  7 19:23:04 ubuntu systemd[1]: Condition check resulted in Rebuild Hardware...
May  7 19:23:04 ubuntu systemd[1]: Condition check resulted in Platform Persis...
May  7 19:23:04 ubuntu systemd[1]: Started Load/Save RF Kill Switch Status.
May  7 19:23:04 ubuntu systemd[1]: Found device VMware_Virtual_S 1.
May  7 19:23:04 ubuntu systemd[1]: Starting File System Check on /dev/disk/by-...
[ File '/var/log/syslog' is unwritable ]
```

[File '/var/log/syslog' is unwritable]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell

Dòng màu đỏ cảnh báo “is writable” có nghĩa là bạn không có quyền chỉnh sửa file này. Tình huống này liên quan đến lệnh sudo đã đề cập ở phần trước. Nếu thật sự muốn chỉnh sửa file mà user bạn đang sử dụng không có quyền chỉnh sửa thì gõ:

```
sudo nano <đường dẫn file>
```

Các phím tắt thường dùng:

Ctrl + X: thoát

Ctrl + O: để lưu file

Ctrl + K: để xóa dòng

Ctrl + W: để tìm kiếm.

Ctrl + W, rồi nhấn tiếp Ctrl + R: để tìm và thay thế

Ctrl + C: hủy thao tác đang định làm

Ctrl + W, Ctrl + T: để nhảy tới một dòng cụ thể.

Alt + /: nhảy tới dòng cuối cùng.

Xem địa chỉ IP của máy

```
ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 00:0c:29:1e:86:a0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.146.128/24 brd 192.168.146.255 scope global dynamic noprefixro
ute ens33
        valid_lft 1620sec preferred_lft 1620sec
    inet6 fe80::efb2:c8e2:ba48:58be/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Bạn chỉ cần chú ý dòng có chữ inet, tiếp theo là địa chỉ IP 192.168.146.128, đây là IP trên máy ảo Ubuntu của tôi. Bạn hình dung IP giống như địa chỉ nhà của bạn. Tức là trong thế giới mạng máy tính thì địa chỉ IP là chuỗi các chữ số để xác định chính xác máy của bạn trong mạng mà bạn đang đứng. Tùy vào bạn đang đứng trong mạng nào thì sẽ có IP riêng. Như vậy một máy tính có thể có nhiều địa chỉ IP. Tạm thời trên máy ảo Ubuntu bạn cần biết địa chỉ IP của nó để tiện truy cập trong các phần sau.

Chú ý các lệnh ở phần sau liên quan đến địa chỉ IP 192.168.146.128 thì bạn phải hiểu là phải sửa lại theo số IP trên máy ảo của bạn nhé!

Gõ lệnh nhanh với phím Tab

Trong cửa sổ Terminal, khi gõ đường dẫn của file hoặc thư mục thì bạn nên sử dụng phím tab để Ubuntu hiển thị kí tự tiếp theo. Nếu chỉ có một tinh huống tiếp theo thì Ubuntu sẽ hiển thị luôn thư mục hoặc file cho các bạn. Ngược lại khi có nhiều tinh huống (tên file, thư mục giống nhau) thì bạn gõ tiếp phím tab để thấy các thư mục và file có thể.

Ví dụ: Bạn thử gõ

```
ls /o
```

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

Rồi gõ phím tab xem. Ubuntu sẽ hiện ra cho bạn lệnh sau để bạn gõ tiếp cho nhanh vì trong thư mục / (đọc là thư mục root) chỉ có một thư mục con “opt”.

```
ls /opt/
```

Bạn thử gõ tiếp tab 1 lần, 2 lần xem sao. Ubuntu sẽ hiển thị các thư mục con bên trong /opt/ cho bạn xem

```
hadoop/      hadoop-2.10.0/ jdk/
```

Tương tự gõ tiếp chữ h rồi tab

```
ls /opt/h
```

Gõ tab, kết quả

```
ls /opt/hadoop
```

Chuyển đổi giữa Ubuntu và Windows

Khi bạn làm việc trong Ubuntu – máy ảo chạy trong phần mềm VMware thì mọi thao tác chuột và gõ phím thì sẽ có tác dụng trong Ubuntu. Nếu bạn muốn rời Ubuntu để trở về làm việc với Windows thì dùng phím Ctrl + Alt.

Khởi động lại Ubuntu và Linux nói chung

```
sudo telinit 6
```

Cài đặt Java cho Ubuntu

Mở cửa sổ lệnh copy & paste dòng lệnh sau. Nhấn Enter để cài đặt Java 8.

```
sudo apt-get install openjdk-8-jdk
```

Sau đó làm theo hướng dẫn bằng cách nhấn Y khi được hỏi Y/n.

Khi máy chính chạy Windows có kết nối Internet thì mặc định máy ảo Ubuntu cũng được kết nối Internet. Lệnh trên Ubuntu sẽ tải Java 8 từ Internet về và cài vào máy.

