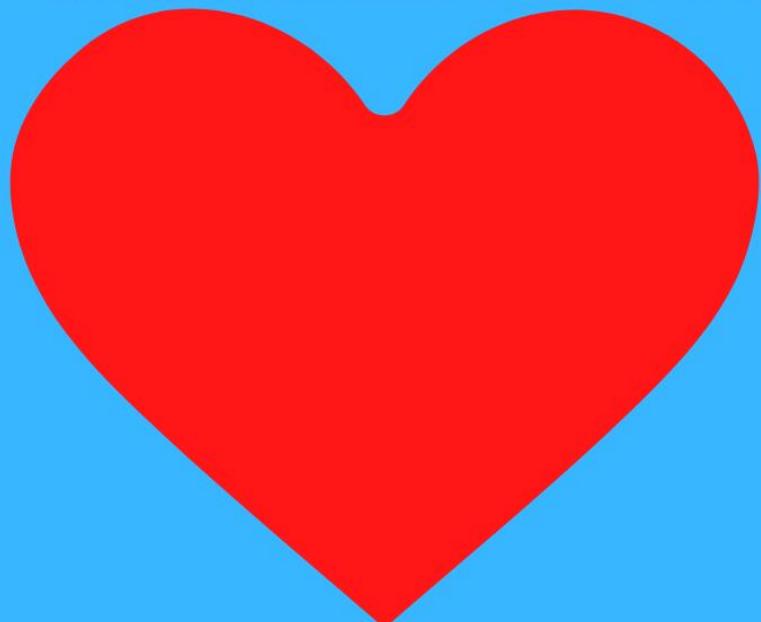


LÊ NGỌC THẠCH

Chạm tới AI trong 10 ngày

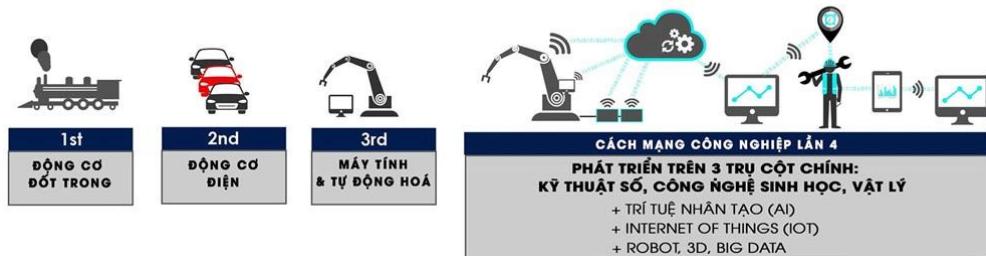


Đây là eBook của riêng bạn – đề nghị không chia sẻ cho ai khác!

Giới thiệu

Hiện tại tôi cảm thấy rất may mắn sống trong cuộc cách mạng công nghiệp lần thứ 4. Từ khóa mà tôi, bạn và hầu hết mọi người được nghe nói tới gần đây là Trí tuệ nhân tạo (AI – Artificial Intelligent), một cuộc cách mạng rất đặc biệt.

LỊCH SỬ 4 CUỘC CÁCH MẠNG CÔNG NGHIỆP



(Nguồn: Internet)

Cuốn eBook này được ra đời trong bối cảnh của một người mong muốn mang kiến thức về khoa học máy tính để đem lại giá trị cho khách hàng của mình. Tính từ năm 1998 khi tôi đi làm bán thời gian trong thời sinh viên đến bây giờ thì khái niệm “khách hàng” trong suy nghĩ của tôi cũng thay đổi nhiều:

Ban đầu, khách hàng của tôi chính là **Khách hàng của công ty tôi** làm bán thời gian. Vài năm sau tôi hiểu và bổ sung thêm vào khái niệm Khách hàng bao gồm luôn cả **công ty mà tôi đã và đang làm**. Rồi vài năm sau nữa tôi hiểu các **Đồng nghiệp**, của tôi cũng chính là Khách hàng. Khi tôi nhận lời đi chia sẻ kiến thức, kinh nghiệm thực tiễn cho các lớp về Công nghệ thông tin thì **Sinh viên** chính là Khách hàng của tôi. Đặc biệt trong cuộc sống, tôi nhận ra **Người bạn đời** của mình cũng chính là Khách hàng của mình. Việc chăm sóc khách hàng như thế nào thì thú thật tôi phải học tập nhiều từ những người làm kinh doanh. Chủ đề khách hàng không nằm trong nội dung cuốn sách này. Nhưng qua đây, tôi rất cảm ơn các khách hàng của mình. Nếu không có họ thì cuốn eBook này khó có thể ra đời ngay tại thời điểm này.

Đặc biệt tôi biết ơn anh Nguyễn Văn Tuấn – người đã truyền cảm hứng cho tôi từ các lớp học, từ các chia sẻ của anh về Phân tích dữ liệu.

Tôi cảm ơn các Khách hàng đã thật sự tin cậy và đặt hàng ngay khi kế hoạch của eBook được phát thảo.

Về cách trình bày nội dung của cuốn sách tôi sẽ viết theo phong cách của người đi làm, các từ ngữ chuyên môn thì tôi cố gắng giữ nguyên bản gốc tiếng Anh. Vì vậy câu cú có thể bao gồm cả tiếng Anh và tiếng Việt với mục tiêu cốt lõi là để bạn đọc nắm được vấn đề và tham khảo thêm tài liệu tiếng Anh sau này cũng dễ dàng hơn.

Dù tôi sẽ cố gắng hết sức nhưng chắc chắn trong eBook này không tránh khỏi sai sót. Mọi góp ý xin nhắn về email ThachLN@gmail.com để phiên bản sau hoàn thiện hơn nữa.

Lê Ngọc Thạch

Chạm tới AI trong 10 ngày

Mục lục

| | |
|--|-----|
| Giới thiệu | ii |
| Quy ước | 6 |
| Ngày 1 – Chủ đề: Giới thiệu về ngôn ngữ thống kê, ngôn ngữ lập trình | 9 |
| Bài 1: Tóm tắt về thống kê (Statistics) | 11 |
| Bài 2: Ngôn ngữ lập trình Python và ngôn ngữ thống kê R | 17 |
| Bài 3: Ngôn ngữ R và phần mềm RStudio | 25 |
| Bài 4: Ngôn ngữ Python và phần mềm Anaconda | 40 |
| Bài 5: Cài đặt thêm phần mềm | 58 |
| Bài 6: Nhập liệu, biên tập, lưu trữ dữ liệu với R | 60 |
| Bài 7: Nhập liệu, biên tập, lưu trữ dữ liệu với Python | 70 |
| Ngày 2 – Chủ đề: Biểu đồ | 77 |
| Bài 8: Các loại biểu đồ | 79 |
| Bài 9: Vẽ biểu đồ trong R | 85 |
| Bài 10: Vẽ biểu đồ trong Python | 117 |
| Bài 11: Nguyên tắc soạn biểu đồ | 132 |
| Bài 12: Giới thiệu Matplotlib | 134 |
| Ngày 3 – Phân tích mô tả | 145 |
| Bài 13: Phân tích mô tả dữ liệu Bank Marketing | 147 |
| Bài 14: So sánh 2 tỉ lệ | 182 |
| Bài 15: Mô hình kiểm định giả thuyết | 193 |
| Bài 16: Ứng dụng minh họa | 194 |
| Ngày 4 – Chủ đề: Dữ liệu lớn | 201 |
| Bài 17: Cách xử lý tập hợp dữ liệu lớn | 202 |
| Bài 18: Sử dụng Ubuntu | 236 |
| Bài 19: Cài đặt Hadoop | 244 |
| Bài 20: Trải nghiệm Hadoop với R và Python | 253 |
| Ngày 5 – Chủ đề: Dự báo bằng mô hình hồi qui tuyến tính | 267 |
| Bài 21: Giới thiệu mô hình hồi qui tuyến tính | 268 |
| Bài 22: Diễn giải mô hình hồi qui tuyến tính | 275 |

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

| | |
|--|-----|
| Bài 23: Mô hình hồi qui tuyến tính đa biến..... | 290 |
| Bài 24: Tìm mô hình “tối ưu” | 295 |
| Bài 25: Dự báo bằng mô hình hồi qui tuyến tính | 298 |
| Ngày 6 – Chủ đề: Dự báo bằng mô hình hồi qui logistic | 302 |
| Bài 26: Giới thiệu mô hình hồi qui logistic..... | 303 |
| Bài 27: Mô hình hồi qui logistic đa biến (Multiple logistic regression model)..... | 309 |
| Bài 28: Tìm mô hình “tối ưu” | 314 |
| Bài 29: Dự báo bằng mô hình hồi qui logistic | 321 |
| Ngày 7 – Chủ đề: Phân tích đa biến | 328 |
| Bài 30: Xử lý giá trị trống | 329 |
| Bài 31: Mô hình phân tích phân định (Linear discriminant analysis).. | 339 |
| Bài 32: Mô hình thành phần (Principal Component Analysis) | 349 |
| Bài 33: Mô hình phân tích cụm/nhóm (cluster analysis) | 359 |
| Ngày 8 – Chủ đề: Machine Learning | 370 |
| Bài 34: Giới thiệu Machine learning | 371 |
| Bài 35: Mô hình SVM | 373 |
| Bài 36: Mô hình Random Forest | 386 |
| Bài 37: Mô hình Artificial Neural Network | 391 |
| Bài 38: Machine Learning với Python Tensorflow..... | 397 |
| Ngày 9 – Chủ đề: Recommendation..... | 426 |
| Bài 39: Giới thiệu phương pháp gợi ý Collaborative filtering | 427 |
| Bài 40: Triển phương pháp gợi ý Collaborative filtering bằng R | 437 |
| Ngày 10 – Chủ đề: Natural Language Processing..... | 443 |
| Bài 41: Các kỹ thuật cơ bản | 444 |
| Bài 42: Trích đặc trưng (Feature extraction)..... | 449 |
| Bài 43: Giới thiệu ứng dụng phân tích cảm xúc (Sentiment Analysis) | 459 |
| Bài 44: Giới thiệu ứng dụng phân tích từ vựng (Word Embedding) ... | 470 |
| Bài 45: Giới thiệu ứng dụng xác định chủ đề (Topic Modeling) | 480 |
| Ngày 11 – Chủ đề: Computer Vision | 489 |
| Bài 46: Giới thiệu Face recognition | 490 |
| Bài 47: Giới thiệu mô hình CNN | 505 |

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

| | |
|--|-----|
| Ngày 12 – Chủ đề: Nhận diện tiếng nói (Speech Recognition) | 524 |
| Bài 48: Giới thiệu đặc trưng của âm thanh | 525 |
| Bài 49: Mô hình Chuyển giọng nói thành văn bản | 531 |
| Tạm kết thúc | 533 |

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 độc giả của ebook Chạm tới AI trong 10  
ngày.')  
print('Welcome to ebook Touch on AI in ten days.')  
print('{} + {} = {}'.format(1, 2, (1 + 2)))
```

Code cho Python thì cũng tương tự như trên nhưng màu viền bên trái sẽ là vàng đậm như sau:

```
print('Đây là code Python')
```

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

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

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

Lệnh thực thi trong hệ điều hành

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

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

```
pip install python-docx
```

Cặp dấu nháy

Các dữ liệu dạng chuỗi (string, text, char nói chung là có nghĩa giống nhau trong R và 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).

Chạm tới AI trong 10 ngày

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.

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 “quit()” trong cửa sổ console để thoát chương trình R Studio.* Trong câu hướng dẫn này thì lệnh quit() được gõ vào R Studio **KHÔNG** bao gồm cặp dấu nháy.

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 trình tự bấm chọn menu

Khi cần trình bày thứ tự các nút bấm, hoặc các mục cần bấm trong các thao tác thì sẽ dùng dấu lớn hơn >. Ví dụ khi hướng dẫn bạn vào trang web “<https://github.com/vncorenlp/VnCoreNLP>”, bấm vào nút “Clone”, sau đó bấm tiếp vào nút hoặc link “Download Zip” thì sẽ viết gọn như sau:

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

Các từ viết tắt tiếng Anh 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 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 phản ánh 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:\ai2020\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 thì dùng dấu xuyệt phải (right slash) để phân cách thư mục. Ví dụ: /mnt/d/ai2020.

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:

```
read.csv("D:/ai2020/data/test.csv")
```

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

```
read.csv("D:\\ai2020\\data\\\\test.csv")
```

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

Lời nhắn

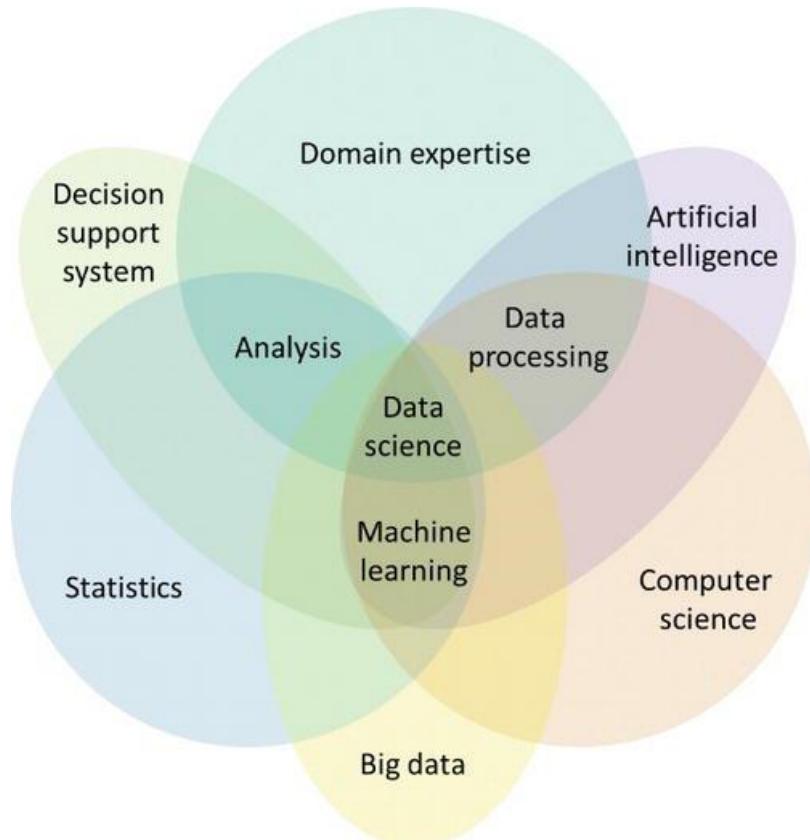
eBook này là của riêng bạn thông qua mua trực tiếp hoặc được tặng chính thức. Nên bạn được toàn quyền sử dụng và **KHÔNG** chia sẻ với bất kỳ ai khác nhé, **KHÔNG** lưu trữ trên internet nói chung để hạn chế đến tay người không thật sự cần nó nhé!

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.

Lê Ngọc Thạch

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 R và Python

Ngày đầu tiên sẽ gồm 7 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 R và Python

Bài 3: Hướng dẫn làm quen với ngôn ngữ R và phần mềm để thực hành R, RStudio

Bài 4: Hướng dẫn làm quen với ngôn ngữ Python và phần mềm để thực hành Anaconda, Spyder.

Bài 5: Chia sẻ thêm các trải nghiệm thực tế để giúp các bạn làm việc trên máy tính thuận tiện hơn.

Bài 6: Hướng dẫn chuẩn bị dữ liệu, các thao tác biên tập cơ bản và lưu trữ dữ liệu với R.

Bài 7: Hướng dẫn chuẩn bị dữ liệu, các thao tác biên tập cơ bản và lưu trữ dữ liệu với Python.

Với mục tiêu của tài liệu là giúp các bạn tiếp cận, hoặc là chạm tới lĩnh vực AI trong một thời gian rất ngắn – 10 ngày nên tài liệu khá ôm đòn khi mong muốn đáp ứng nhu cầu sử dụng R và Python để học tập về Phân tích dữ liệu, AI nói chung.

Đây là cuốn sách đầu tiên nêu thật sự tôi cũng không biết là cách trình bày vừa cả Python và R có phù hợp với bạn đọc hay không? Cho dù các bạn thấy bối rối một chút nhưng tôi tin là việc trình bày các khái niệm về Data analytics, AI rất đời thường và hướng dẫn các bạn trải nghiệm ngay trên các phần mềm sẽ giúp các bạn chạm tới hai lĩnh vực này một cách hiệu quả.

Bây giờ chúng ta có thể bắt đầu.

Bài 1: Tóm tắt về thống kê (Statistics)

Ôn tập khái niệm

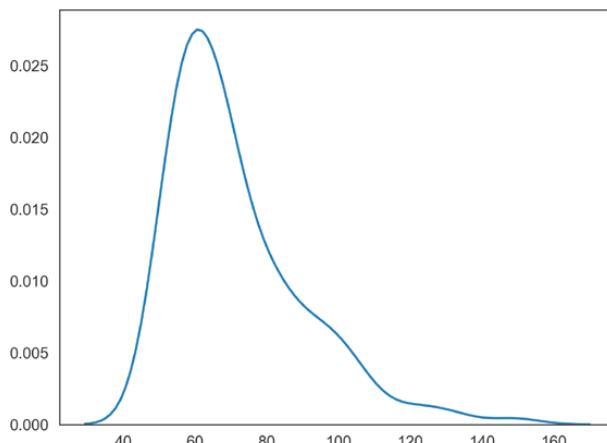
Thống kê (statistics) là công việc tổng hợp gồm nhiều việc nhỏ hơn như: **đặt câu hỏi, thu thập dữ liệu, trình bày dữ liệu, phân tích dữ liệu, diễn giải dữ liệu và suy diễn** (ra thông tin mới).

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:

Chạm tới AI trong 10 ngày

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

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.

Mode: 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 mười 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 (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 nếu tính tổng các độ lệch thì sẽ không phản ánh 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 phương sai là:

Chạm tới AI trong 10 ngày

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

Độ lệch chuẩn (**standard deviation**): được tính bằng căn bậc hai của phương sai.

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

Interquartile range: là khoảng giữa hai trung vị 25% và 75%.

Giá trị trung bình có trọng số của mẫu (sample)

Weighted Arithmetic Mean

Trung bình theo trọng số

Ví dụ: Theo dõi giá cổ phiếu VNM (Công ty Sữa Việt Nam) giá thấp nhất và cao nhất trong 5 ngày từ 6/9/2019 đến 12/9/2019

| Ngày | 12/9/2019 | 11/9/2019 | 10/9/2019 | 9/9/2019 | 6/9/2019 |
|---------------|-----------|-----------|-----------|----------|----------|
| Giá thấp nhất | 121.4 | 122.1 | 123.3 | 122.5 | 122.0 |
| Giá cao nhất | 123.3 | 124.0 | 124.3 | 123.9 | 122.9 |

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 tuyển 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.
- 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)

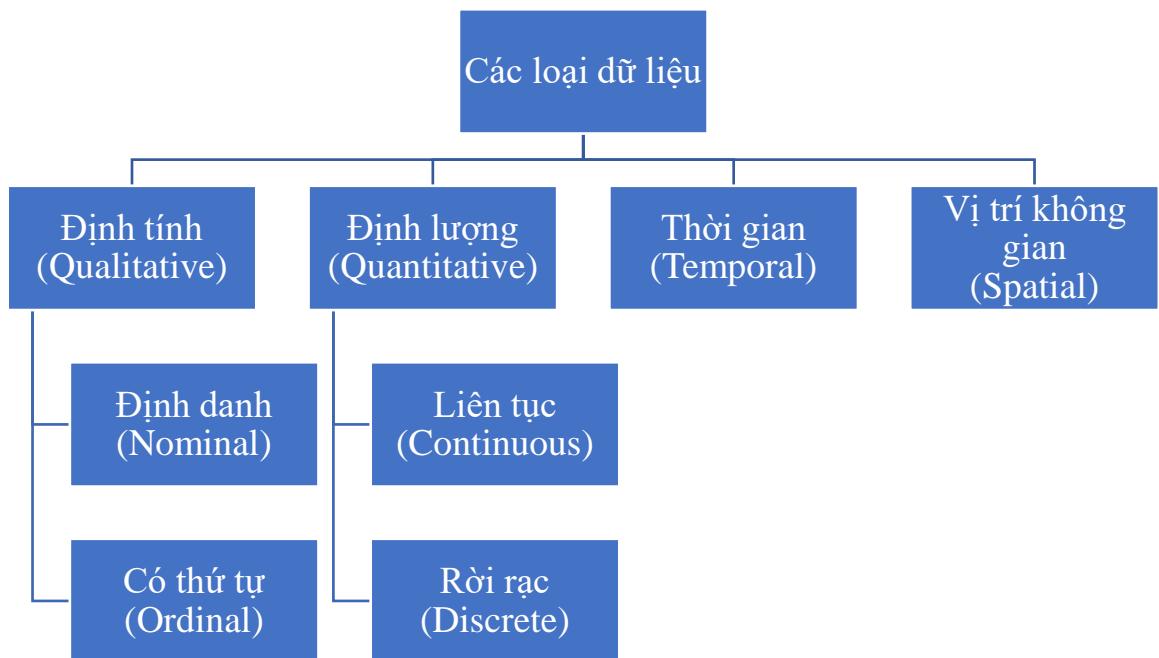
Ngoài hai dạng dữ liệu Định tính và Định lượng mà bạn thường gặp ở trên thì còn có hai loại khác như:

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 trực hai chiều.

Chạm tới AI trong 10 ngày

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:

- 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 xem màu nào nhiều nhất. Tức ra đường bạn sẽ thấy rất nhiều xe với nhiều màu khác nhau. Nhưng tựa 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 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. |
| Numerical | Mean/Median | 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 phụ thuộc (dependent variable). Đôi khi gọi là outcome.
- Biến độc lập (independent variable). Đôi khi gọi là biến tiên lượng (predictor variable)

Bài 2: Ngôn ngữ lập trình Python và ngôn ngữ thống kê R

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 hai ngôn ngữ là R và Python vừa đủ để các bạn trải nghiệm các khái niệm về thống kê, về kiểu dữ liệu đã học trong ngày hôm nay.

Dù Python và R có nhiều khác biệt nhưng phần này tôi sẽ giới thiệu khái niệm chung nhất và cơ bản nhất để bạn có thể làm quen nhanh chóng với Python và R.

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 đựng 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 cao |
| 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 |
|----------------------------|--------------|
| 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.

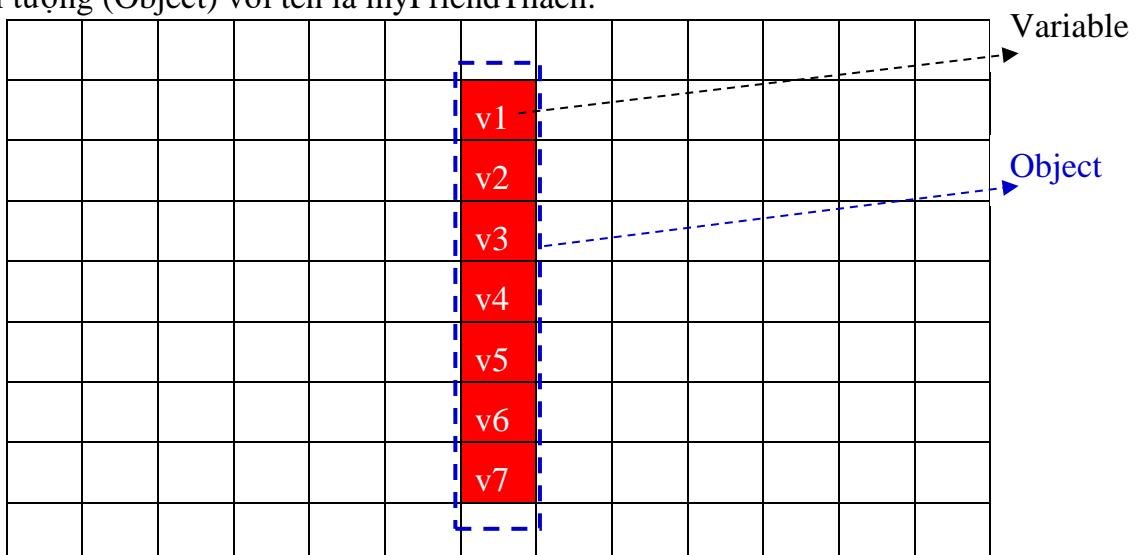
Object là một khái niệm gom nhiều loại thông tin để mô tả một vật, một người hay nói chung là một đối tượng nào đó. Nói cụ thể hơn là Object sẽ chứa trong nó nhiều biến. Chúng ta mô tả lại ví dụ trình bày thông tin cho người bạn “Thạch” của chúng ta ở trên dưới dạng một object như sau:

| Object: myFriendThach | |
|-----------------------|-----------------------------|
| fullName | Lê Ngọc Thạch |
| height | 165 cao |
| 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 gói 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.

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: <-

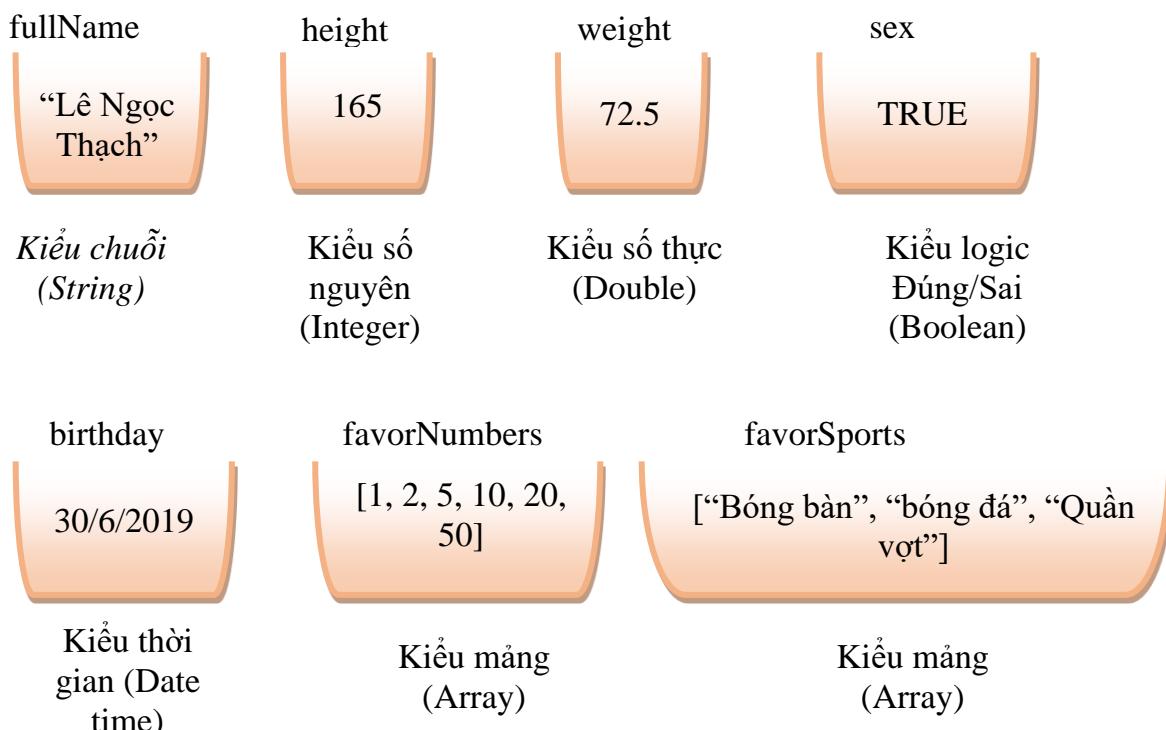
Để thuận tiện thì trong tài liệu này tôi sẽ dùng dấu = làm phép gán.

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)

Đối với bạn nào muốn thực hành ngay trong R hoặc Python thì hình dung các lệnh như sau:

Code R:

```
fullName = 'Lê Ngọc Thạch'  
height = 165  
weight = 72.5  
sex = TRUE  
birthday = as.Date('30/6/2019', format = '%d/%m/%Y')
```

Chạm tới AI trong 10 ngày

```
favorNumbers = c(1, 2, 5, 10, 20, 50)
favorSports = c('Bóng bàn', 'Bóng đá ', 'Quần vợt ')
```

Code Python:

```
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]
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.

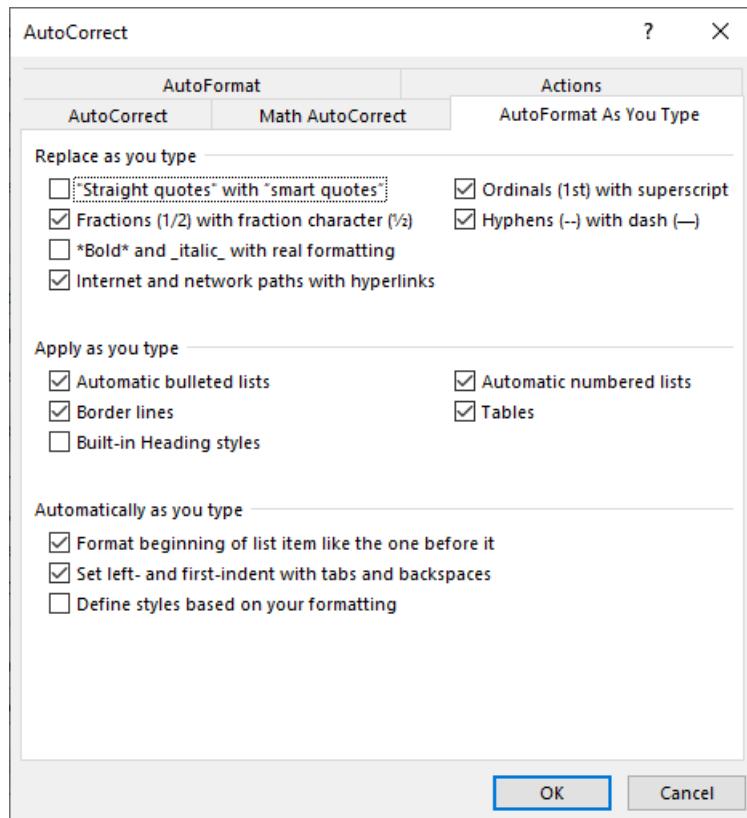


Nhấn phím nháy đơn sẽ nhanh hơn là nhấn phím nháy đôi (phải kèm thêm phím Shift). Vì vậy trong tài liệu này tôi sẽ dùng dấu nháy đơn để bao đóng các dữ liệu dạng chuỗi (string), hay còn gọi là văn bản (text).

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).



- Kiểu số thì cơ bản là giống nhau trong R và Python
- Kiểu luận lý (logical) thì các giá trị¹ 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 phần mềm R hiểu được giá trị thời gian bao gồm ngày tháng năm thì dùng lệnh

```
as.Date('30/6/2019', format = '%d/%m/%Y')
```

hoặc viết ngắn hơn

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

```
as.Date('30/6/2019', format = '%d/%m/%Y')
```

- Trong Python thì phải thêm một lệnh " import datetime" để sử dụng được lệnh

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

Chức năng của 2 lệnh này là báo cho phần mềm R và 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'.

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

%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, hoặc danh sách (List) thì trong R và Python có quy ước khác nhau.

Trong R, sử dụng lệnh c() để liệt kê các phần tử của mảng cách nhau bởi dấu phẩy.

```
favorNumbers = c(1, 2, 5, 10, 20, 50)
```

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 R và phần mềm RStudio

R có thể được hiểu theo hai nghĩa sau:

- Là một phần phần mềm: tức là chương trình trên máy tính cho phép bạn triển khai mã lệnh R.
- Là một ngôn ngữ thống kê, hoặc là ngôn ngữ lập trình: tức là bạn có thể viết các phần mềm phục vụ cho công việc thống kê, phân tích dữ liệu bằng ngôn ngữ lập trình R.

R là một ngôn ngữ lập trình mạnh và linh động (flexible and powerful language) cho việc phân tích dữ liệu chất lượng cao. Bạn có thể sử dụng R mà không cần phải là một lập trình viên hoặc là một chuyên gia về máy vi tính. Có kiến thức về lập trình là một lợi thế, tuy nhiên không phải là bắt buộc.

Cài đặt R

Tải chương trình cài đặt tại:

<https://www.r-project.org/>

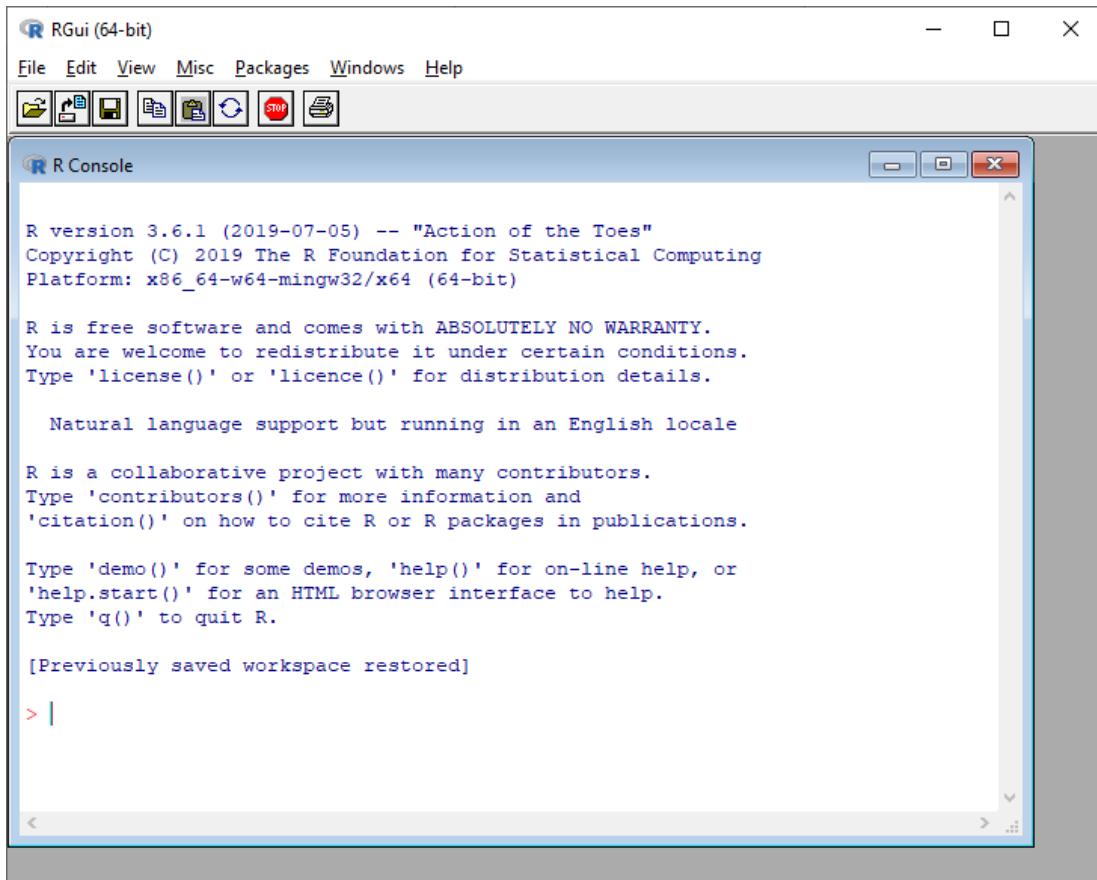
Tôi đang sử dụng R phiên bản 3.6.1 tại link:

<https://cloud.r-project.org/bin/windows/base/R-3.6.1-win.exe>.

Quá trình cài đặt phần mềm R cơ bản là rất đơn giản. Bạn cứ việc nhấn “Next” và “OK” theo gợi ý mặc định trong quá trình cài đặt là được. Sau khi cài đặt thì bạn có thể khởi động và sử dụng R ngay mà KHÔNG CẦN khởi động lại máy tính.

Giao diện của R như sau:

Chạm tới AI trong 10 ngày

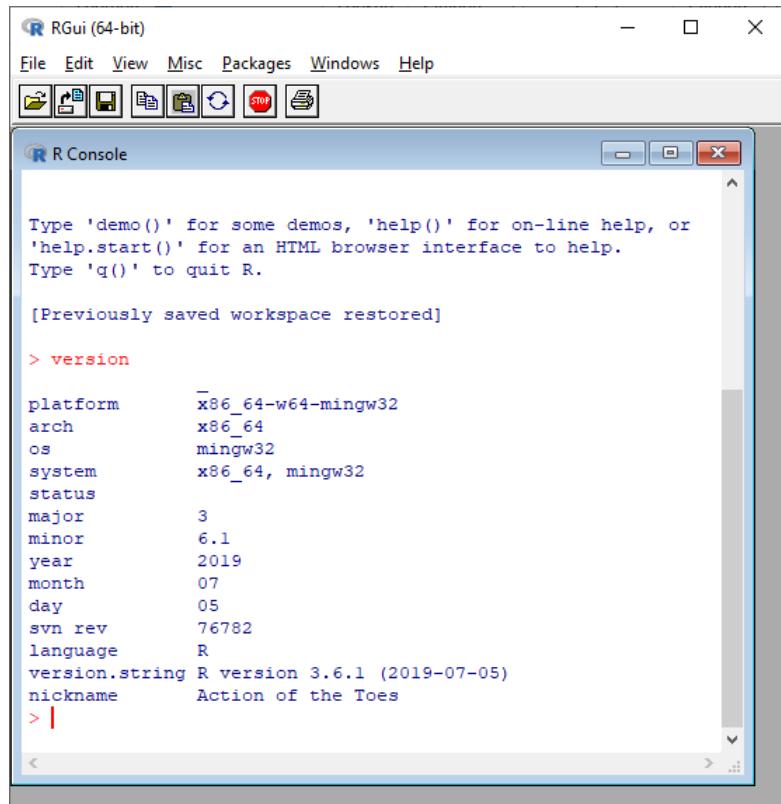


Nó có cửa sổ tên là R Console². Phía dưới Console có dấu mũi tên màu đỏ và một dấu gạch đứng gọi là con nháy (cursor). Cursor cho biết bạn gõ lệnh từ bàn phím vào tại vị trí nó đang đứng.

Ví dụ bạn gõ lệnh "version" (không có cặp dấu nháy đôi, tất cả là **chữ thường**) rồi nhấn Enter để R biết bạn đã gõ xong lệnh. Kết quả như sau:

² Trong lĩnh vực máy tính, “console” thông thường là cửa sổ để phần mềm hiển thị kết quả cho người dùng xem; hoặc là nơi mà người dùng có thể ra lệnh cho phần mềm làm việc; hoặc cả hai: vừa cho phép người dùng gõ lệnh và vừa hiển thị kết quả luôn.

Chạm tới AI trong 10 ngày

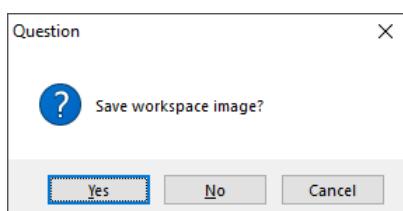


Tùy theo máy của bạn thì kết quả của lệnh “version” sẽ hiển thị ra sẽ khác.