Sau khi cài xong bạn kiểm tra lại bằng cách xem phiên bản của Java bằng lệnh:

```
java -version
```

Kết quả sẽ tựa như sau:

```
openjdk version "1.8.0_252"
OpenJDK Runtime Environment (build 1.8.0_252-8u252-b09-
1ubuntul-b09)
OpenJDK 64-Bit Server VM (build 25.252-b09, mixed mode)
```

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

Vì đường dẫn Java khá dài nên bất tiện cho các bước cấu hình sau này. Vì thế tôi dùng lệnh link (viết tắt là ln) để ánh xạ thêm thư mục /opt/jdk để dễ nhớ.

Ánh xạ thư mục java vào /opt/jdk:

```
sudo ln -nsf /usr/lib/jvm/java-8-openjdk-amd64 /opt/jdk
```

Ánh xạ tiếp để dùng cho Java trong R:

```
sudo ln -nsf /opt/jdk /usr/lib/jvm/default-java
```

Thiết lập biến môi trường JAVA_HOME

```
sudo nano /etc/environment
```

Thêm dòng này:

```
export JAVA_HOME=/opt/jdk
```

Thực thi lệnh sau để thiết lập trên có tác dụng:

```
source /etc/environment
```

Kiểm tra biến môi trường JAVA_HOME

```
echo $JAVA_HOME
```

Cài đặt SSH Server cho Ubuntu

Hãy tưởng tượng máy ảo Ubuntu mà bạn vừa cài thì nó cũng giống y chang như một cái máy mà bạn thuê trên Internet. Bạn có thể đã nghe thuật ngữ VPS, Cloud VPS (VPS = Virtual Private Server). Để làm việc từ xa với máy ảo này thì cần cấu hình thêm SSH Server một chút.

Đầu tiên kiểm tra dịch vụ SSH bằng lệnh sau:

```
sudo systemctl status ssh.service
```

Kết quả:

```
Unit ssh.service could not be found.
```

Cài đặt ssh

```
sudo apt install ssh
```

(Nhấn Y theo gợi ý trong quá trình cài đặt)

Cài đặt xong kiểm tra lại bằng lệnh sau:

```
sudo service ssh status
```

```
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
  Active: active (running) since Thu 2020-05-07 21:47:33 PDT; 31s ago
    Docs: man:sshd(8)
          man:sshd_config(5)
   Main PID: 11833 (sshd)
     Tasks: 1 (limit: 2285)
    Memory: 1.3M
      CGroup: /system.slice/ssh.service
              └─11833 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

May 07 21:47:33 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
May 07 21:47:33 ubuntu sshd[11833]: Server listening on 0.0.0.0 port 22.
May 07 21:47:33 ubuntu sshd[11833]: Server listening on :: port 22.
May 07 21:47:33 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
```

Nếu bạn thấy chữ Active: active (running) có nghĩa là bạn đã cài thành công và ssh server đang chạy.

Chú ý trong quá trình xem status thì nhất nút Q để thoát.

Để cho phép truy cập từ xa thì bạn cần mở tường lửa trên Ubuntu bằng lệnh sau:

```
sudo ufw allow ssh
```

Làm cho biến môi trường này có hiệu lực

Thực hiện lệnh sau:

```
source /etc/environment
```

Kiểm tra lại giá trị biến môi trường JAVA_HOME bằng lệnh sau:

```
Echo $JAVA_HOME
```

Thiết lập mật khẩu cho tài khoản root

Sau khi cài xong thì nên thiết lập tài khoản root bằng lệnh sau:

```
sudo passwd root
```

Để cho phép đăng nhập là khoản root từ xa qua phần mềm SSH thì thực hiện lệnh sau:

```
sudo sed -i 's/#PermitRootLogin prohibit-
password/PermitRootLogin yes/' /etc/ssh/sshd_config
```

Sau đó khởi động lại SSH Server:

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

```
sudo service ssh restart
```

Cài tường lửa cho Ubuntu

```
sudo apt install firewalld
```

Sử dụng công cụ truy cập từ xa

Để làm việc từ xa thì trên máy chủ chạy dịch vụ SSH Server thì cần cài lên máy client (như Windows bạn đang làm việc) cài một phần mềm SSH Client. SSH Client phổ biến trên Windows là Putty. Bạn vào trang <https://www.putty.org/>

The screenshot shows a web browser window with the URL <https://www.putty.org/>. On the left, there is a screenshot of the Putty Configuration dialog box. On the right, there is text and a download link:

Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

You can download PuTTY [here](#).

Vào link download PuTTY trong hình trên để download file putty.exe về chạy ngay. Tùy theo máy bạn là 32-bit hoặc 64-bit thì tải file tương ứng.