Đến đây bạn nên học cách thoát khỏi phần mềm R. Có nhiều cách thoát khỏi một phần mềm như:

- Bấm chuột vào nút X (gọi là nút Close) ở góc phải trên của phần mềm.
- Vào menu File > Exit
- Gõ lệnh “quit()” (Không gõ gấp dấu nháy đôi và CÓ cặp dấu mở đóng ngoặc).
Lệnh nhanh hơn là “q()”

Trước khi thoát chương trình thì R hỏi như hộp thoại (dialog) bên dưới:



Bạn nhấn “No” để thoát.

| Đúc kết | |
|---|---|
| Bạn đã biết cài đặt R và gõ lệnh vào R Console! | |
| Các lệnh bạn nên thực hành | |
| version | Xem phiên bản của R và vài thông tin khác |
| q() | Thoát khỏi phần mềm R |

Cài đặt các gói phần mềm

Một trong các điểm mạnh của R là một phần mềm tổ chức dưới dạng package (các gói phần mềm được phát triển và bổ sung vào R rất thuận tiện), còn gọi là library (thư viện). Trong R có rất nhiều package cung cấp các chức năng cho thống kê nói riêng, phân tích dữ liệu nói chung. Cụ thể trong tài liệu này sẽ khai thác vài package thường được sử dụng như sau:

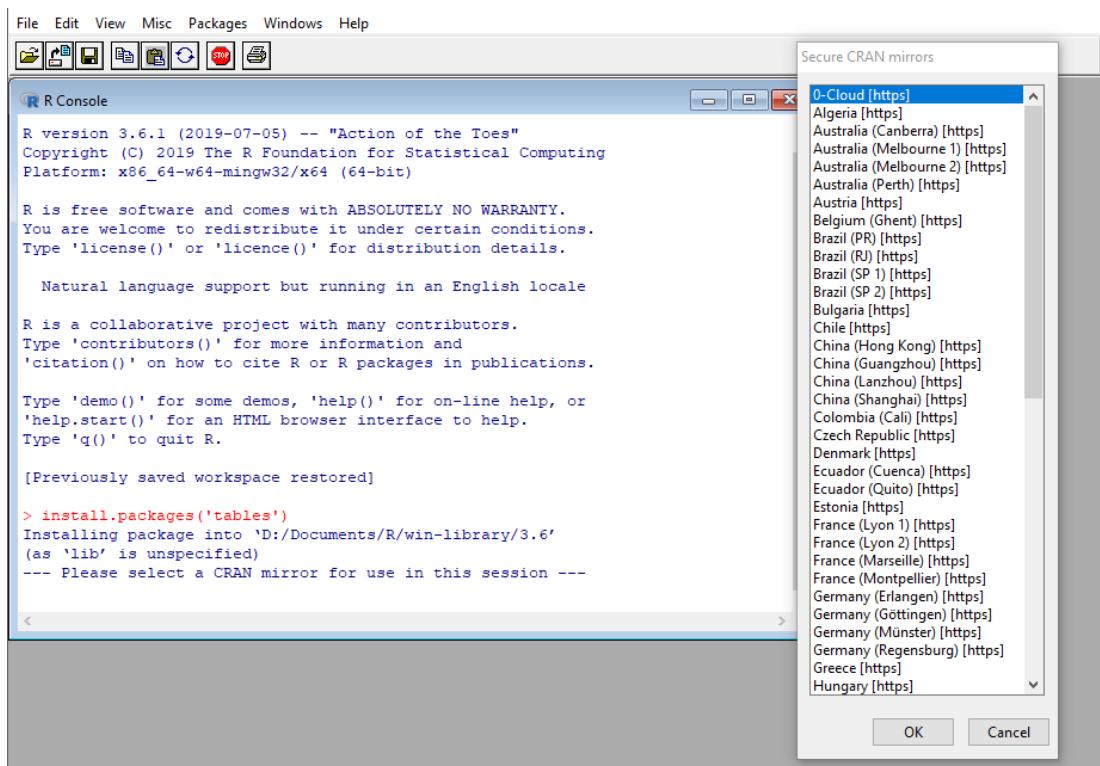
| Tên package | Chức năng |
|-------------|--|
| BMA | Cài đặt phương pháp chọn mô hình bayes Bayes |
| ggplot2 | Dùng để vẽ biểu đồ |
| tables | Phân tích dữ liệu dưới dạng bảng số liệu |
| Xlsx | Đọc dữ liệu từ file Excel |

Để cài đặt package thì dùng lệnh:

```
install.packages('<tên package>')
```

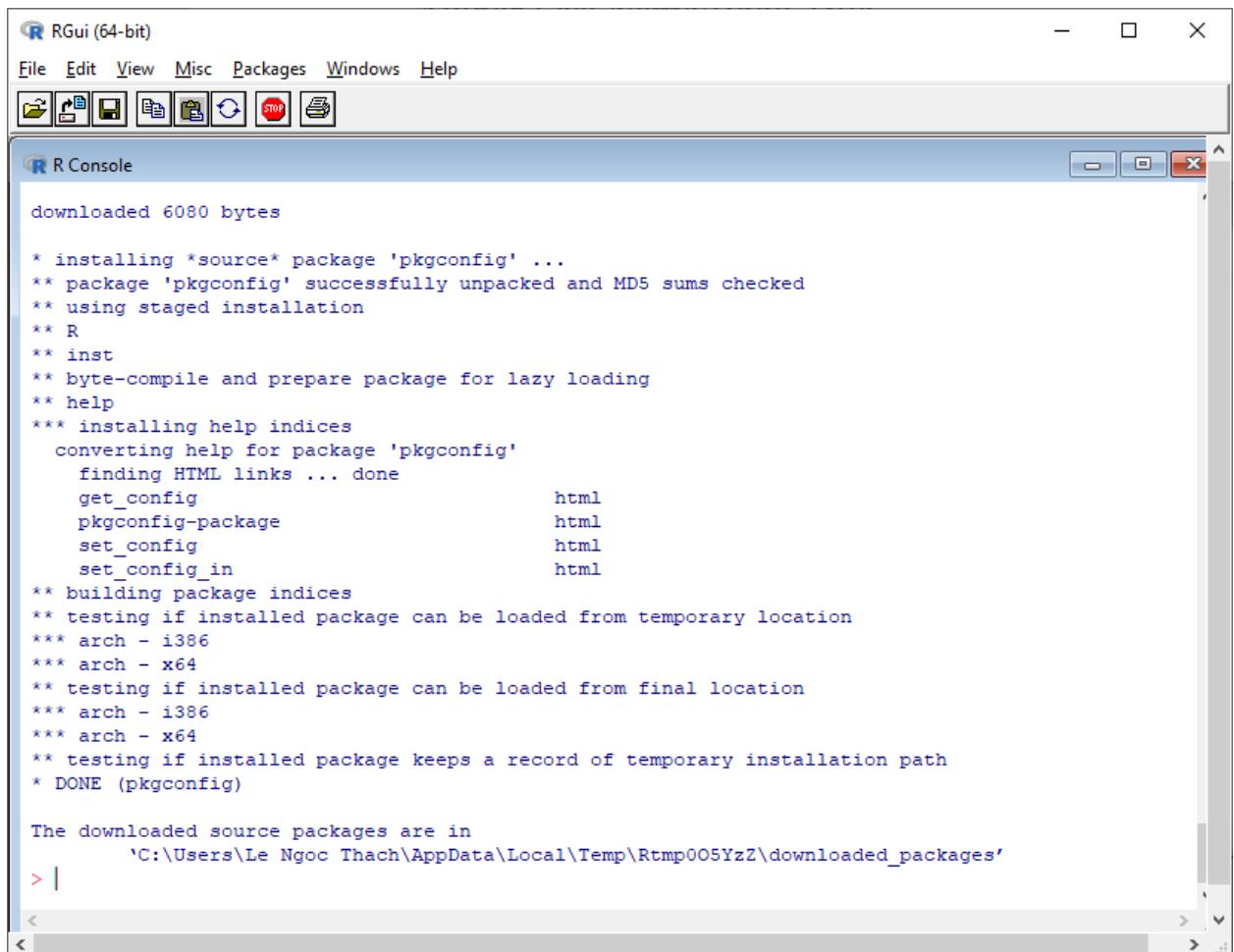
Ví dụ lệnh sau đây cài đặt package “tables”.

Chạm tới AI trong 10 ngày



Các package trong R được lưu trữ trên rất nhiều server trên Internet nên một cửa sổ “Secure CRAN mirrors” được hiển thị để bạn chọn địa chỉ để download. Bạn cứ chọn địa chỉ mặc định “0-Cloud [https] và nhấn OK là được. Hãy theo dõi cửa sổ “R Console” rất nhiều thông tin được hiển thị trong quá trình download và setup package.

Chạm tới AI trong 10 ngày



```
downloaded 6080 bytes

* installing *source* package 'pkgconfig' ...
** package 'pkgconfig' successfully unpacked and MD5 sums checked
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
  converting help for package 'pkgconfig'
    finding HTML links ... done
      get_config                      html
      pkgconfig-package                html
      set_config                       html
      set_config_in                   html
** building package indices
** testing if installed package can be loaded from temporary location
*** arch - i386
*** arch - x64
** testing if installed package can be loaded from final location
*** arch - i386
*** arch - x64
** testing if installed package keeps a record of temporary installation path
* DONE (pkgconfig)

The downloaded source packages are in
  'C:\Users\Le Ngoc Thach\AppData\Local\Temp\Rtmp005Yzz\downloaded_packages'
> |
```

Xem lại version của package đã cài bằng lệnh packageVersion:

```
> packageVersion('tables')
[1] '0.8.8'
```

Lệnh cài đặt các packages thì chỉ cần thực hiện một lần. Sau đó để sử dụng thì chỉ cần khai báo:

```
library(<tên thư viện>)
```

Trong tài liệu này để thuận tiện cho các bạn copy & paste code thì tôi sử dụng đoạn code sau để cài đặt thư viện nếu máy mình chưa có. Ví dụ cần dùng 2 thư viện dplyr và MASS thì sử dụng đoạn code sau ở đầu file mã nguồn R.

```
packages <- c('dplyr', 'MASS')
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

library(dplyr)
```

library(MASS)

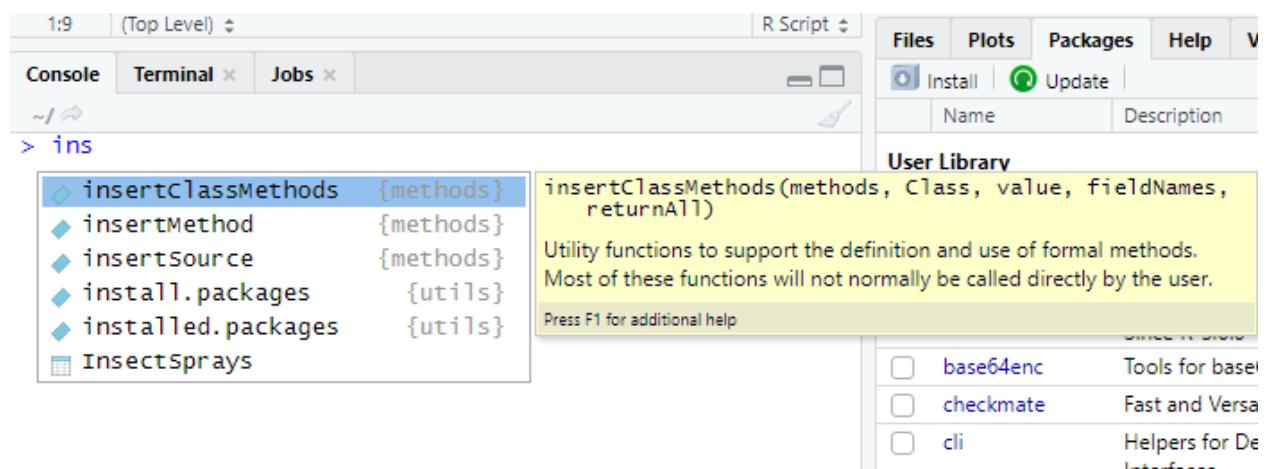
Ngôn ngữ thống kê R

Bạn có thể bắt đầu làm quen với ngôn ngữ R và thực hành với bộ phần mềm RStudio mà thôi sẽ giới thiệu dưới đây.

Sử dụng RStudio

Trong bài trước thì bạn đã làm quen với phần mềm R. Bạn có thể thực hiện lệnh trong cửa sổ R Console. Tuy nhiên bạn sẽ gặp khó khăn khi phải nhớ và gõ đầy đủ lệnh.

Phần mềm RStudio sẽ giúp các bạn giải quyết khó khăn trên và còn mang lại nhiều tiện ích khác nữa. Ví dụ khi bạn gõ vài chữ của lệnh như “ins” thì RStudio sẽ bật ra các lệnh bắt đầu bằng chữ “ins” để bạn nhìn thấy. Lúc này bạn có thể gõ theo hoặc dùng phím mũi tên **↑↓** để chọn lệnh rồi nhấn Enter. Ngoài ra một khung nền vàng nhạt hiển thị cú pháp và tài liệu hướng dẫn của lệnh.



Thật là tiện lợi phải không?

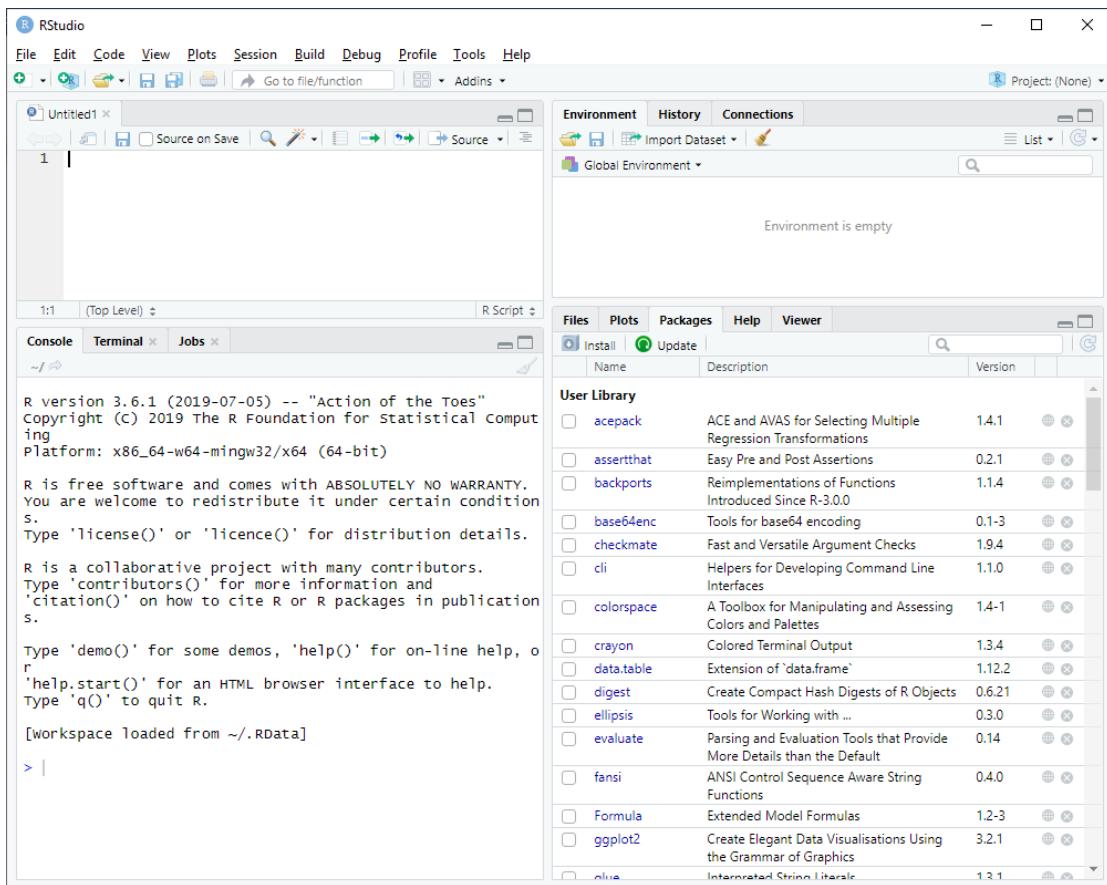
Tải mềm RStudio tại:

<https://www.rstudio.com/>

Bạn chỉ cần cài bản “RStudio Desktop” FREE là đủ để thực thành khi đọc tài liệu này. Tại thời điểm tài liệu này được viết thì tôi sử dụng RStudio Desktop Free phiên bản 1.2.5.

Giao diện RStudio tổng quan như sau:

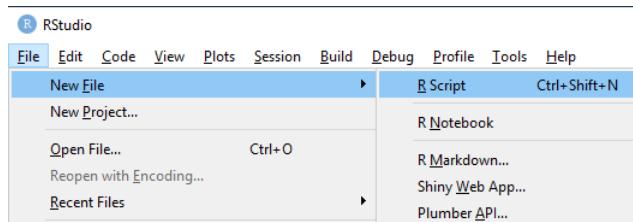
Chạm tới AI trong 10 ngày



Trong RStudio, cửa sổ ở góc trái bên dưới có tab “Console” có chức năng tương tự như cửa sổ R Console trong phần mềm R. Trong tab “Console” có dấu mũi tên để bạn gõ lệnh như tôi đã đề cập ở trên.

Một vài chức năng quan trọng giúp bạn sử dụng R hiệu quả hơn như sau:

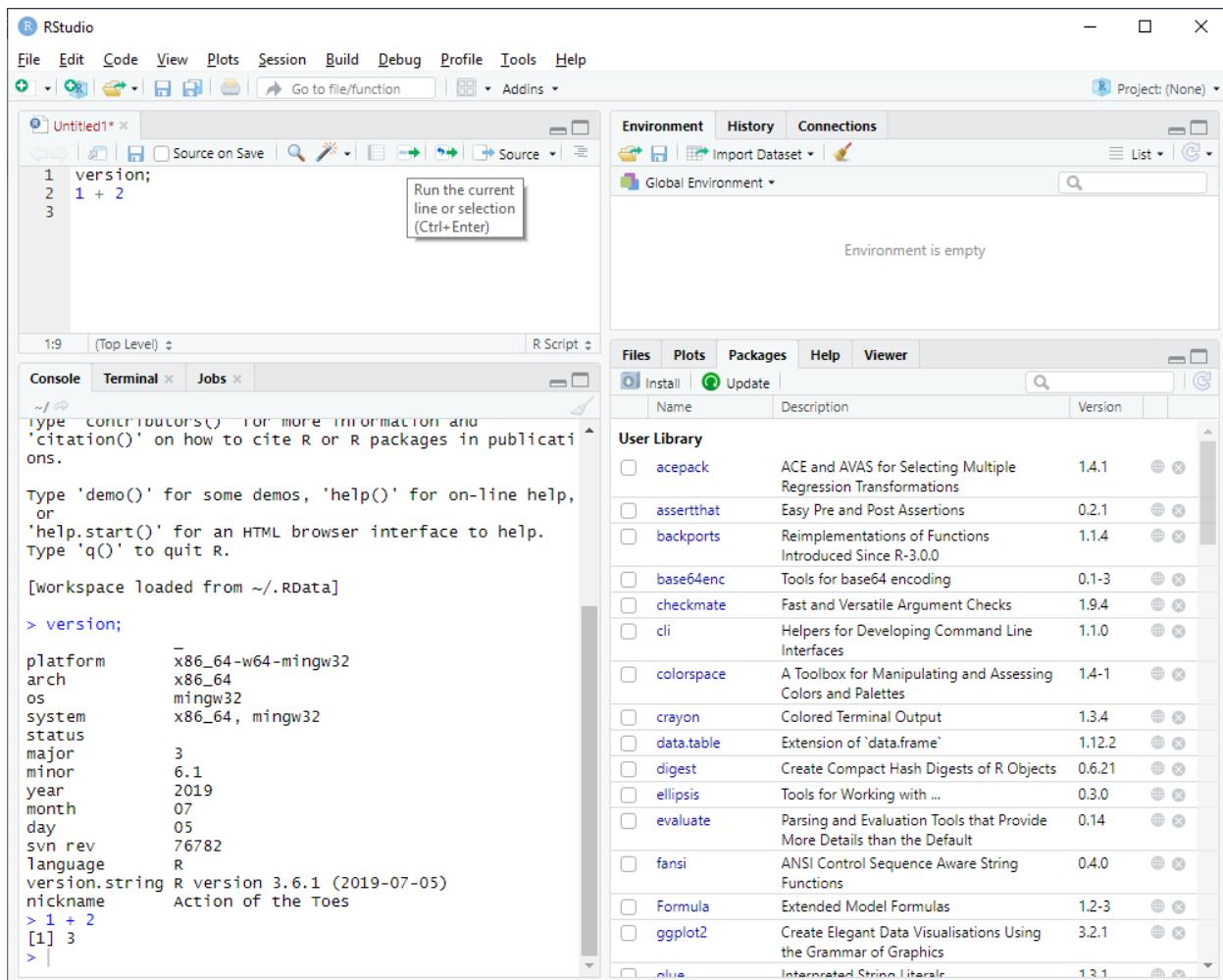
- Mở file mã nguồn R bằng cách vào menu File > New File > R Script hoặc dùng tổ hợp phím Ctrl + Shift + N



Cửa sổ Untitled<n> (<n> là số thứ tự) cho phép bạn soạn mã nguồn R và lưu lại thành file để dùng lại sau này như chỉnh sửa, chạy lại các lệnh trong file có sẵn.

Khi bạn gõ xong một lệnh trong cửa sổ viết R Script, bạn có thể thực thi dòng lệnh ngay tại vị trí cursor đang đứng bằng cách nhấn tổ hợp phím Ctrl + Enter hoặc bấm vào biểu tượng 

Chạm tới AI trong 10 ngày

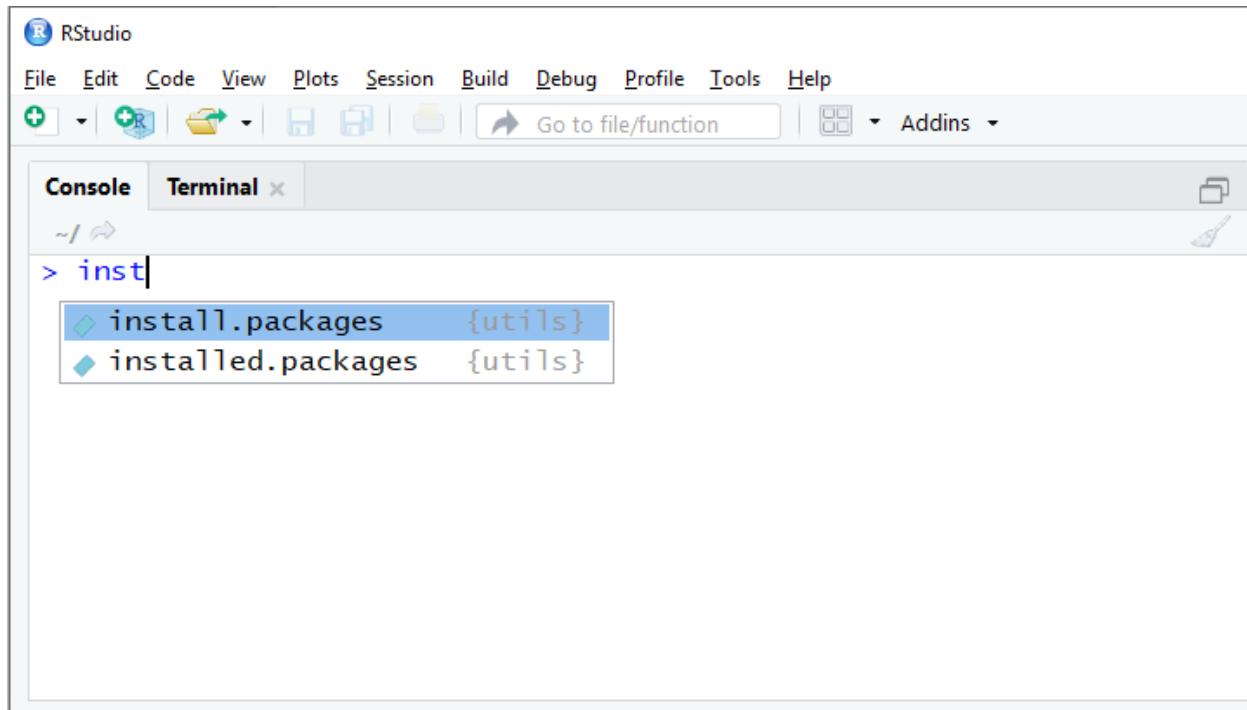


Để dọn dẹp cửa sổ console thì dùng tổ hợp phím Ctrl + L (L là ký tự thứ hai trong chữ clear).

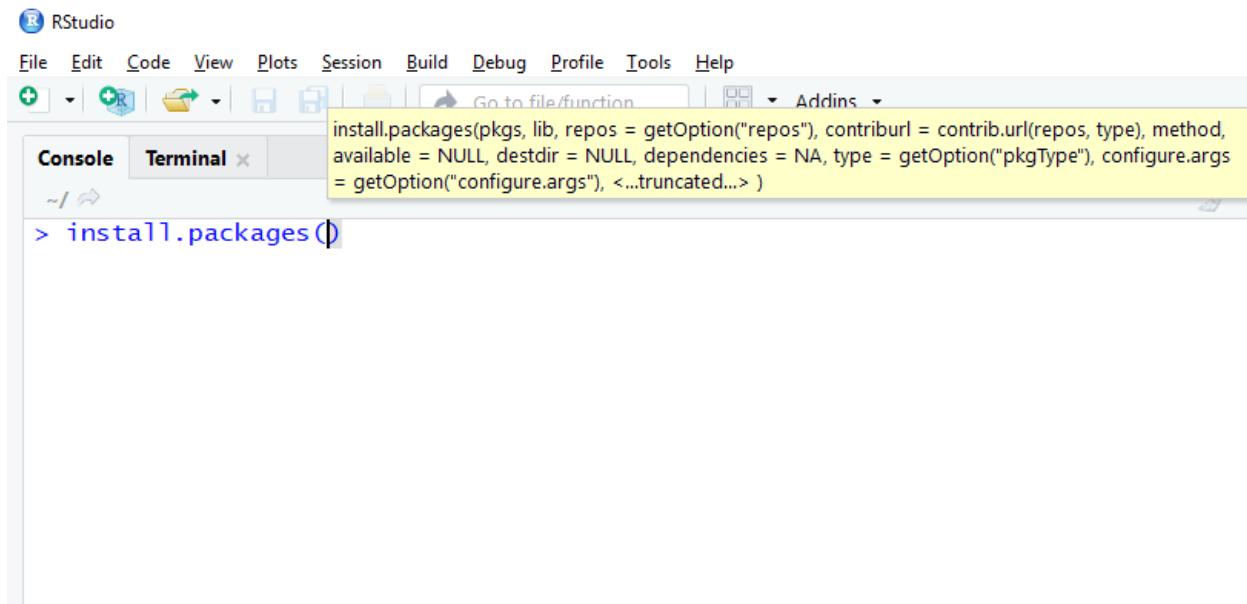
Trong RStudio, bạn chỉ cần gõ một phần của lệnh – ví dụ gõ inst – thì RStudio sẽ tự gợi ý các lệnh bắt đầu bằng phân lệnh đã gõ. Trường hợp RStudio chưa bật ra danh sách lệnh thì hãy nhấn phím Ctrl + Space.

Trong hình bên dưới là lệnh install.packages được hiển thị đầu tiên. Bạn cần nhấn phím Enter để chọn nó.

Chạm tới AI trong 10 ngày



Sau khi nhấn Enter để chọn lệnh thì RStudio hiển thị sẵn cặp dấu ngoặc và con nháy nằm sẵn ở giữa dấu ngoặc để bạn gõ tiếp các tham số. Ngoài ra, một hướng dẫn tổng quát của lệnh cũng được hiển thị (phần chữ nhạt nền vàng nhạt) để bạn biết các tham số của lệnh (hoặc của hàm).



Ví dụ lệnh sau đây cài thư viện xlsx để đọc dữ liệu từ file Microsoft Excel.

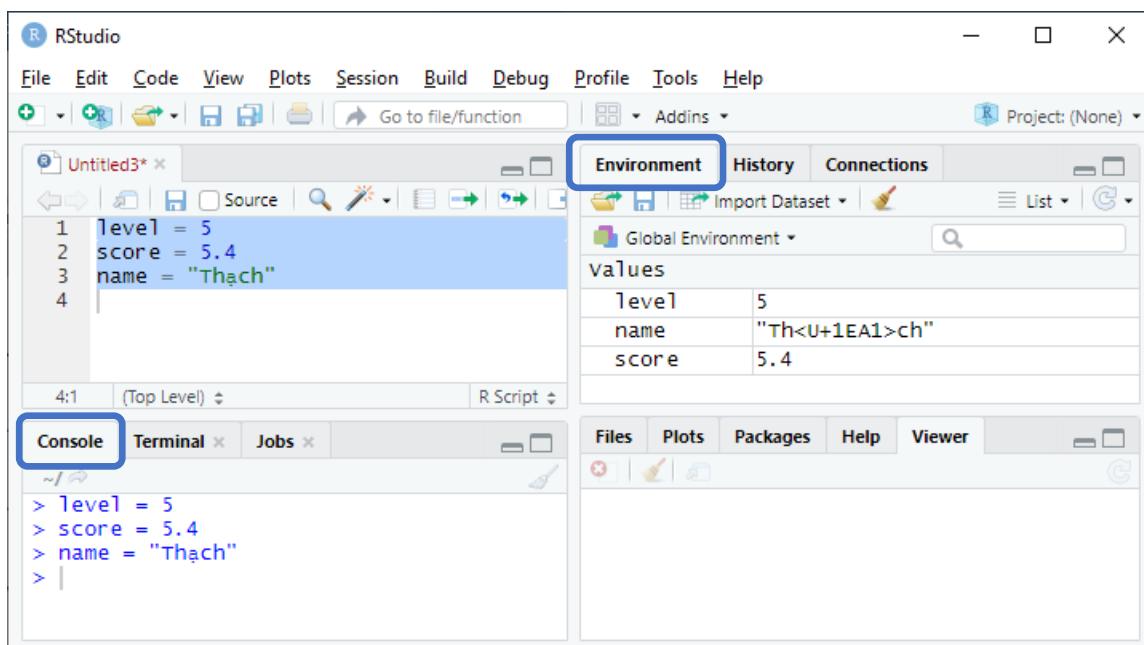
```
> install.packages('xlsx')
```

Thực hành phép gán

Khởi động phần mềm RStudio, mở file mới bằng cách nhấn tổ hợp phím Ctrl + Shift + N. Sau đó gõ 3 lệnh sau:

```
level = 5  
score = 5.4  
name = 'Thạch'
```

Thực thi 3 dòng lệnh bằng cách bôi cả 3 dòng rồi nhấn phím Ctrl+Enter. Quan sát giá trị các biến trong thẻ “Environment” và quan sát các lệnh được thực thi trong cửa sổ “Console” bên góc trái dưới.



Chạm tới AI trong 10 ngày

The screenshot shows the RStudio interface. In the top-left pane, there is an R script titled "Untitled1" with the following code:

```
1 level = 5
2 score = 5.4
3 name = "Thach"
4 birthday = date()
5 birthday
6 sex = TRUE
7 sex
8
```

In the top-right pane, under the "Environment" tab, the variables are listed with their values:

| values | |
|----------|----------------------------|
| birthday | "Tue Sep 24 21:37:19 2019" |
| level | 5 |
| name | "Thach" |
| score | 5.4 |
| sex | TRUE |

In the bottom-left pane, the "Console" tab shows the execution of the script:

```
> level = 5
> score = 5.4
> name = "Thach"
> birthday = date()
> birthday
[1] "Tue Sep 24 21:37:19 2019"
> sex = TRUE
> sex
[1] TRUE
>
```

Dòng lệnh 1 gán giá trị 5 vào biến level.

Dòng lệnh 2 gán giá trị 5.4 vào biến score.

Dòng lệnh 3 gán chuỗi kí tự (string) “Thạch” vào biến name. Bạn có thể dùng cặp dấu nháy đơn hoặc đôi để bao đóng chuỗi.

Dòng lệnh 4 thực hiện lệnh date() để lấy thời gian hiện tại của máy tính gán vào biến birthday. Dòng số 5 chỉ có biến birthday có nghĩa là xem giá trị của biến. Kết quả birthday được hiển thị ra màn hình với định dạng: Thứ Tháng Ngày Giờ:Phút:Giây Năm.

Dòng lệnh 6 gán cho biến sex là một giá logic (Đúng hoặc Sai). Giá trị TRUE (chú ý viết hoa) có nghĩa là Nam hay Nữ là do người viết lệnh quy ước.

Trong RStudio bạn có thể quan sát các biến đang tồn tại trong cửa sổ bên phải, tab Environment:

The screenshot shows the RStudio interface with the "Environment" tab selected. It displays the current variables and their values:

| values | |
|----------|----------------------------|
| birthday | "Mon Sep 23 22:21:33 2019" |
| level | 5 |
| name | "Thach" |
| score | 5.4 |

Tham khảo thêm:

<https://www.w3schools.in/r/data-types/>

Sử dụng chú thích

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

```
# Khai báo biến tuổi  
age = 40  
  
# Gán biến giới tính. 1 = Male; 0 = Female  
sex = 1  
name = "Lê Ngọc Thạch"
```

Gọi hàm

Như đã giới thiệu ở trên, R cung cấp rất nhiều thư viện (library) hay còn gọi là package do cách chuyên gia về thống kê, chuyên gia lập trình tạo nên và chia sẻ hầu như là miễn phí cho cộng đồng. Trong mỗi thư viện có rất nhiều hàm (function), đôi khi gọi là phương thức (method) giúp chúng ta thực hiện các phân tích cơ bản.

Bây giờ chúng ta thử gọi hàm để trải nghiệm các khái niệm cơ bản về thống kê đã được nhắc đến trong bài 1.

Sử dụng hàm c() để thiết lập danh sách điểm của 10 môn học và gán kết quả cho biến scores:

```
scores = c(6, 8, 9, 4, 5, 7, 8, 6, 5, 7)
```

- Gọi hàm mean như sau:

```
mean(scores)
```

sẽ cho kết quả: 6.4

- Gọi hàm median:

```
median(scores)
```

sẽ cho kết quả: 6.5

Vì sao median(scores) lại cho ra 6.5 nhỉ?

Bạn còn nhớ khái niệm **median** là trung vị được xác định như sau chứ?

Trung vị là 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.

Vì thế, nếu bạn gọi hàm sort(scores) để xếp thứ tự danh sách điểm:

```
sort(scores)
```

Thì kết quả như sau:

```
4 5 5 6 6 7 7 7 8 9
```

Vì dãy số này có 10 phần tử nên trung vị được tính là trung bình của 2 số ở chính giữa: $(6 + 6) / 2 = 6.5$

- Làm sao tính Mode cho biến scores nhỉ? Nhắc là Mode là phần tử được lặp lại nhiều nhất trong dãy số. Sau khi dùng hàm sort(scores) thì bạn dễ dàng thấy điểm số 7 được lặp lại nhiều nhất (3 lần).

Trong R không có sẵn hàm Mode, nên chúng ta phải tự tính. Cái tính hơi lòng vòng một chút như sau. Phần này tôi sẽ đưa vào mục nâng cao để các bạn yêu thích lập trình giải trí sau.

- Tính phương sai như thế nào nhỉ? Cần nhắc lại một chút:

Phương sai (variance) dùng để đo độ lệch, hoặc là mức độ cách biệt của các giá trị so với giá trị mean

Trong phần khái niệm đã có nêu cách tính phương sai bằng tay. Trong R thì dùng hàm var(scores) để tính phương sai của danh sách điểm:

```
var(scores)
```

Kết quả là: 2.266667

- Độ lệch chuẩn (standard) được tính bằng căn bậc hai của phương sai. Vì vậy bạn có thể tự tính bằng cách gọi hàm sqrt (square root) như sau:

```
sqrt(var(scores))
```

Kết quả là: 1.505545

- Bạn cũng có thể gọi hàm sd(scores) để tính ra kết quả tương tự:

```
sd(scores)
```

Kết quả là: 1.505545

Như vậy bạn đã làm quen và có thể tự gọi một số hàm có sẵn trong R để cảm nhận được các khái niệm thống kê cơ bản trên một dãy số điểm đơn giản. Đây là những bước khởi động khá tốt để bạn quen với việc gõ hàm vào trong R để thực thi.

Cài đặt thư viện

Chạm tới AI trong 10 ngày

Bài tập thực hành

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

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

Code R:

```
fullName = 'Lê Ngọc Thạch'  
height = 165  
weight = 72.5  
sex = TRUE  
birthday = as.Date('30/6/2019', format = '%d/%m/%Y')  
favorNumbers = c(1, 2, 5, 10, 20, 50)  
favorSports = c('Bóng bàn', 'Bóng đá ', 'Quần vợt ')
```

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

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

```
typeof(<tên biến>)
```

Ví dụ:

```
typeof(fullName)
```

Online Quizzes

Phần này không có trong kế hoạch. Thực tế thì tùy thuộc vào các nhà tài trợ mà tôi có duy trì trang web này hay không? Nếu bạn thấy trang web có tồn tại thì coi như may mắn nhé. Hãy sử dụng tài khoản được cấp truy cập vào website:

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

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

| Title | Time Limit | Due Date/Time |
|---------------------------|------------|---------------|
| Ngày 1 - Ôn tập bài 3 - R | n/a | n/a |

Bài 4: Ngôn ngữ Python và phần mềm Anaconda

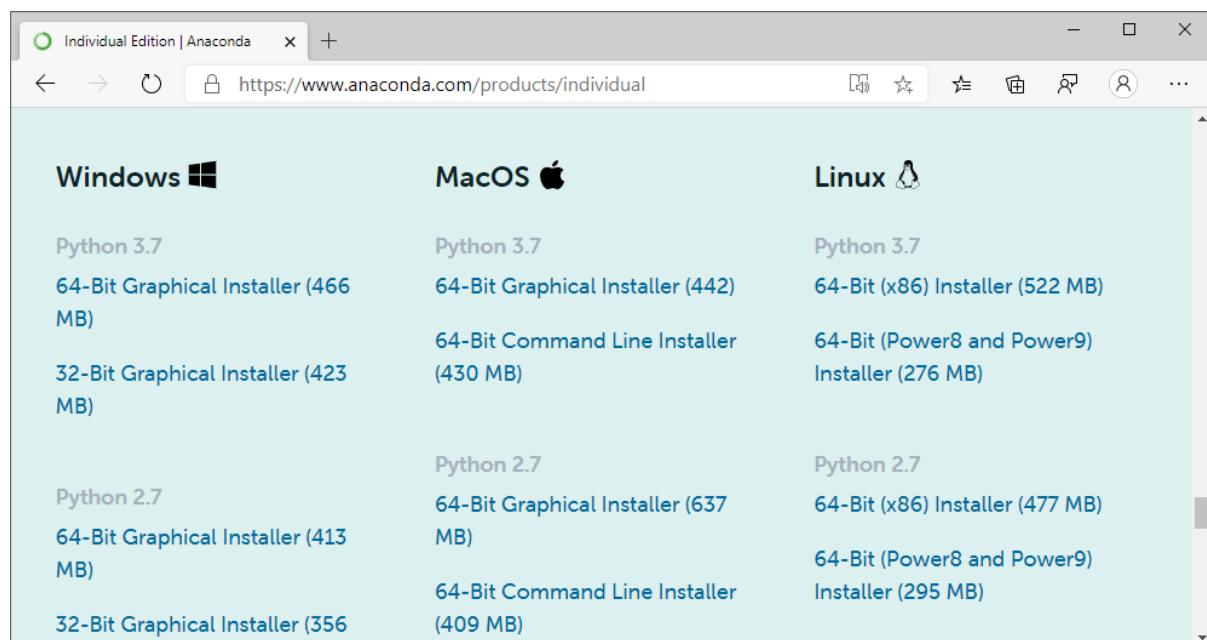
Anaconda

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:

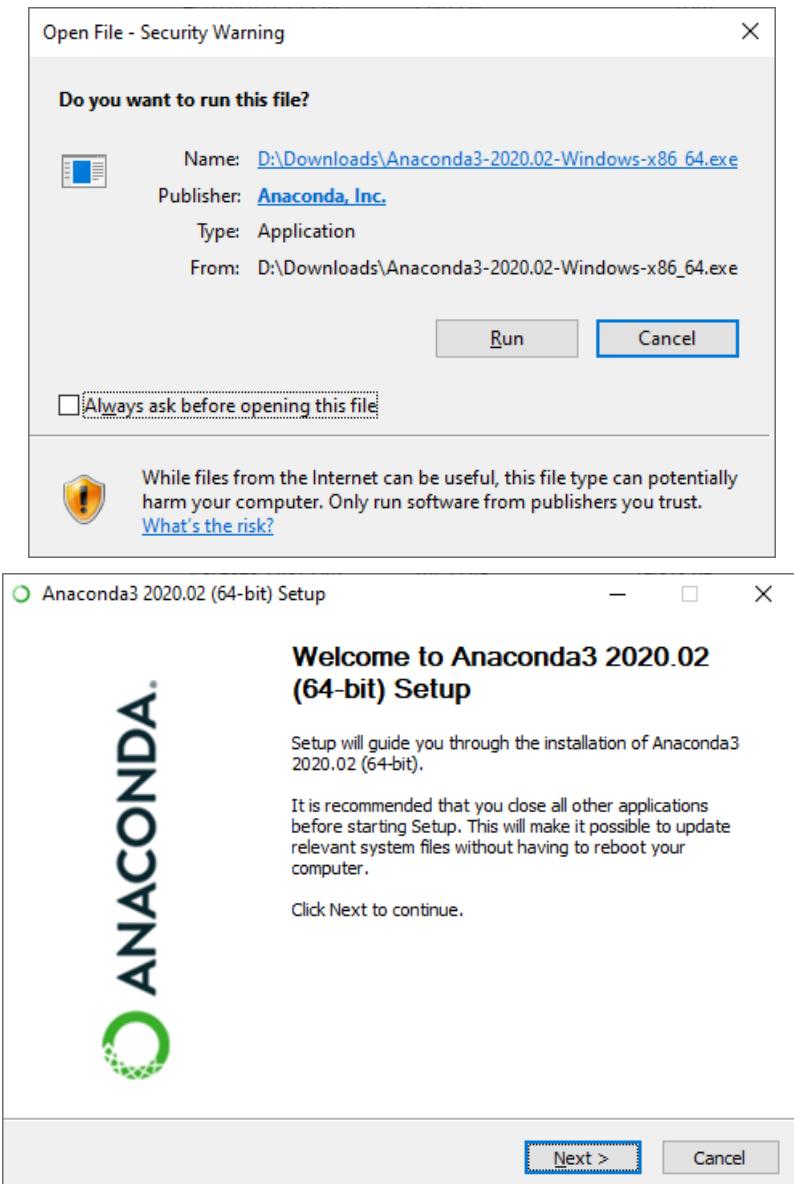


Sau đó bấm mục của gói cài đặt tùy theo máy của bạn. Tại thời điểm viết phần này thì Anconda cung cấp 2 phiên bản phổ biến cho Python 3.7 và Python 2.7. Có nhiều sự khác biệt lớn giữa hai phiên bản Python 3 (gọi chung là 3.x) và Python 2 (gọi chung là 2.x); cũng có nhiều lý do vì sao mọi người đang dùng 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).