The screenshot shows a web browser window with the URL <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. There is a green header bar with the text "Alternative binary files". Below it, there is text and download links:

The installer packages above will provide versions of all of these (except PuTTYtel), but you can download standalone binaries one by one if you prefer.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

putty.exe (the SSH and Telnet client itself)

32-bit: putty.exe	(or by FTP)
(signature)	
64-bit: putty.exe	(or by FTP)
(signature)	

pscp.exe (an SCP client, i.e. command-line secure file copy)

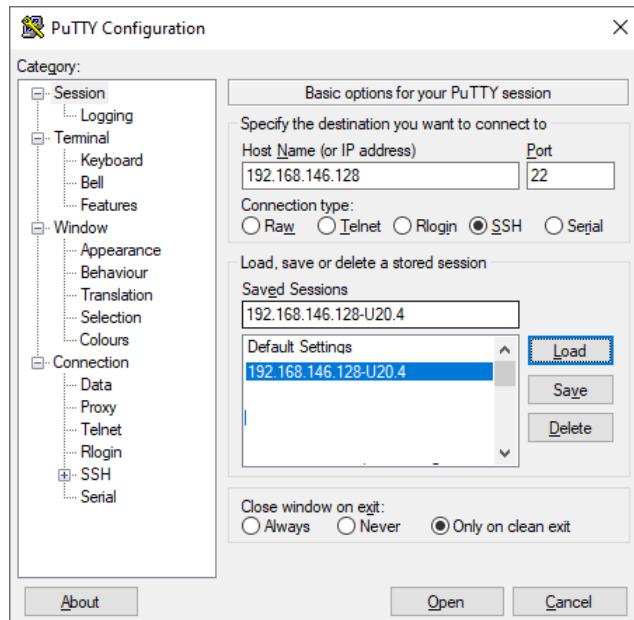
32-bit: pscp.exe	(or by FTP)
(signature)	
64-bit: pscp.exe	(or by FTP)
(signature)	

psftp.exe (an SFTP client, i.e. general file transfer sessions much like FTP)

32-bit: psftp.exe	(or by FTP)
(signature)	

Khởi động putty.exe và điền thông số: **địa chỉ IP** và **port** (mặc định SSH sẽ là port 22)

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



Nhấn nút Open để kết nối với server.

Điền **username** và **password** để đăng nhập. Sau khi đăng nhập thì ra cửa sổ lệnh như bên dưới. Cửa sổ lệnh này chính là Terminal chạy trên máy chủ. Vì vậy bạn làm việc qua cửa sổ lệnh này tức là đang ngồi làm việc trên máy chủ.

A terminal window titled 'root@ubuntu: ~'. The session starts with 'login as: root' followed by 'root@192.168.146.128's password:'. The user enters their password. The terminal then displays a welcome message for Ubuntu 20.04 LTS, including links for documentation, management, and support. It also mentions that Ubuntu 20.04 LTS is out and provides a link to the blog post. The terminal then shows update information: '0 updates can be installed immediately.' and '0 of these updates are security updates.' Finally, it shows system status: 'Your Hardware Enablement Stack (HWE) is supported until April 2025.' and 'Last login: Sun May 10 21:05:24 2020 from 192.168.146.1'. The prompt at the end is 'root@ubuntu:~#'. A small green bar is visible at the bottom of the terminal window.

Cài đặt MySQL server trên Ubuntu

Cập nhật hệ thống

```
sudo apt update
```

Cài đặt gói phần mềm mysql server

```
sudo apt install mysql-server
```

Thiết lập security

```
sudo mysql_secure_installation
```

(Làm theo hướng dẫn trên màn hình và ghi nhớ mật khẩu của account mysql root).

Đăng nhập vào cửa sổ lệnh mysql

```
sudo mysql
```

Tham khảo thêm tại:

<https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04>

Bài 20: Cài đặt Hadoop 3.2

Bài này sẽ giúp bạn cài đặt Hadoop 3.2 trong Ubuntu để làm quen và trải nghiệm. Bạn cũng có thể áp dụng các lệnh tương tự cho môi trường Linux, MacOS hoặc Ubuntu trong Windows.

Có vài tình huống bạn muốn sử dụng Hadoop 2 thì tham khảo bài viết trong phần Phụ lục.

Tải phần mềm

Bạn vào trang web <https://hadoop.apache.org/releases.html> để xem các phiên bản hiện tại của Hadoop.

Version	Release date	Source download	Binary download	Release notes
2.10.0	2019 Oct 29	source (checksum signature)	binary (checksum signature)	Announcement
3.1.3	2019 Oct 21	source (checksum signature)	binary (checksum signature)	Announcement
3.2.1	2019 Sep 22	source (checksum signature)	binary (checksum signature)	Announcement
2.9.2	2018 Nov 19	source (checksum signature)	binary (checksum signature)	Announcement

To verify Hadoop releases using GPG:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a [mirror site](#).
2. Download the signature file hadoop-X.Y.Z-src.tar.gz.asc from [Apache](#).
3. Download the [Hadoop KEYS](#) file.
4. gpg --import KEYS
5. gpg --verify hadoop-X.Y.Z-src.tar.gz.asc

To perform a quick check using SHA-512:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a [mirror site](#).

Phần này sẽ giúp bạn cài nhanh Hadoop phiên bản 3.2 lên máy ảo Ubuntu.