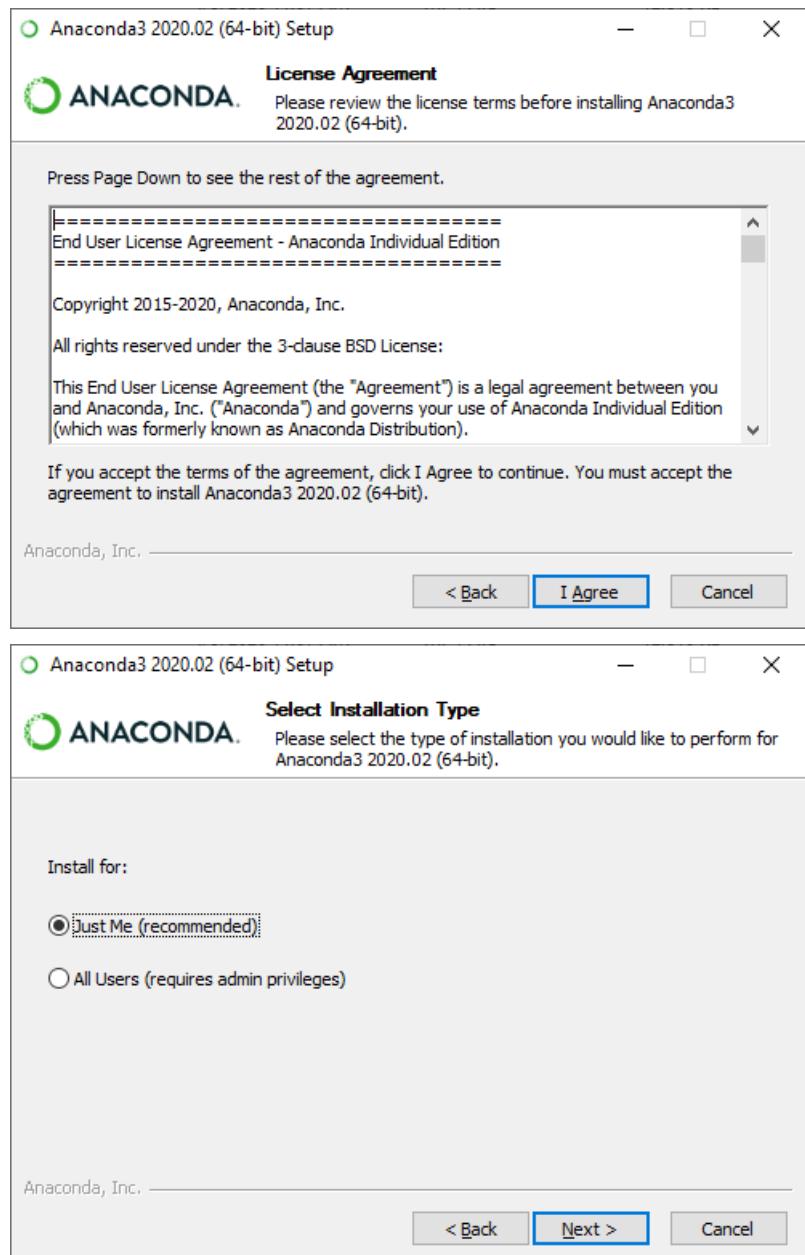
Tôi sẽ dùng phiên bản 3.7 và file tải về là “[64-Bit Graphical Installer \(466 MB\)](#)” do máy tôi dùng Windows 64 bit. Nếu máy bạn đang dùng Windows 32 bit thì tải link “[32-Bit Graphical Installer \(423 MB\)](#)”. Tương tự nếu bạn dùng MacOS hoặc Linux thì bấm vào link tương ứng phía trên mà hình.

Chạm tới AI trong 10 ngày

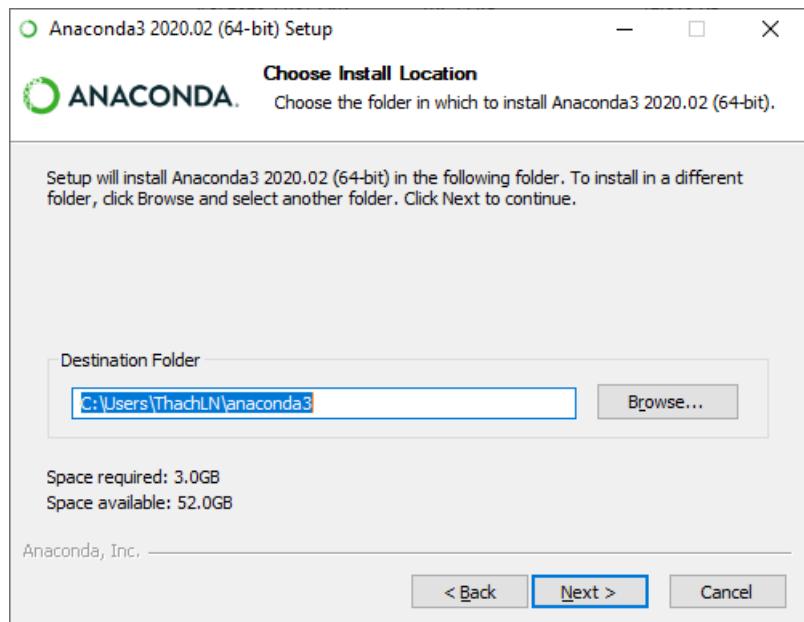
Quá trình cài đặt khá đơn giản. Cơ bản là cứ bấm “Next” và “Agree” rồi làm theo hướng dẫn.



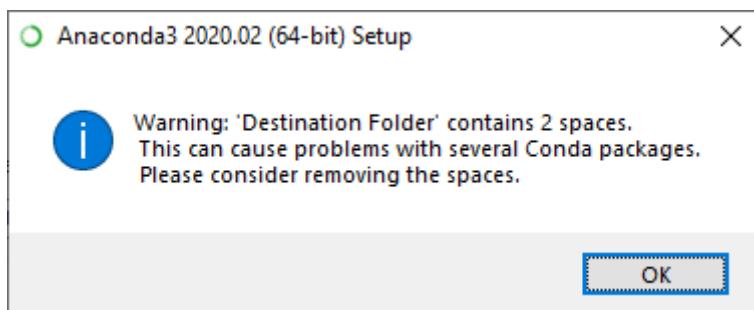
Chạm tới AI trong 10 ngày



Chạm tới AI trong 10 ngày

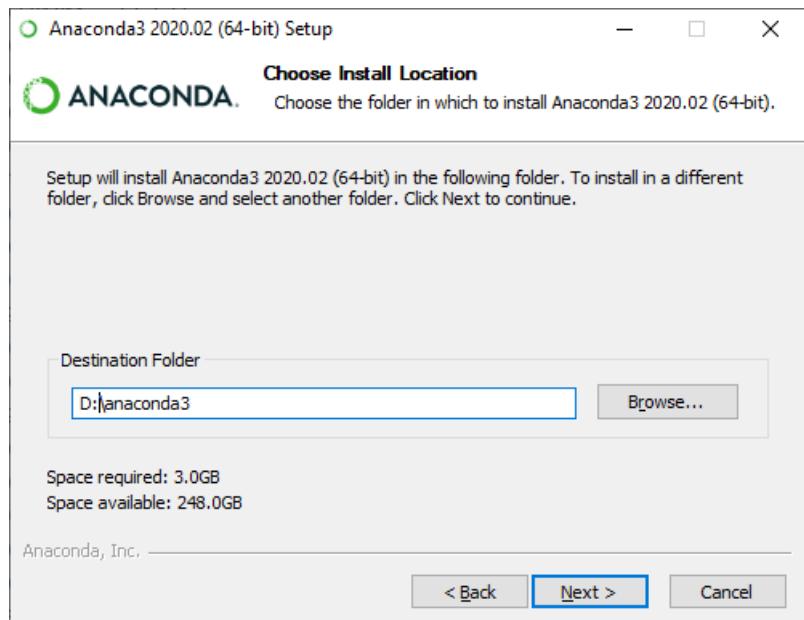


Tới bước chọn thư mục “Destination Folder” thì nếu bạn bấm “Next” mà tên thư mục của bạn có khoảng trắng (ví dụ tôi dùng tên đầy đủ để đăng nhập vào máy nên có khoảng trắng) thì sẽ bị cảnh báo như bên dưới:

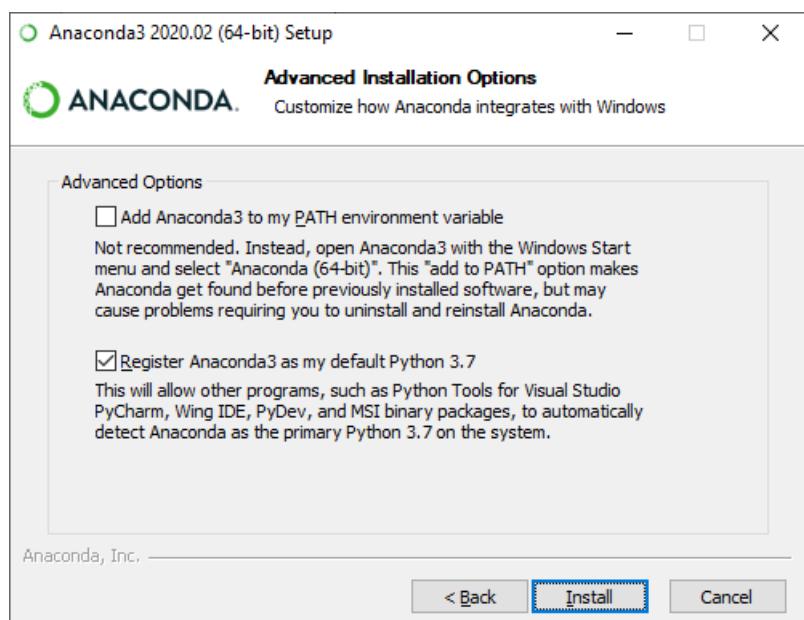


Lúc này bạn nên bấm OK rồi bấm tiếp “Back” trên màn hình tiếp theo để quay lại bước chọn “Destination Folder”. Ví dụ tôi chọn lại thư mục “D:\anaconda3” không có khoảng trắng và dễ quản lý.

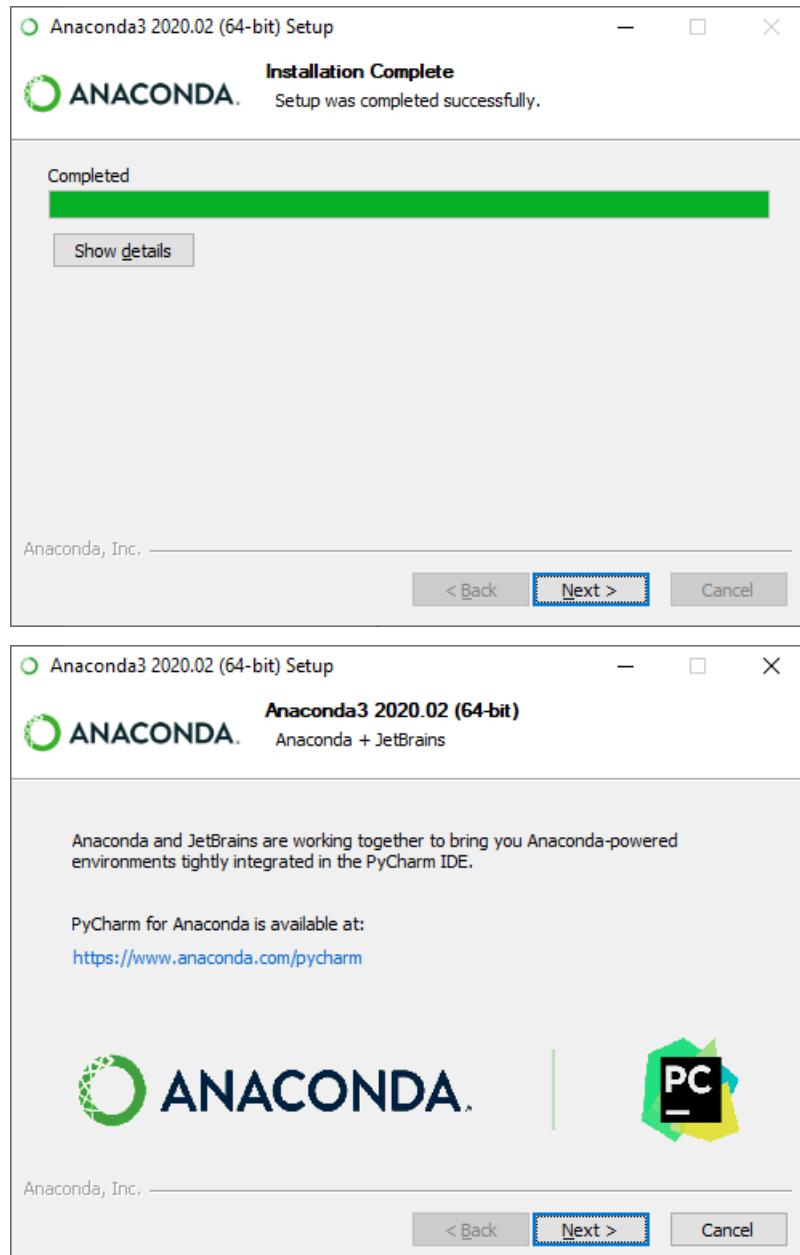
Chạm tới AI trong 10 ngày



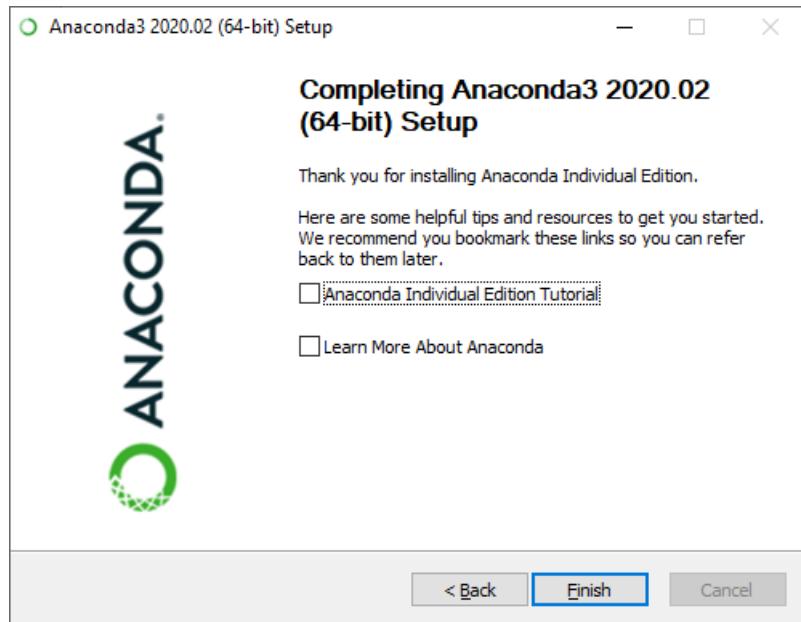
Rồi cứ thế bấm “Next”, “Install” và “Next” cho đến khi “Finish” là xong.



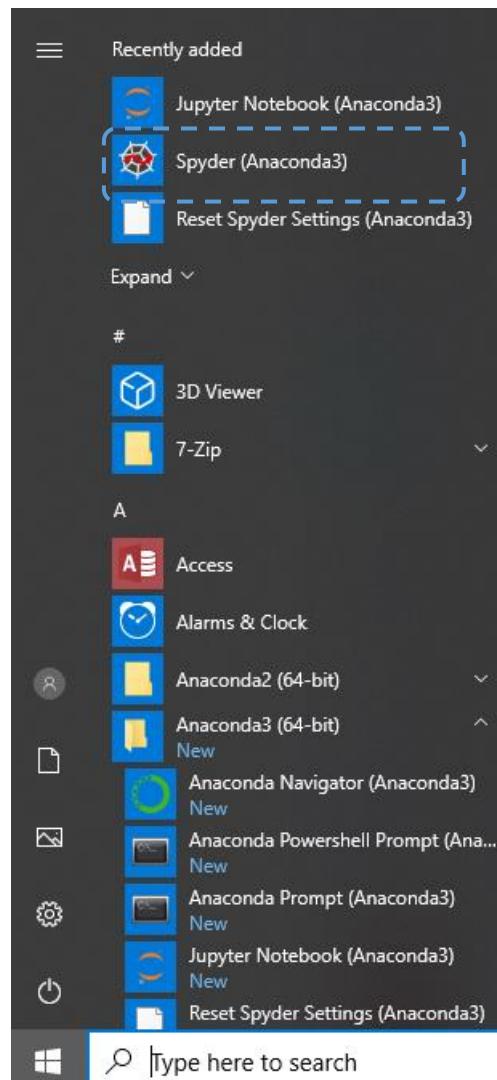
Chạm tới AI trong 10 ngày



Chạm tới AI trong 10 ngày



Sau khi cài xong, vào nút 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 chương trình Spyder (Anaconda 3)



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.

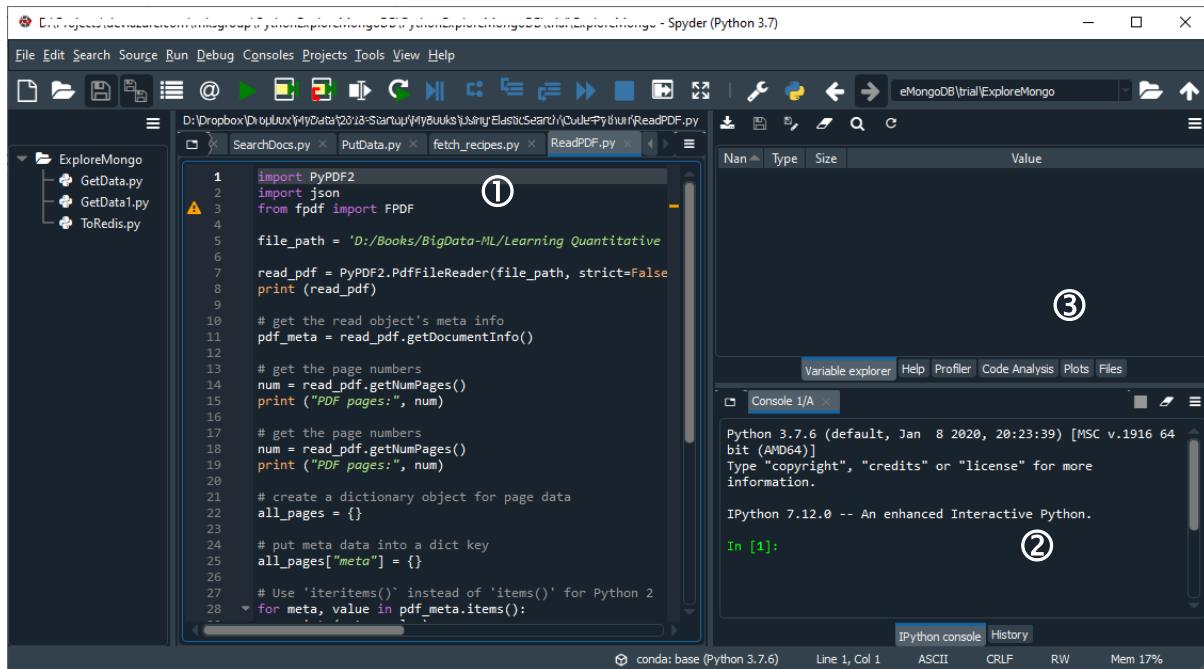
Ngôn ngữ lập trình Python

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:

Chạm tới AI trong 10 ngày



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.

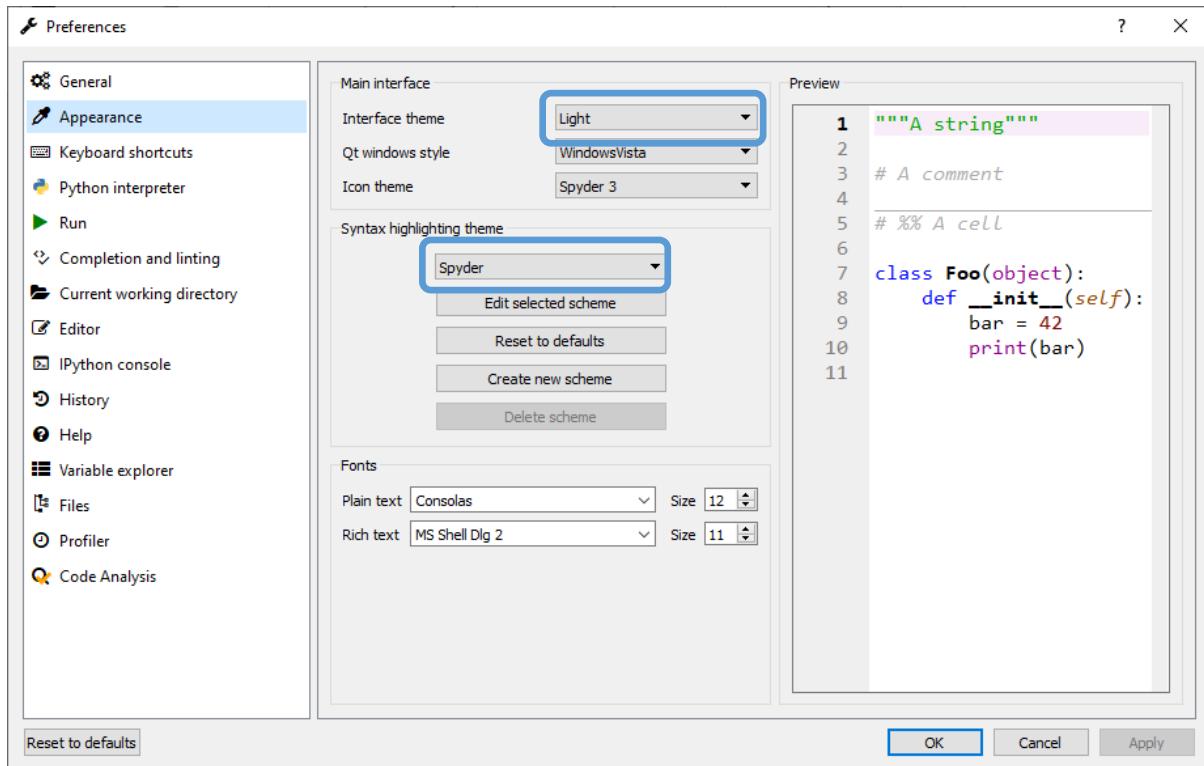
Đổi theme

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**

Chạm tới AI trong 10 ngày



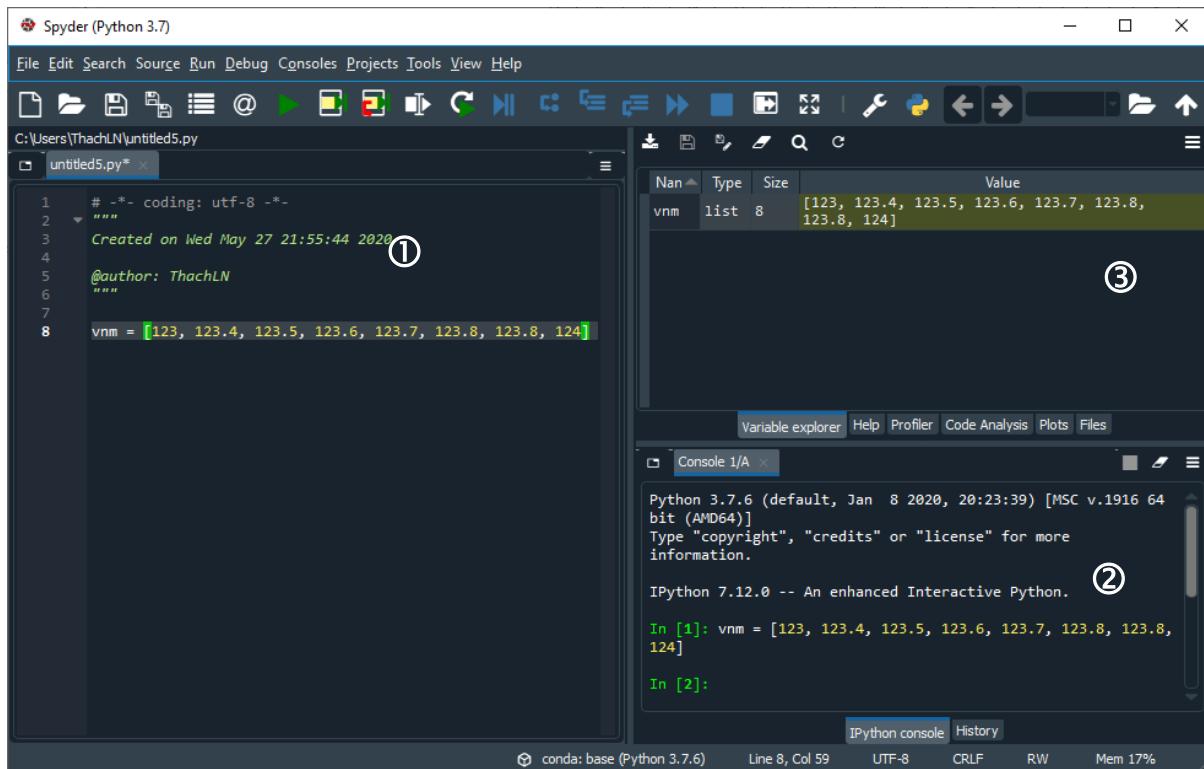
Thực thi lệnh

Chọn dòng lệnh cần thực thi, nhấn phím F9. Spyder phiên bản cũ hơn có thể nhấn phím Ctrl + Enter.

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]
```

Chạm tới AI trong 10 ngày



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

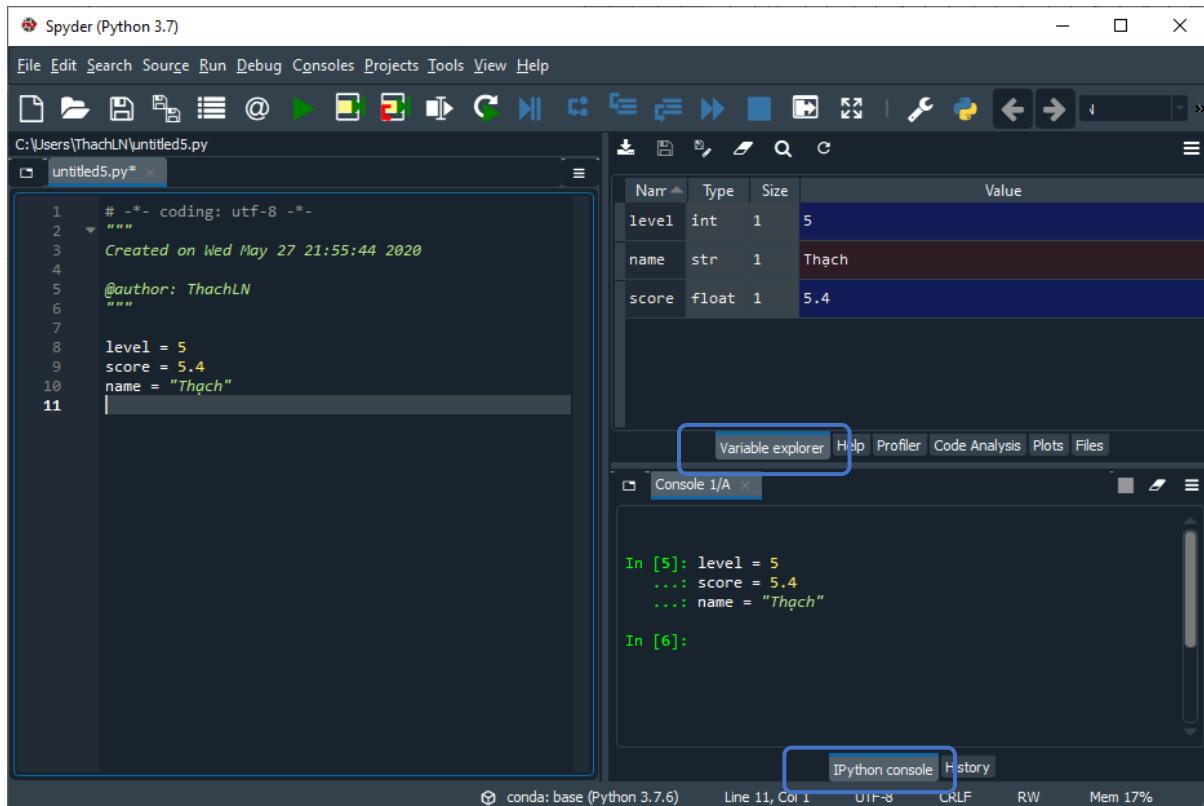
Hãy khởi động chương trình Spyder, mở file mới bằng cách nhấn Ctrl + N. Sau đó gõ 3 lệnh sau:

```
level = 5
```

Chạm tới AI trong 10 ngày

```
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.



Cài đặt các gói phần mềm

Tương tự như R, Python cũng cung cấp rất nhiều gói thư viện.

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

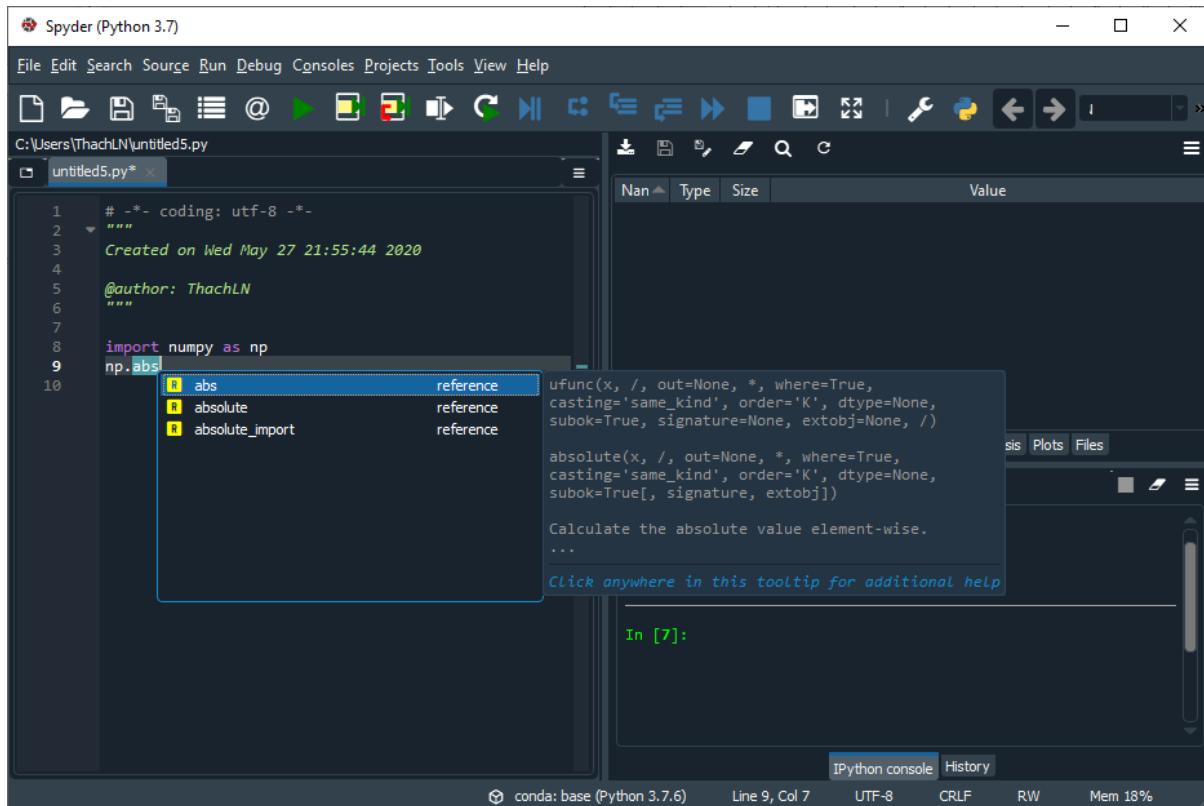
Ví dụ để sử dụng thư viện **numpy** thì sử dụng lệnh

```
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.

Chạm tới AI trong 10 ngày



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]
```

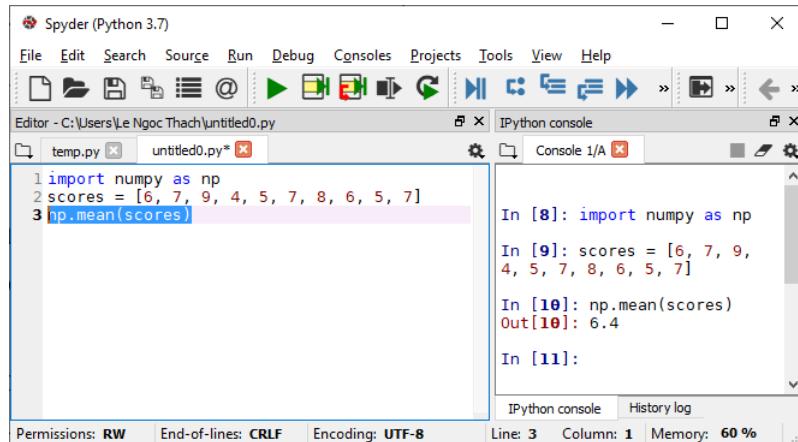
- Gọi hàm mean của thư viện numpy thông qua kí hiệu np:

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

sẽ cho kết quả: 6.4

Chạm tới AI trong 10 ngày

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. 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 phân pháp 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 các hàm np.std(x):

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

sẽ cho kết quả: 1.505545305418162

Bài tập thực hành

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 sau để biết thêm kiểu dữ liệu của biến:

```
type(<tên biến>)
```

Chạm tới AI trong 10 ngày

Ví dụ:

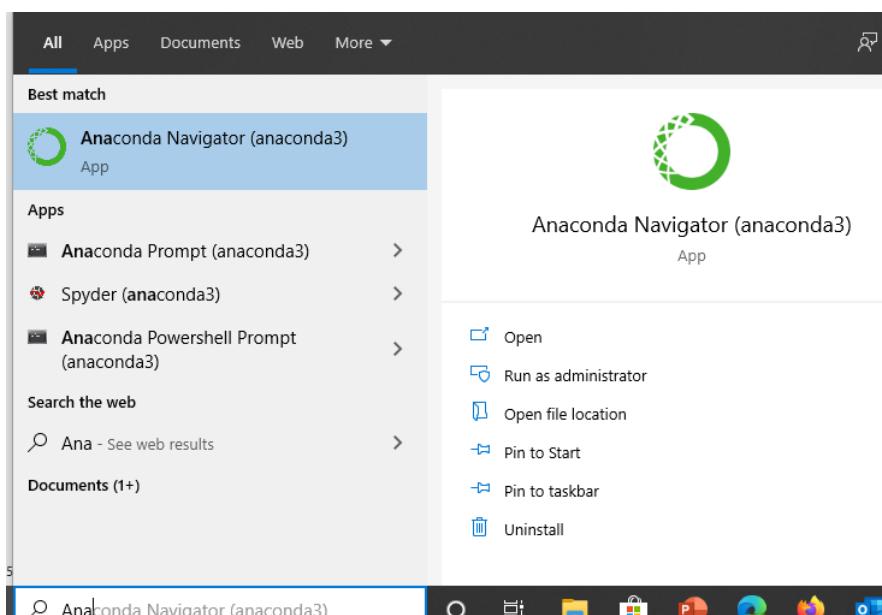
```
type(fullName)
```

Cho kết quả là: str

str có nghĩa là String (chuỗi)

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).



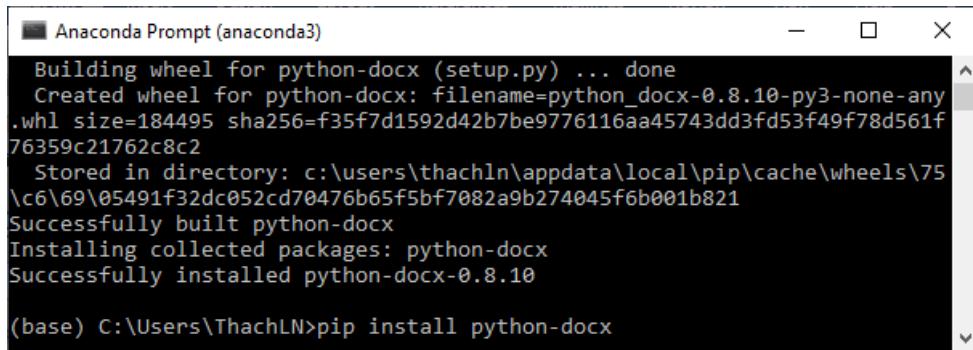
Trong cửa sổ Anaconda Prompt gõ lệnh:

```
pip install <tên thư viện>
```

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

```
pip install python-docx
```

Chạm tới AI trong 10 ngày

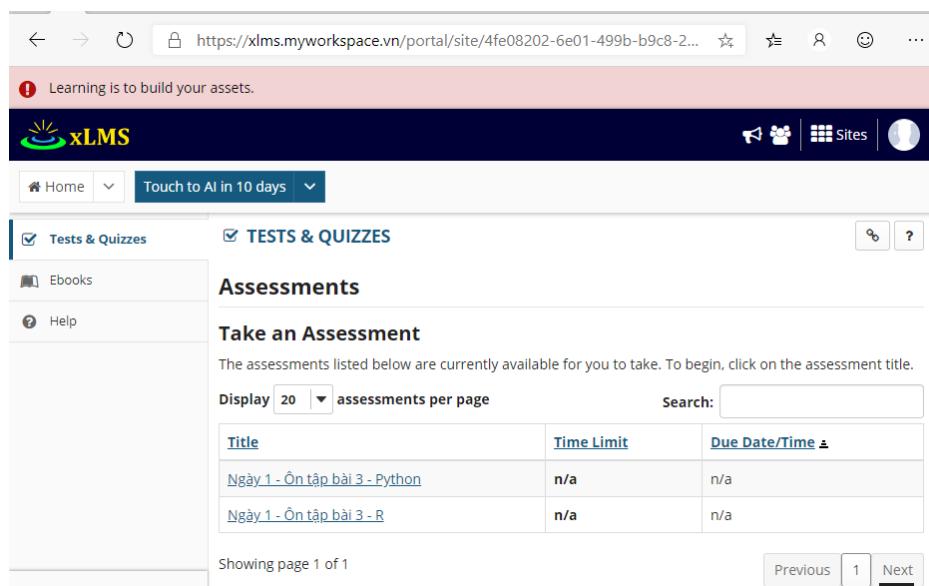


```
Anaconda Prompt (anaconda3)
Building wheel for python-docx (setup.py) ... done
Created wheel for python-docx: filename=python_docx-0.8.10-py3-none-any
.whl size=184495 sha256=f35f7d1592d42b7be9776116aa45743dd3fd53f49f78d561f
76359c21762c8c2
Stored in directory: c:\users\thachln\appdata\local\pip\cache\wheels\75
\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 the xlms.myworkspace.vn portal interface. At the top, there's a navigation bar with icons for back, forward, search, and user profile. Below it is a banner with the text "Learning is to build your assets." and the xlms logo. The main content area has a dark header with "Tests & Quizzes" and "TESTS & QUIZZES" checked. On the left, there's a sidebar with "Home", "Ebooks", and "Help" buttons. The main content area displays a table of assessments under the heading "Assessments". The table has columns for "Title", "Time Limit", and "Due Date/Time". There are two entries: "Ngày 1 - Ôn tập bài 3 - Python" and "Ngày 1 - Ôn tập bài 3 - R", both with "n/a" in the Time Limit column. At the bottom, there's a search bar and buttons for "Previous", "1", and "Next".

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

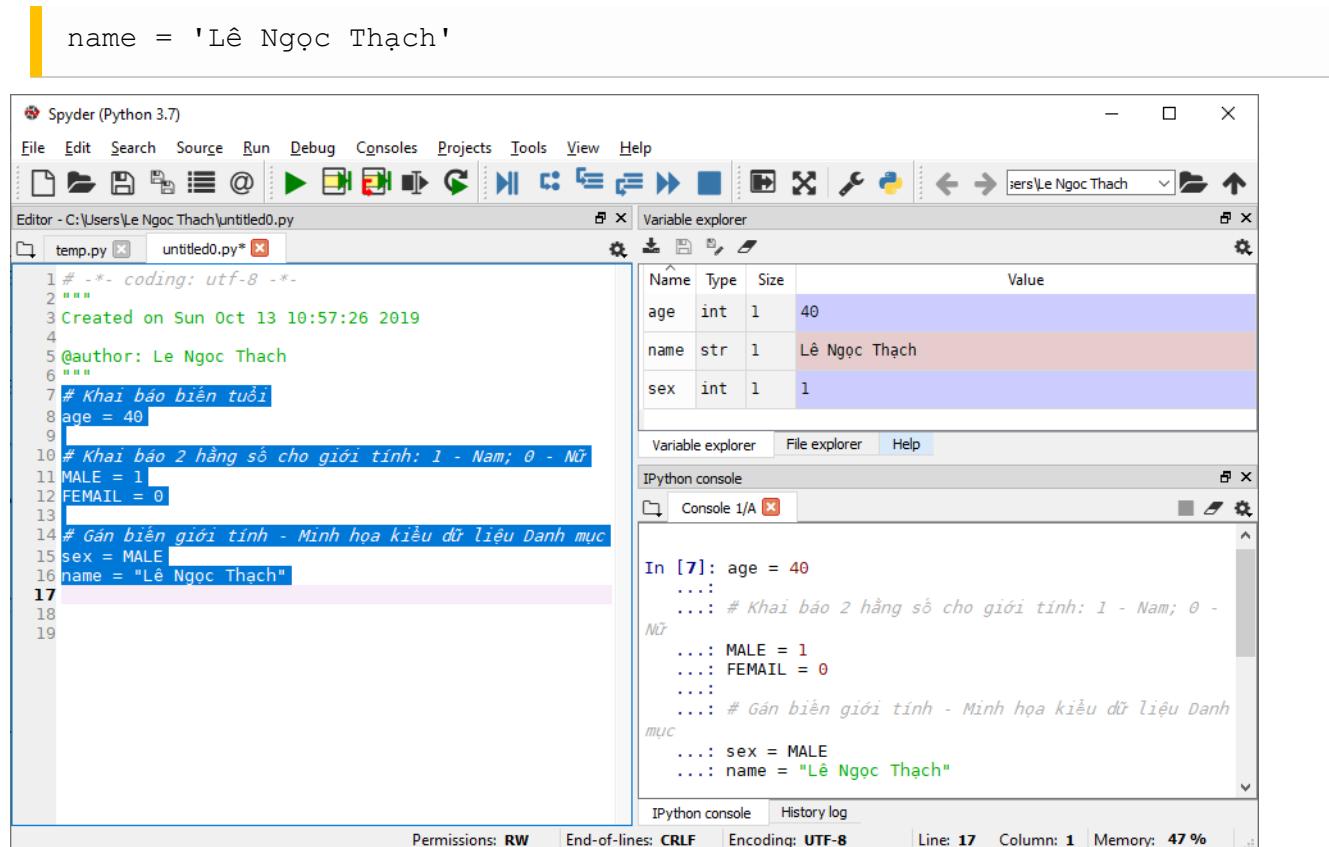
Sử dụng chú thích

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 nay 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
```

Chạm tới AI trong 10 ngày



The screenshot shows the Spyder IDE interface. In the top left, there's a yellow bar with the text "name = 'Lê Ngọc Thạch'". The main window is titled "Spyder (Python 3.7)". It has a menu bar with File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help. Below the menu is a toolbar with icons for file operations like Open, Save, Run, and Debug. The "Editor - C:\Users\Le Ngoc Thach\untitled0.py" tab is open, displaying Python code. The code includes comments about encoding, creation date, author, and gender variables. The "Variable explorer" panel on the right shows three variables: age (int, 40), name (str, Lê Ngọc Thạch), and sex (int, 1). Below the editor is an "IPython console" window showing the execution of the code. The status bar at the bottom indicates permissions (RW), end-of-lines (CRLF), encoding (UTF-8), line (17), column (1), and memory usage (47%).