Tạo thư mục soft trong thư mục home của user (dùng kí hiệu dấu ngã ~)

```
mkdir ~/soft
```

Chuyển thư mục hiện hành vào thư mục soft mới tạo

```
cd ~/soft
```

Tải gói phần mềm hadoop phiên bản 3.2.0 về thư mục hiện hành bằng lệnh wget <url>:

```
wget https://archive.apache.org/dist/hadoop/common/hadoop-3.2.0/hadoop-3.2.0.tar.gz
```

Giải nén

Giải nén ra thư mục /opt

```
sudo tar -xvzf ./hadoop-3.2.0.tar.gz -C /opt
```

Tạo ảnh xạ thư mục /opt/hadoop-3.2.0 vào /opt/hadoop. Bước này giống như tạo shortcut trên Windows, thay vì truy cập vào đường dẫn dài /opt/hadoop-3.2.0 thì tôi tạo một đường dẫn ngắn hơn gọi là alias hoặc shortcut /opt/hadoop. Ngoài ra khi cần thử thử nghiệm các phiên bản hadoop khác nhau thì chỉ cần ánh xạ lại khi cần. Sử dụng lệnh link (ln)

```
sudo ln -nsf /opt/hadoop-3.2.0 /opt/hadoop
```

Kiểm tra lại nội dung thư mục bằng lệnh list (ls):

```
ls /opt/hadoop
```

```
bin include libexec NOTICE.txt sbin  
etc lib LICENSE.txt README.txt share
```

Cấu hình các biến môi trường cho Hadoop

Sửa file environment bằng lệnh:

```
sudo nano /etc/environment
```

Thêm nội dung được bôi đậm:

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/  
bin:/usr/games:/usr/local/games:/opt/hadoop/bin"  
  
JAVA_HOME=/opt/jdk  
HADOOP_HOME=/opt/hadoop  
HADOOP_MAPRED_HOME=/opt/hadoop  
HADOOP_CONF_DIR=/opt/hadoop/etc/hadoop/  
HDFS_NAMENODE_USER="root"  
HDFS_DATANODE_USER="root"  
HDFS_SECONDARYNAMENODE_USER="root"  
YARN_RESOURCEMANAGER_USER="root"  
YARN_NODEMANAGER_USER="root"  
# Hai biến bên dưới để dùng cho rhdfs trong R  
HADOOP_COMMON_LIB_NATIVE_DIR=/opt/hadoop/lib/native  
HADOOP_CMD=/opt/hadoop/bin/hadoop
```

Làm cho các thiết lập biến môi trường ở trên có tác dụng ngay luôn bằng lệnh:

```
source /etc/environment
```

Kiểm tra bằng cách xem giá trị của biến môi trường HADOOP_HOME bằng lệnh:

```
echo $HADOOP_HOME
```

Kết quả:

```
/opt/hadoop
```

Sửa file cấu hình của Hadoop

Sửa file core-site.xml bằng lệnh:

```
sudo nano /opt/hadoop/etc/hadoop/core-site.xml
```

Dán nội dung sau để thay thế cho nội dung 2 dòng <configuration></configuration> hiện tại:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://192.168.146.128:9000</value>
  </property>
</configuration>
```

Cách làm như sau:

Bước 1: Copy đoạn cấu hình ở trên bằng phím Ctrl + C

Bước 2: Chạy lệnh sudo nano... ở trên trong máy ảo Ubuntu, bạn di chuyển trong trỏ đến 2 dòng có thẻ <configuration> và </configuration> nhấn Ctrl + K để xóa.

Sau đó nhấn Ctrl + Shift + V để dán nội dung cấu hình vào file core-site.xml

Bước 3: Nhấn Ctrl + O để lưu

Bước 4: Nhấn Ctrl + X để thoát trình soạn thảo nano.

Thực hiện thay đổi file hdfs-site.xml với lệnh:

```
sudo nano /opt/hadoop/etc/hadoop/hdfs-site.xml
```

Với nội dung:

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

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.webhdfs.enabled</name>
        <value>true</value>
    </property>
</configuration>
```

Tiếp tục thực hiện sửa file mapred-site.xml với lệnh:

```
sudo nano /opt/hadoop/etc/hadoop/mapred-site.xml
```

Với nội dung:

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
        <name>mapreduce.application.classpath</name>
        <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
    </property>
</configuration>
```

Tiếp tục sửa file yarn-site.xml bằng lệnh:

```
sudo nano /opt/hadoop/etc/hadoop/yarn-site.xml
```

Với nội dung:

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
```

```
<property>
    <name>yarn.nodemanager.env-whitelist</name>

    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
</configuration>
```

Thiết lập khóa cho lệnh ssh

Cần chuyển tài khoản sang root để thực hiện phần này bằng lệnh sau:

```
su -l
```

Tiếp theo thực hiện lệnh ssh để kết nối từ xa qua SSH:

```
ssh localhost
```

Tiếp theo thực hiện 3 lệnh sau.

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys
```

Thực hiện lệnh exit hai lần để thoát ssh và trở về user bình thường.