Chú ý là tôi có tình cờ lúc dùng cặp nháy đôi, có lúc dùng cặp nháy đơn để bao đóng chuỗi để giúp bạn nhò là dùng cái nào cũng được, ý nghĩa là như nhau trong cả R và 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>".

Trong cửa sổ Anaconda Powershell Prompt thực hiện lệnh:

```
conda install spyder=4.1.3
```

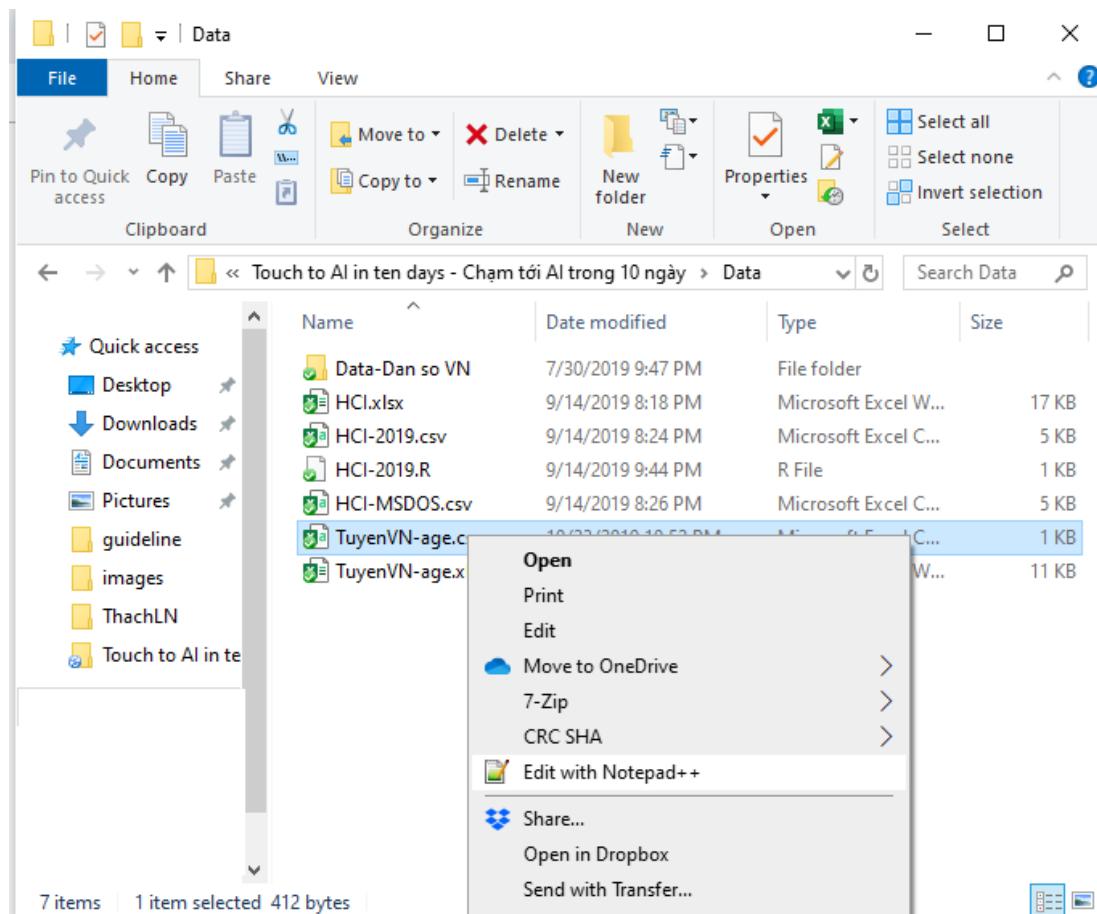
Nếu bạn kiểm tra phiên bản đã phát hành của Spyder lớn hơn 4.1.3 thì sửa lại lệnh trên cho phù hợp.

Bài 5: 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.

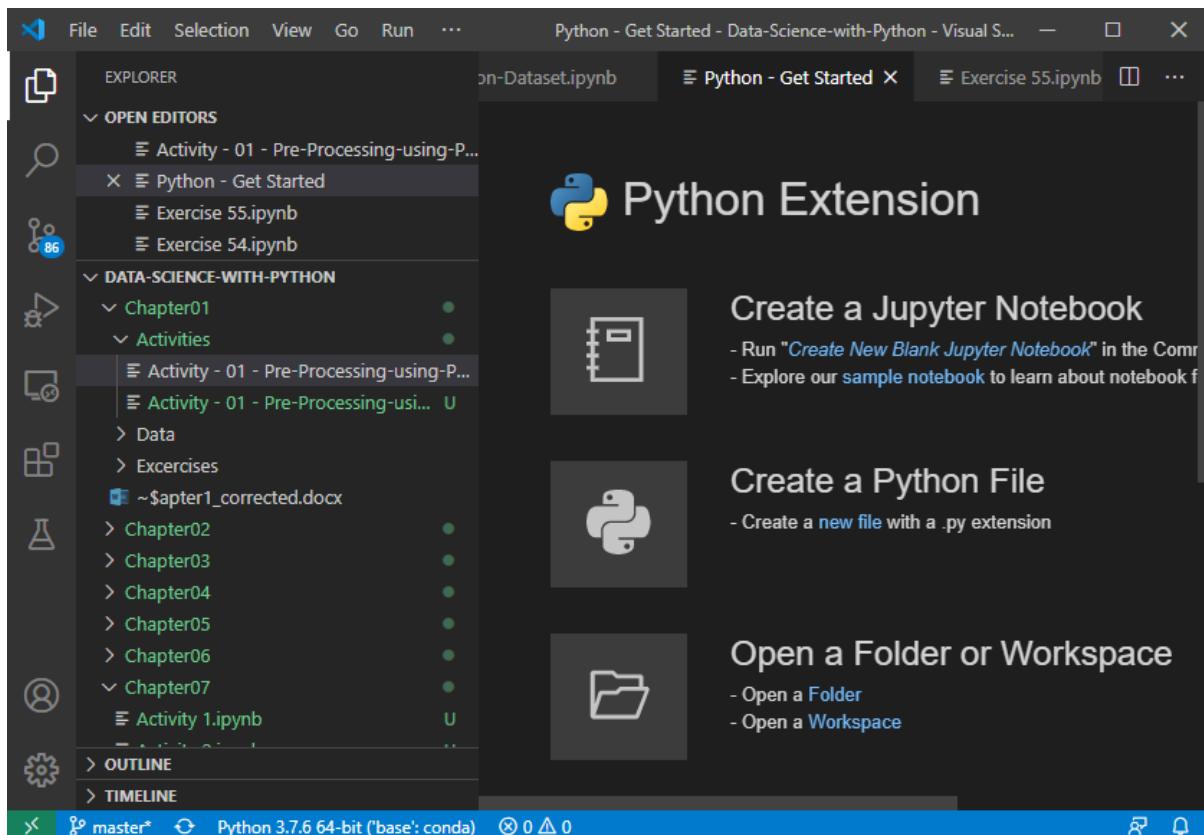


Phần mềm Visual Studio Code (VSC)

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.

Chạm tới AI trong 10 ngày



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

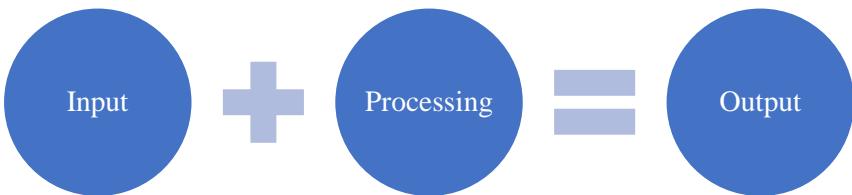
Phần mềm 7-zip

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>

Bài 6: Nhập liệu, biên tập, lưu trữ dữ liệu với R

Các bạn trong về công nghệ thông tin thì chắc là khá quen với việc mô tả một hệ thống xử lý thông tin gồm 3 phần:



Tôi viết tắt thành công thức IPO cho dễ nhớ. IPO này là Input, Processing và Output nhé, chứ không phải là Initial Public Offering (Phát hành [cổ phiếu] lần đầu tiên ra công chúng) đối với các công ty khởi nghiệp. Việc liên tưởng IPO này cũng thú vị vì khởi nghiệp đang là một xu thế. Có thể nói là ước mơ của nhiều thanh niên Việt Nam chúng ta. Đó là một dấu hiệu rất tốt. Trong việc xử lý dữ liệu cho công việc phân tích, cũng như cho các hệ thống AI làm việc thì chuẩn bị dữ liệu là cực kỳ quan trọng. Công sức cho việc chuẩn bị này được hầu hết các chuyên gia chia sẻ là tiêu tốn khá nhiều thời giờ - thường là 80% đến 90% công việc của cả dự án AI. Vì vậy phần này tôi sẽ cung cấp cho các bạn cách thức nhập liệu, các thao tác biên tập cơ bản và cả lưu lại dữ liệu sau khi biên tập.

Dữ liệu trong R

Trong R thì thông thường dữ liệu được tổ chức thành các df.frame.

Nhập liệu trực tiếp

Trong R thì dùng hàm c() để kết hợp (combine) hoặc mọc nối (concatenation) các dữ liệu thành frame. Để dễ nhớ thì các bạn có thể liên tưởng c() là column, là cột. Tức là tổ chức thành các cột dữ liệu.

Ví dụ ta có số liệu về độ tuổi và chiều cao của vài cầu thủ bóng đá trong đội tuyển Việt Nam trong năm 2019 như sau:

| Tuổi (Năm 2019) | Chiều cao (cm) | Tên |
|-----------------|----------------|-----------------------|
| 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 |

Chạm tới AI trong 10 ngày

| | | |
|----|-----|--------------------|
| 22 | 178 | Nguyễn Tiến Linh |
| 23 | 170 | Nguyễn Văn Toàn |
| 24 | 168 | Nguyễn Công Phượng |

Chúng ta có thể sử dụng hàm c() để nhập liệu cho Tuổi và Chiều cao như sau:

```
> age = c(26, 23, 23, 20, 20, 23, 23, 30, 30, 24, 24, 24, 24, 26, 26,  
22, 22, 24, 24, 23, 23, 22, 22, 23, 23, 24)  
> height = c(186, 180, 180, 185, 185, 173, 173, 169, 169, 180, 180,  
173, 173, 170, 170, 168, 168, 176, 176, 168, 168, 178, 178, 170, 170,  
168)
```

Hai biến (variable) age và height hiện tại đang là hai biến riêng rẽ, ta cần kết nối chúng lại với nhau thành một đối tượng (trong R gọi là df.frame) để phân tích:

```
> tuyenvn = df.frame(age, height)
```

Để xem lại thông tin của đối tượng 'tuyenvn' thì trong cửa sổ lệnh của R bạn gõ tên đối tượng (ở đây là tuyenvn) rồi nhấn Enter để báo cho phần mềm R biết là bạn cần xem thông tin tuyenvn.

Phần mềm RStudio sẽ hiển thị kết quả:

```
> tuyenvn  
  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
```

Chạm tới AI trong 10 ngày

| | | |
|----|----|-----|
| 22 | 22 | 178 |
| 23 | 22 | 178 |
| 24 | 23 | 170 |
| 25 | 23 | 170 |
| 26 | 24 | 168 |

Lưu lại dữ liệu

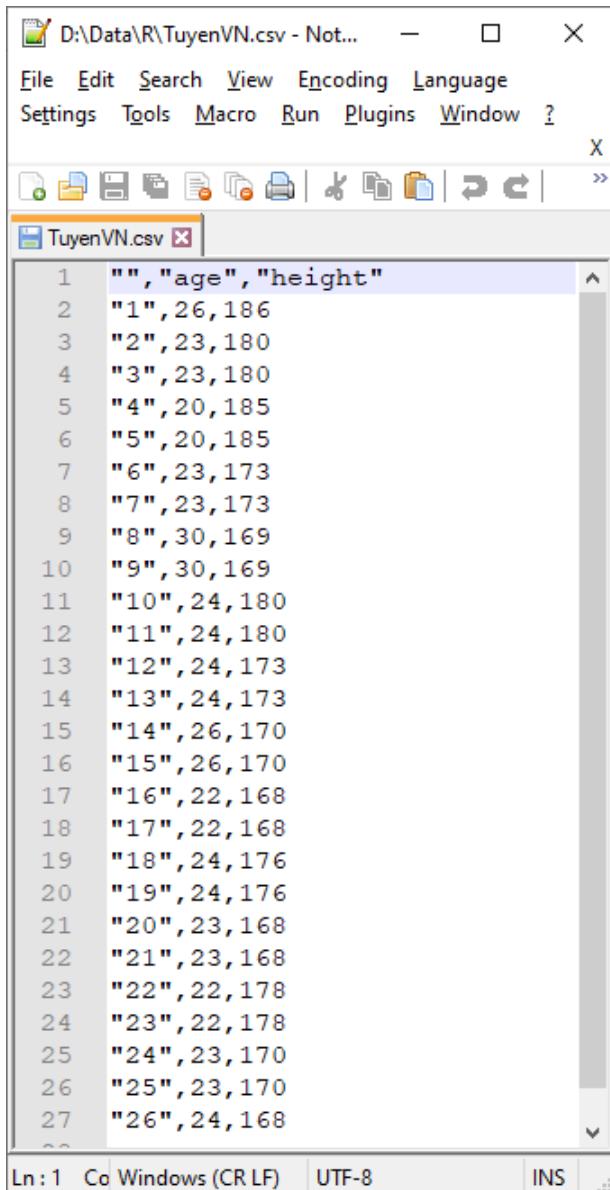
Để lưu lại lại các số liệu này vào file như csv thì dùng lệnh:

```
write.csv(data frame, đường dẫn file)
```

Ví dụ lệnh sau sẽ lưu data frame "tuyenvn" vào file 'TuyenVN.csv' trong thư mục D:/Data/R (Chú ý là trong phần mềm R và RStudio dùng dấu xuyệt phẩ / để phân cách thư mục).

```
write.csv(tuyenvn, "D:/Data/R/TuyenVN.csv")
```

Chạm tới AI trong 10 ngày



| | age | height | |
|----|---------------------|--------|--|
| 1 | "", "age", "height" | | |
| 2 | "1", 26, 186 | | |
| 3 | "2", 23, 180 | | |
| 4 | "3", 23, 180 | | |
| 5 | "4", 20, 185 | | |
| 6 | "5", 20, 185 | | |
| 7 | "6", 23, 173 | | |
| 8 | "7", 23, 173 | | |
| 9 | "8", 30, 169 | | |
| 10 | "9", 30, 169 | | |
| 11 | "10", 24, 180 | | |
| 12 | "11", 24, 180 | | |
| 13 | "12", 24, 173 | | |
| 14 | "13", 24, 173 | | |
| 15 | "14", 26, 170 | | |
| 16 | "15", 26, 170 | | |
| 17 | "16", 22, 168 | | |
| 18 | "17", 22, 168 | | |
| 19 | "18", 24, 176 | | |
| 20 | "19", 24, 176 | | |
| 21 | "20", 23, 168 | | |
| 22 | "21", 23, 168 | | |
| 23 | "22", 22, 178 | | |
| 24 | "23", 22, 178 | | |
| 25 | "24", 23, 170 | | |
| 26 | "25", 23, 170 | | |
| 27 | "26", 24, 168 | | |

Nhập liệu từ file csv

Rất nhiều nguồn dữ liệu được chia sẻ trên Internet dưới dạng file csv. CSV là viết tắt của chữ comma-separated values (các giá trị được phân cách bởi dấu phẩy). CSV là file văn bản đơn giản (không có định dạng), hay còn gọi là plain text, hoặc text.

Chúng ta sẽ nạp lại file dữ liệu đã lưu trong phần trước tại thư mục "D:/Data/R/TuyenVN.csv" bằng lệnh sau:

```
df = read.csv("D:/Data/R/TuyenVN.csv")
```

Bạn cũng có thể đọc dữ liệu trực tiếp từ Internet:

```
df = read.csv('https://thachln.github.io/datasets/TuyenVN.csv')
```

Trong trường hợp bạn không muốn gõ đường dẫn file trực tiếp thì dùng 2 lệnh sau:

Chạm tới AI trong 10 ngày

```
f = file.choose()  
df = read.csv(f)
```

Lệnh file.choose() mở hộp thoại để bạn chọn file một cách trực quan.

Sau đó gán đường dẫn của file đã chọn vào biến f.

Lệnh read.csv(f) sẽ nạp dữ liệu file từ đường dẫn trong biến f.