Chuẩn bị dữ liệu cho Hadoop

```
sudo /opt/hadoop/bin/hdfs namenode -format
```

Khởi động hadoop

```
sudo /opt/hadoop/sbin/start-all.sh
```

Xem qua kết quả log của hadoop

Xem thư mục log bằng lệnh list:

```
ls /opt/hadoop/logs
```

Kết quả:

```
hadoop-root-datanode-ubuntu.log  userlogs
hadoop-root-datanode-ubuntu.out  yarn-root-nodemanager-ubuntu.log
hadoop-root-namenode-ubuntu.log  yarn-root-nodemanager-ubuntu.out
hadoop-root-namenode-ubuntu.out  yarn-root-resourcemanager-ubuntu.log
```

Thử xem file log “hadoop-root-datanode-**ubuntu**.log”:

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

```
sudo nano /opt/hadoop/logs/hadoop-root-datanode-ubuntu.log
```

Chữ “ubuntu” được bôi đậm là tên máy của bạn, hãy thay thế vào, hoặc lúc gõ lệnh dùng phím tab để hiển thị ra tên file cho đúng.

Không cần phải hiểu hết file này chứa cái gì, bạn chỉ cần đọc lướt qua để cảm nhận hadoop khi khởi động lên, nó ghi chú lại kết quả cho chúng ta biết nó khởi động như thế nào. Quá trình ghi chú của các phần mềm thì người ta gọi là **logging**, file được ghi chú ra gọi là **file log**.

Bạn sẽ thấy có dòng log có port 9864:

```
INFO org.apache.hadoop.hdfs.server.datanode.web.DatanodeHttpServer: Listening HTTP traffic on /0.0.0.0:9864
```

Bấm Ctrl + X để thoát lệnh nano.

Xem tiếp file log “hadoop-root-namenode-ubuntu.log”:

```
nano /opt/hadoop/logs/hadoop-root-namenode-ubuntu.log
```

Bạn sẽ thấy có dòng log có port 9000 như sau:

```
INFO org.apache.hadoop.ipc.Server: IPC Server Responder: starting
INFO org.apache.hadoop.ipc.Server: IPC Server listener on 9000: starting
INFO org.apache.hadoop.hdfs.server.namenode.NameNode: NameNode RPC up at: 192.168.146.128/192.168.146.128:9000
INFO org.apache.hadoop.hdfs.server.namenode.FSNamesystem: Starting services required for active state
INFO org.apache.hadoop.hdfs.server.namenode.FSDirectory: Initializing quota with 4 thread(s)
INFO org.apache.hadoop.hdfs.server.namenode.FSDirectory: Quota initialization completed in 58 milliseconds
```

Mở tường lửa để truy cập Hadoop từ xa

Câu hỏi đặt ra là bạn có thể truy cập vào Hadoop đã cài ở trên từ cái máy thật Windows được không? Câu trả lời là được nếu Ubuntu cho phép.

Chúng ta cho phép bằng cách mở port bằng các lệnh sau:

```
sudo firewall-cmd --permanent --add-port=9864/tcp
sudo firewall-cmd --permanent --add-port=9000/tcp
sudo firewall-cmd --permanent --add-port=50075/tcp
sudo firewall-cmd --permanent --add-port=8088/tcp
sudo firewall-cmd --reload
```

Kiểm tra lại các port đã mở trên máy

Cài đặt nmap

```
sudo apt install nmap
```

Quét port

```
sudo nmap -sT -O localhost
```

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

Truy cập Hadoop từ trình duyệt

Từ máy tính chạy Windows, bạn mở trình duyệt truy cập vào địa chỉ <http://192.168.146.128:9864/>

Hadoop Overview Utilities ▾

DataNode on server3:9866

Cluster ID:	CID-c377adbf-7b3b-4e8d-bd04-27d7af70d56b
Version:	3.2.0, re97acb3bd8f3befd27418996fa5d4b50bf2e17bf

Block Pools

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
server3:9000	BP-839391685-192.168.135.132-1609292448693	RUNNING	2s	32 minutes	0 B (64 MB)

Volume Information

Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
/tmp/hadoop-root/dfs/data	DISK	32 KB	29.9 GB	0 B	0 B	0

Hadoop, 2019.

Truy cập <http://192.168.146.128:8088/>

 All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory U
0	0	0	0	0	0 B

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes
1	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Reason

No data available

Showing 0 to 0 of 0 entries

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

Như vậy đến đây, trong tay các bạn đã có một hệ thống Big Data với phần mềm Hadoop chạy trên máy ảo Ubuntu. Gọi là Big Data System nhưng chưa có data gì hết và chưa biết nếu dùng R hoặc Python thì phân tích dữ liệu trên Hadoop này như thế nào?

Có nhiều câu hỏi cần phải trả lời. Nhưng thôi, hãy dừng lại và ăn mừng thành quả mà chúng ta đã học và làm được cái đã!

Khởi động lại Hadoop

Khi tắt và bật lại máy ảo Ubuntu thì bạn cần chạy lại Hadoop bằng các lệnh sau:

```
su -l  
cd /opt/hadoop/sbin  
rm -frd ..../logs/*  
./start-all.sh  
tail -f ..../logs/hadoop-root-datanode-ubuntu.log
```

Chủ động dừng Hadoop

Tài liệu ở trên đã hướng dẫn bạn cài đặt, khởi động và trải nghiệm nhanh Hadoop. Khi cần dừng chạy Hadoop thì bạn thực hiện lệnh sau:

```
sudo /opt/hadoop/sbin/stop-all.sh
```

Bài 21: Trải nghiệm Hadoop với Python

Ánh xạ địa chỉ IP để truy cập máy ảo từ máy host

Trong Ubuntu bạn gõ lệnh hostname sẽ thấy tên máy, và gõ ip a sẽ thấy địa chỉ IP:

```
hostname  
ubuntu  
ip a  
...  
inet 192.168.146.128/24  
...
```

Việc tiếp theo là trên máy chính của mình (gọi tắt là máy host) đang chạy Windows cần cấu hình để ánh xạ địa chỉ IP và hostname của Ubuntu ở trên bằng cách sau:

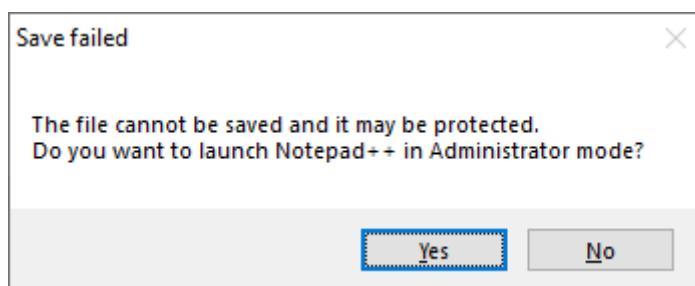
Dùng Notepad++ mở file:

C:\Windows\System32\drivers\etc\hosts

Thêm dòng sau vào file:

```
192.168.146.128 ubuntu
```

Bấm Ctrl + S trong Notepad++ để lưu. Tuy nhiên Notepad++ sẽ hỏi bạn:



Chọn Yes. Sau đó có thể bị hỏi tiếp và Yes một lần nữa. Lúc này hãy bấm Ctrl + S để lưu file.

Từ lúc này trở đi, bạn có thể đứng từ máy Windows, truy cập vào Hadoop trên Ubuntu thông qua tên máy như sau:

http://ubuntu:50070/

http://ubuntu:50075/

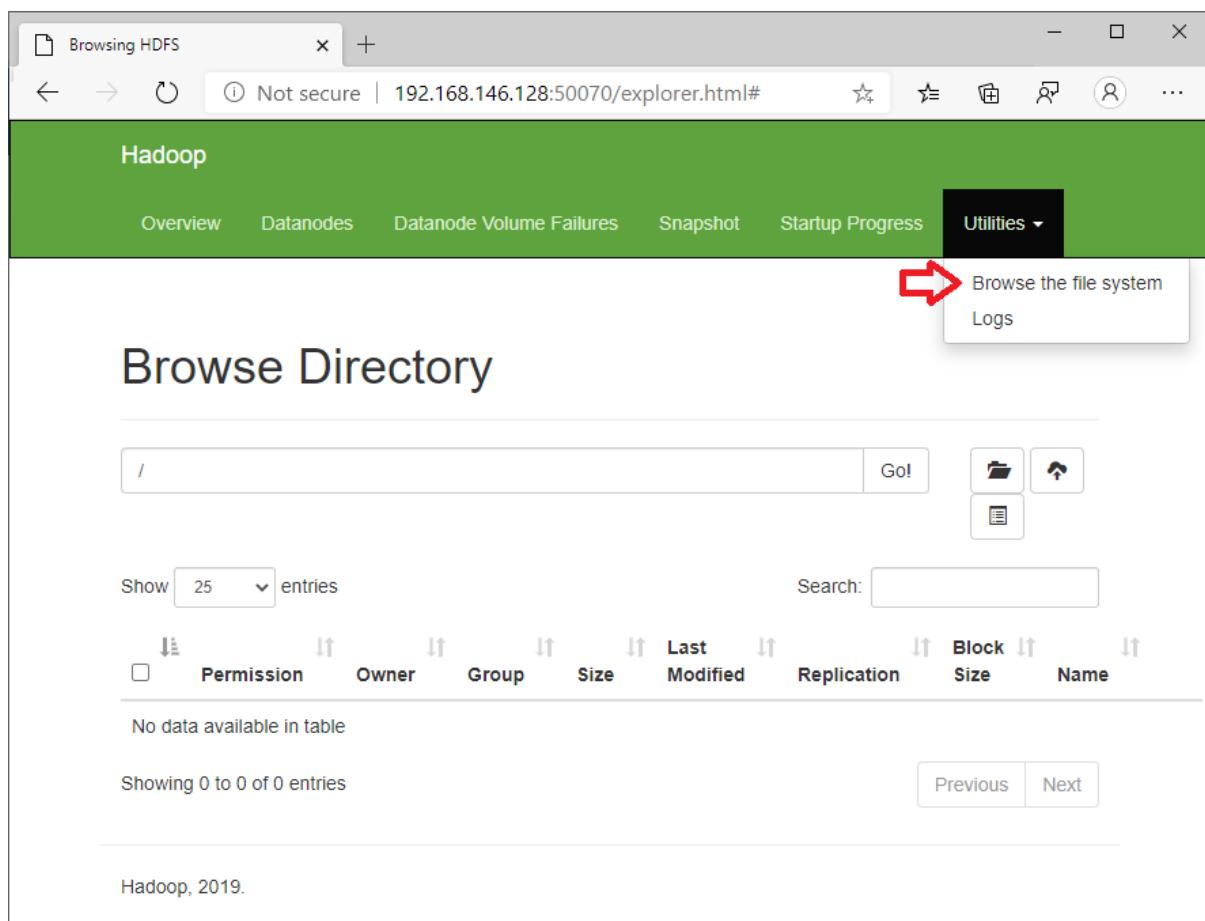
http://ubuntu:8088/

Trải nghiệm Hadoop từ xa

Để thử truy cập Hadoop từ xa thì dùng công cụ curl. Trên Ubuntu cài đặt curl như sau:

```
apt install curl
```

Lệnh sau đây sẽ tạo thư mục “mydemo1” bên trong thư mục “/user/root” của Hadoop. Trước khi thực hiện lệnh này, bạn vào trình duyệt với địa chỉ <http://192.168.146.128:50070/>, vào menu Utilities > Browse the file system để xem dữ liệu.



The screenshot shows a web browser window titled "Browsing HDFS". The address bar shows "Not secure | 192.168.146.128:50070/explorer.html#". The main content area is titled "Hadoop" and "Browse Directory". It features a search bar with a "/" prefix and a "Go!" button, along with icons for folder, file, and upload. Below the search bar are buttons for "Show 25 entries" and "Search". A table header with columns: "Permission", "Owner", "Group", "Size", "Last Modified", "Replication", "Block Size", and "Name". A message "No data available in table" is displayed. At the bottom, it says "Showing 0 to 0 of 0 entries" and has "Previous" and "Next" buttons. A footer at the bottom left says "Hadoop, 2019."

```
curl -i -X PUT curl -i -X PUT  
"http://192.168.146.128:50070/webhdfs/v1/user/root/mydemo1?op=M  
KDIRS&user.name=root"
```

Kết quả thực hiện lệnh:

```
HTTP/1.1 200 OK  
Cache-Control: no-cache  
Expires: Fri, 08 May 2020 08:15:04 GMT  
Date: Fri, 08 May 2020 08:15:04 GMT  
Pragma: no-cache  
Expires: Fri, 08 May 2020 08:15:04 GMT  
Date: Fri, 08 May 2020 08:15:04 GMT
```

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

```
Pragma: no-cache
Content-Type: application/json
X-FRAME-OPTIONS: SAMEORIGIN
Set-Cookie: hadoop.auth="u=root&p=root&t=simple&e=1588961704808&s=41DZivv8Quhz6
WpfMo1QCBGLtspjlovfUKH2Uhkyxvc="; Path=/; HttpOnly
Transfer-Encoding: chunked
```

Theo dõi dữ liệu Hadoop qua trình duyệt:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	May 08 15:15	0	0 B	mydemo1

Hãy thử xóa thư mục vừa tạo bằng lệnh sau:

```
curl -i -X DELETE
"http://192.168.146.128:50070/webhdfs/v1/user/root/mydemo1?op=DELETE&user.name=root"
```

Theo dõi tiếp trên browser thì bạn thấy thư mục “mydemo1” đã bị xóa. Tuy nhiên thư mục /user/root vẫn còn. Như vậy lệnh tạo thư mục thì tạo một đường dẫn mà thư mục cha không cần phải có trước, Hadoop sẽ tự tạo.

Hãy thử xóa luôn thư mục “root” và “user”!

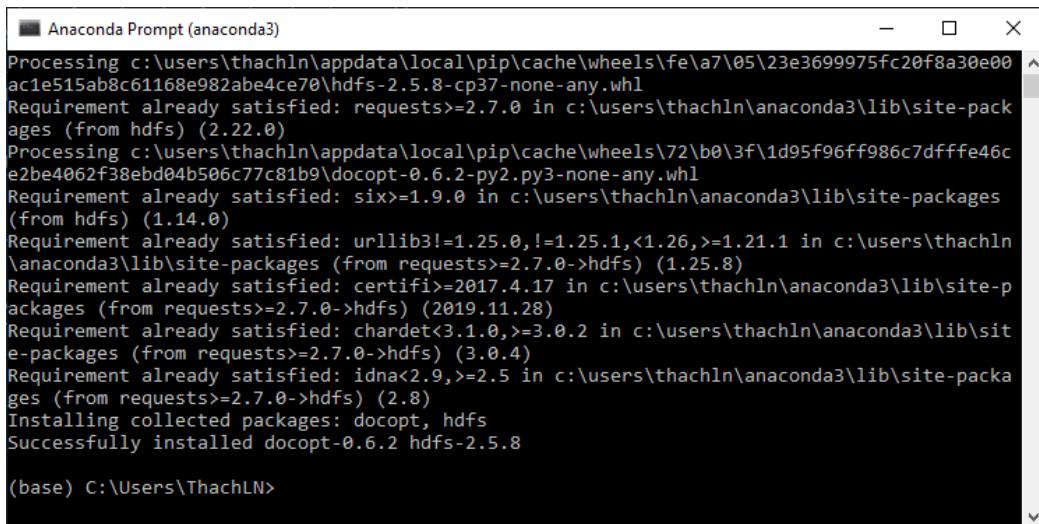
Lưu trữ dữ liệu lên Hadoop

Để minh họa cách sử dụng Hadoop cho phân tích dữ liệu, phần này sẽ dùng Python.

Đầu tiên bạn mở dấu nhắc Python của Anaconda để cài thư viện hdfs:

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

```
pip install hdfs
```



```
Anaconda Prompt (anaconda3)
Processing c:\users\thachln\appdata\local\pip\cache\wheels\fe\05\23e3699975fc20f8a30e00
ac1e515ab8c61168e982abe4ce70\hdfs-2.5.8-cp37-none-any.whl
Requirement already satisfied: requests>=2.7.0 in c:\users\thachln\anaconda3\lib\site-packages (from hdfs) (2.22.0)
Processing c:\users\thachln\appdata\local\pip\cache\wheels\72\b0\3f\1d95f96ff986c7dfffe46c
e2be4062f38ebd04b506c77c81b9\docopt-0.6.2-py2.py3-none-any.whl
Requirement already satisfied: six>=1.9.0 in c:\users\thachln\anaconda3\lib\site-packages (from hdfs) (1.14.0)
Requirement already satisfied: urllib3!=1.25.0,!>=1.25.1,<1.26,>=1.21.1 in c:\users\thachln\anaconda3\lib\site-packages (from requests>=2.7.0->hdfs) (1.25.8)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\thachln\anaconda3\lib\site-packages (from requests>=2.7.0->hdfs) (2019.11.28)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\thachln\anaconda3\lib\site-packages (from requests>=2.7.0->hdfs) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\thachln\anaconda3\lib\site-packages (from requests>=2.7.0->hdfs) (2.8)
Installing collected packages: docopt, hdfs
Successfully installed docopt-0.6.2 hdfs-2.5.8

(base) C:\Users\ThachLN
```

Tiếp theo mở Spyder và chạy đoạn code sau:

```
import pandas as pd
from hdfs import InsecureClient

# Đọc dữ liệu mẫu
df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', sep=';')

# Kết nối vào Hadoop
client_hdfs = InsecureClient('http://192.168.146.128:50070',
user='root')

# Lưu Dataframe vào Hadoop
with client_hdfs.write('/user/root/datasets/bank-additional-full.csv',
encoding = 'utf-8') as writer:
    df.to_csv(writer)
```

Như nội dung chú thích trong source code, sau khi thực thi xong thì chúng ta mong đợi dữ liệu Bank Marketing sẽ được lưu lên hệ thống Hadoop với đường dẫn “/root/datasets/bank-additional-full.csv”. Chú ý đường dẫn này bắt đầu bằng dấu xuyệt phẩy (/ đọc là root, ý là thư mục gốc, khác với tài khoản root nhé). Thư mục gốc này là trên hệ thống Hadoop nhé. Cần phân biệt với thư mục gốc trong đĩa cứng của Ubuntu.

Nếu bạn may mắn thì kết quả như sau:

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

Browsing HDFS

Hadoop

Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user/root

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rwxr-Xr-X	root	supergroup	5.17 MB	May 08 16:04	1	128 MB	bank-additional-full.csv

Showing 1 to 1 of 1 entries

Có thể bấm vào tên file “bank-additional-full.csv” Download, xem nhanh đầu file (Head the file..) và cuối file (Tail the file...):

Browsing HDFS

Hadoop

File information - bank-additional-full.csv

Download Head the file (first 32K) Tail the file (last 32K)

Block information - Block 0

Block ID: 1073741827
Block Pool ID: BP-1233815817-127.0.1.1-1588920843901
Generation Stamp: 1003
Size: 5423882
Availability:

- ubuntu

File contents

1.94.601-49.5.0.982.4963.6 ves

Xóa dữ liệu trên Hadoop

Trên Ubuntu để xóa thư mục và các dữ liệu bên trong thì dùng lệnh hdfs. Ví dụ để xóa dữ liệu bên trong thư mục của user root và xóa luôn thư mục “root” thì thực hiện lệnh sau:

```
hdfs dfs -rm -R /user/root
```

Phân tích dữ liệu từ Hadoop

Đọc dữ liệu từ Hadoop bằng Python:

```
import pandas as pd
from hdfs import InsecureClient

# Kết nối vào Hadoop
client_hdfs = InsecureClient('http://192.168.146.128:50070', user =
'root')

# Đọc dữ liệu từ Hadoop
with client_hdfs.read('/user/root/bank-additional-full.csv', encoding =
'utf-8') as reader:
    df = pd.read_csv(reader, index_col=0)

df.head()
```