Xem nhanh các dữ liệu của biến df bằng các lệnh sau:

Xem số dòng, số cột của data frame "Data":

```
> dim(df)
```

```
[1] 26 3
```

Xem vài dòng dữ liệu đầu tiên của data frame "data":

```
> head(df)
```

| | X | age | height |
|---|---|-----|--------|
| 1 | 1 | 26 | 186 |
| 2 | 2 | 23 | 180 |
| 3 | 3 | 23 | 180 |
| 4 | 4 | 20 | 185 |
| 5 | 5 | 20 | 185 |
| 6 | 6 | 23 | 173 |

Xuất dữ liệu từ Excel ra file csv

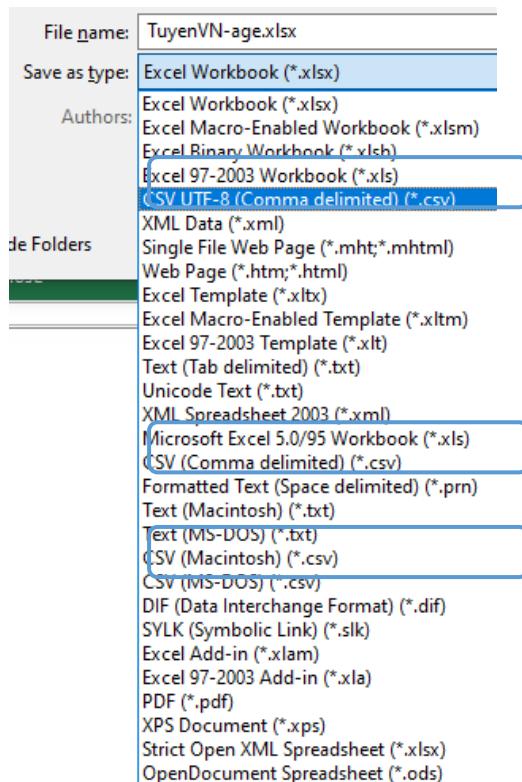
Để làm quen với file CSV thì bạn có thể dùng phần mềm trang tính như Microsoft Excel hoặc Open Office Spreadsheet để soạn dữ liệu rồi lưu dưới dạng CSV. Ví dụ bạn có file Excel chứa danh sách các tuyển thủ bóng đá như hình bên dưới:

| | A | B | C | D |
|----|-----|--------|-----------------------|---|
| 1 | age | height | name | |
| 2 | 26 | 186 | Đặng Văn Lâm | |
| 3 | 23 | 180 | Đỗ Duy Mạnh | |
| 4 | 20 | 185 | Đoàn Văn Hậu | |
| 5 | 23 | 173 | Vũ Văn Thanh | |
| 6 | 30 | 169 | Nguyễn Trọng Hoàng | |
| 7 | 24 | 180 | Quế Ngọc Hải | |
| 8 | 24 | 173 | Phạm Đức Huy | |
| 9 | 26 | 170 | Đỗ Hùng Dũng | |
| 10 | 22 | 168 | Nguyễn Quang Hải | |
| 11 | 24 | 176 | Nguyễn Tuấn Anh | |
| 12 | 23 | 168 | Nguyễn Phong Hồng Duy | |
| 13 | 22 | 178 | Nguyễn Tiến Linh | |
| 14 | 23 | 170 | Nguyễn Văn Toàn | |
| 15 | 24 | 168 | Nguyễn Công Phượng | |
| 16 | | | | |

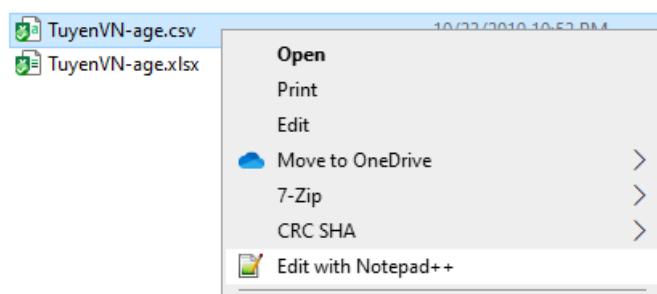
Chạm tới AI trong 10 ngày

Lưu thành file csv bằng cách vào menu File > Save As. Trong hộp thoại Save, mục Save as type bạn thấy có nhiều dạng file CSV như:

CSV UTF-8 (Comma delimited) (*.csv)



Sau khi lưu xong, bạn nên mở lại file csv để xem lại dữ liệu bằng cách trong cửa sổ Windows Explore, nhấp phải chuột vào tên file (ví dụ: “TuyenVN-age.csv”) chọn menu “Edit with Notepad++”. Đây là một sự tiện lợi với Notepad++



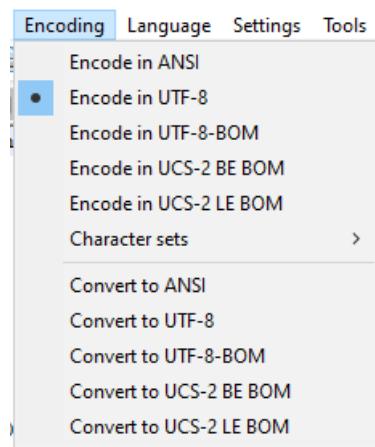
Nội dung file TuyenVN-age.csv:

Chạm tới AI trong 10 ngày

```
1 age,height,name
2 26,186,Đặng Văn Lâm
3 23,180,Đỗ Duy Mạnh
4 20,185,Đoàn Văn Hậu
5 23,173,Vũ Văn Thành
6 30,169,Nguyễn Trọng Hoàng
7 24,180,Quê Ngọc Hải
8 24,173,Phạm Đức Huy
9 26,170,Đỗ Hùng Dũng
10 22,168,Nguyễn Quang Hải
11 24,176,Nguyễn Tuấn Anh
12 23,168,Nguyễn Phong Hồng Duy
13 22,178,Nguyễn Tiến Linh
14 23,170,Nguyễn Văn Toàn
15 24,168,Nguyễn Công Phượng
16
```

Có một điểm chú ý là do Microsoft Excel lưu file dạng UTF-8-BOM. File UTF-8-BOM này có vài kí tự đặc biệt nên hiện tại thư viện trong phần mềm R không "hiểu" các kí tự đặc biệt này. Do đó bạn cần phải lưu lại file csv này dạng UTF-8 without BOM bằng cách:

Vào menu Encoding, chọn "Encode in UTF-8". Sau đó lưu lại file.



Hãy thử dùng lệnh `read.csv(file)` để đọc lại file.

Nhập liệu trực tiếp từ file Excel

Có rất nhiều thư viện giúp bạn nạp dữ liệu từ file Excel. Trong phần này tôi giới thiệu cho bạn thư viện “`readxl`” (read excel)

Ví dụ bạn có dữ liệu 24 cầu thủ của đội tuyển bóng đá Việt Nam trong file Excel như sau:

| Name | Position | Club | BirthYear | Height | Weight | BirthPlace |
|------------------|----------|--------------------|-----------|--------|--------|------------|
| Đặng Văn Lâm | Thủ môn | Muang Thong United | 1993 | 186 | 76 | Nga |
| Nguyễn Tuấn Mạnh | Thủ môn | Khánh Hòa | 1990 | 177 | 72 | Thanh Hóa |

Chạm tới AI trong 10 ngày

| | | | | | | |
|-----------------------|----------|----------------|------|-----|----|-------------|
| Phạm Văn Cường | Thủ môn | Quảng Nam | 1990 | 186 | 70 | Nghệ An |
| Đỗ Duy Mạnh | Hậu vệ | Hà Nội | 1996 | 180 | 70 | Hà Nội |
| Đoàn Văn Hậu | Hậu vệ | Hà Nội | 1999 | 185 | 70 | Thái Bình |
| Trần Văn Kiên | Hậu vệ | Hà Nội | 1996 | 168 | 64 | Nghệ An |
| Nguyễn Thành Chung | Hậu vệ | Hà Nội | 1997 | 180 | 70 | Tuyên Quang |
| Vũ Văn Thanh | Hậu vệ | HAGL | 1996 | 173 | 67 | Hải Dương |
| Nguyễn Hữu Tuấn | Hậu vệ | TP.HCM | 1992 | 179 | 70 | Đà Nẵng |
| Nguyễn Trọng Hoàng | Hậu vệ | Viettel | 1989 | 169 | 67 | Nghệ An |
| Bùi Tiến Dũng | Hậu vệ | Viettel | 1995 | 176 | 75 | Hà Tĩnh |
| Quế Ngọc Hải | Hậu vệ | Viettel | 1995 | 180 | 77 | Nghệ An |
| Phạm Đức Huy | Tiền vệ | Viettel | 1995 | 73 | 65 | Hải Dương |
| Đỗ Hùng Dũng | Tiền vệ | Hà Nội | 1993 | 170 | 67 | Hà Nội |
| Nguyễn Quang Hải | Tiền vệ | Hà Nội | 1997 | 168 | 65 | Hà Nội |
| Nguyễn Tuấn Anh | Tiền vệ | HAGL | 1995 | 176 | 65 | Thái Bình |
| Lương Xuân Trường | Tiền vệ | HAGL | 1995 | 178 | 72 | Tuyên Quang |
| Nguyễn Phong Hồng Duy | Tiền vệ | HAGL | 1996 | 168 | 67 | Bình Phước |
| Nguyễn Huy Hùng | Tiền vệ | Quảng Nam | 1992 | 174 | 69 | Hà Nội |
| Nguyễn Anh Đức | Tiền đạo | Bình Dương | 1985 | 181 | 72 | Sông Bé |
| Nguyễn Tiến Linh | Tiền đạo | Bình Dương | 1997 | 178 | 67 | Hải Dương |
| Nguyễn Văn Toàn | Tiền đạo | HAGL | 1996 | 170 | 61 | Hải Dương |
| Nguyễn Công Phượng | Tiền đạo | Sint Truidense | 1995 | 168 | 65 | Nghệ An |
| Hà Minh Tuấn | Tiền đạo | Quảng Nam | 1991 | 178 | 71 | Quảng Nam |

Cài đặt thư viện

```
install.packages('readxl')
```

Sử dụng thư viện

```
library('readxl')
d = read_excel("D:/Temp/TuyenVN_2019.xlsx")
head(d)
```

| Name | Position | Club | BirthYear | Height | Weight | BirthPlace |
|--------------------|----------|--------------------|-----------|--------|--------|------------|
| <chr> | <chr> | <chr> | <dbl> | <dbl> | <dbl> | <chr> |
| 1 Đặng Văn Lâm | Thủ môn | Muang Thong United | 1993 | 186 | 76 | Nga |
| 2 Nguyễn Tuấn Mạnh | Thủ môn | Khánh Hòa | 1990 | 177 | 72 | Thanh Hóa |
| 3 Phạm Văn Cường | Thủ môn | Quảng Nam | 1990 | 186 | 70 | Nghệ An |
| 4 Đỗ Duy Mạnh | Hậu vệ | Hà Nội | 1996 | 180 | 70 | Hà Nội |
| 5 Đoàn Văn Hậu | Hậu vệ | Hà Nội | 1999 | 185 | 70 | Thái Bình |
| 6 Trần Văn Kiên | Hậu vệ | Hà Nội | 1996 | 168 | 64 | Nghệ An |

Xử lý dữ liệu nâng cao

Phần này tôi gọi là nâng cao vì là ngày đầu tiên nên bạn có thể bỏ qua phần này. Nội dung nâng cao này sẽ được sử dụng cho các vấn đề phức tạp hơn trong các ngày tiếp theo. Tuy nhiên tôi vẫn trình bày ở đây để khi nào cần thiết thì bạn có thể tra cứu, bổ sung kỹ năng khi cần thiết.

Lọc dữ liệu theo cột

Bạn muốn trích xuất dữ liệu từ một data frame có sẵn bằng cách chỉ chọn một vài cột dữ liệu nào đó thôi. Sử dụng hàm `select` trong thư viện `dplyr`.

Ví dụ:

```
packages <- c('dplyr')
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

df = read.csv("https://thachln.github.io/datasets/TuyenVN_2019.csv")
library(dplyr)
names(df)
d1 = select(df, Height, Weight)
head(d1)

  Height Weight
1    186     76
2    177     72
3    186     70
4    180     70
5    185     70
6    168     64
```

Bỏ bớt cột dữ liệu

Bỏ một cột

Dùng cú pháp:

```
dataframe[, -col_index]
```

`col_index`: vị trí của cột dữ liệu (cột bên trái cùng tính từ 1)

Bỏ một cột dùng thư viện `dplyr`

Dùng thư viện `dplyr` với toán tử pipeline (`%>%`)

```
dataframe %>% select(-starts_with("col_name"))
```

Bỏ nhiều cột dùng thư viện dplyr

Dùng thư viện dplyr với toán tử pipeline (%>%)

```
drop_cols = c('colname1', 'colname2', 'colnameN')
dataframe %>% select(-one_of(drop_cols))
```

Bài 7: 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 – khái niệm tương tự như trong R.

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

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/R/TuyenVN.csv', index_col = 0)
```

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

```
data
```

Chạm tới AI trong 10 ngày

The screenshot shows the Spyder Python IDE interface. In the top menu bar, the title is "Spyder (Python 3.7)". Below it, the menu items are: File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help. The toolbar has various icons for file operations like Open, Save, Copy, Paste, etc. 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
```

The right side of the interface shows the IPython console with the following output:

```
In [42]: data
Out[42]:
   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
```

Below the IPython console, there are tabs for "IPython console" and "History log". At the bottom of the interface, there are status bars for "Permissions: RW", "End-of-lines: CRLF", "Encoding: UTF-8", "Line: 11", "Column: 1", "Memory: 57 %", and a zoom control.

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

```
data.shape
```

Kết quả:

```
(26, 3)
```

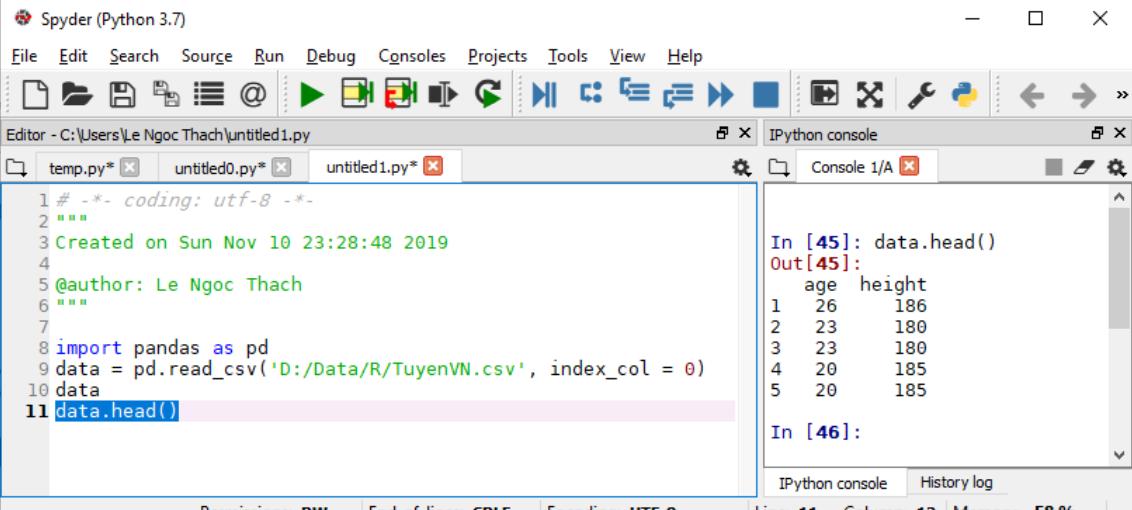
Để xem vài dòng dữ liệu **đầu** và **cuối** của data frame thì dùng hàm head() và tail().

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

Chạm tới AI trong 10 ngày



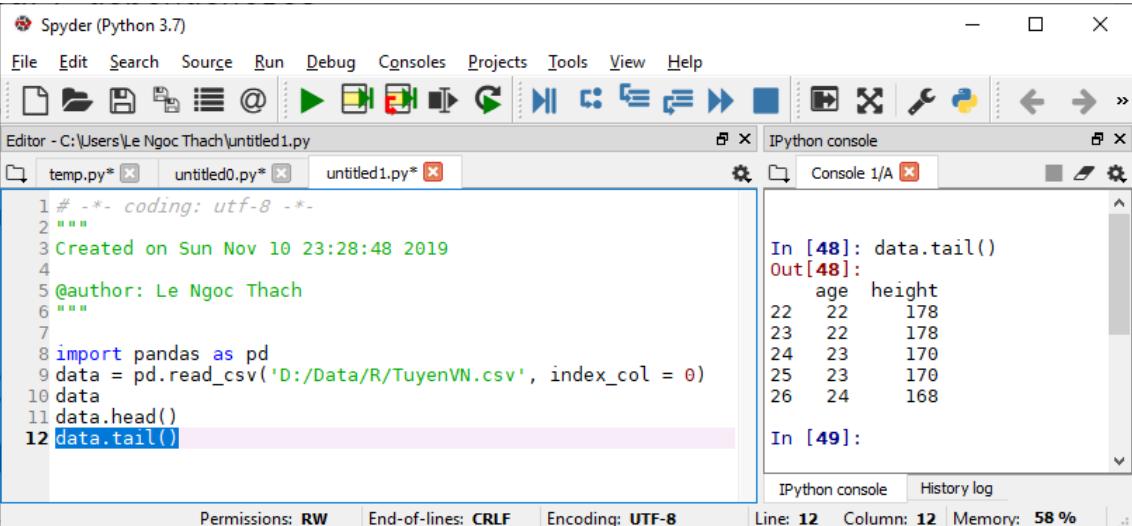
Screenshot of the Spyder Python 3.7 IDE. The code editor shows a script named 'untitled1.py' with the following content:

```
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 IPython console shows the output of the '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
```

Line 11 is highlighted in blue.



Screenshot of the Spyder Python 3.7 IDE. The code editor shows the same script 'untitled1.py' with the 'head()' command removed and replaced by 'tail()':

```
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()
```

The IPython console shows the output of the 'tail()' command:

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

Line 12 is highlighted in blue.

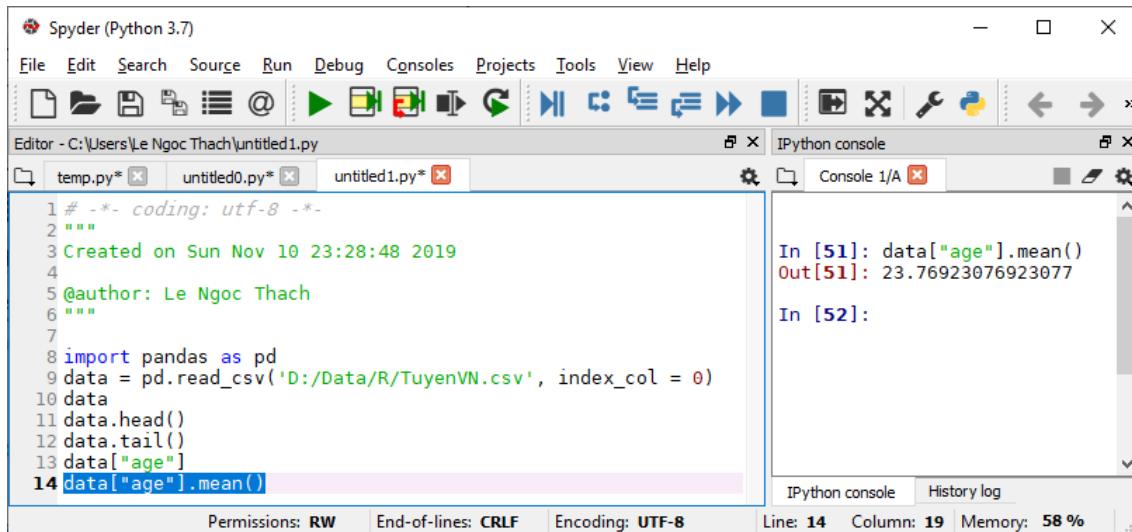
Để lấy ra cột dữ liệu "age" thì dùng cú pháp sau:

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

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

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

Chạm tới AI trong 10 ngày



The screenshot shows the Spyder Python IDE interface. The code editor on the left contains a script named 'temp.py' with the following content:

```
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()
```

The IPython console on the right shows the output of the last command:

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

Below the console, status information is displayed: 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 |
|---------------------------|---------------------------------|
| <code>data.columns</code> | Xem tên các cột của data frame. |

Xử lý dữ liệu nâng cao

Tương tự như mục này trong bài trước của R, tôi gọi là nâng cao vì là ngày đầu tiên nên bạn có thể bỏ qua. Nội dung nâng cao này sẽ được sử dụng cho các vấn đề phức tạp hơn trong các ngày tiếp theo. Khi cần thiết thì tra cứu lại phần này.

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'>
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 dòng 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 |

Chạm tới AI trong 10 ngày

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
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 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.

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.

Lọc dữ liệu theo dòng với điều kiện cho trước

Ví dụ: lọc các dòng dữ liệu trong 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 |

Lưu dataframe ra file csv

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

Ví dụ: `df.to_csv('out.csv')`

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 cá 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 mát 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 điểm 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, đội khi có thể thay ngay các thông tin ẩn (hidden) đằng sau "bức tranh" mà nếu chỉ nhìn con số thì rất khó thấy. Ngày thứ hai này chúng ta sẽ dạo qua các loại biểu đồ và làm quen với công việc phân tích biểu đồ. Trong đó sẽ có các ví dụ minh họa bằng ngôn ngữ **R** và cả **Python** để giúp các bạn khám phá cái hay, cái chưa hay của từng ngôn ngữ và phần mềm tương ứng.

Ngày thứ hai này sẽ gồm 5 bài:

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. Tôi nghĩ dùng từ mô hình hóa bằng máy (nếu tiếng anh viết 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 dự 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 nhập 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 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.

Bài 8: Tóm tắt và giúp các bạn phân biệt được mục đích cơ bản của các loại biểu đồ.

Bài 9: 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 10: 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 11: Sưu tầm vài nguyên tắc soạn biểu đồ.

Bài 12: Giúp bạn làm quen với thư viện vẽ biểu đồ rất phổ biến trong Python.

Bài 8: 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 R và 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.

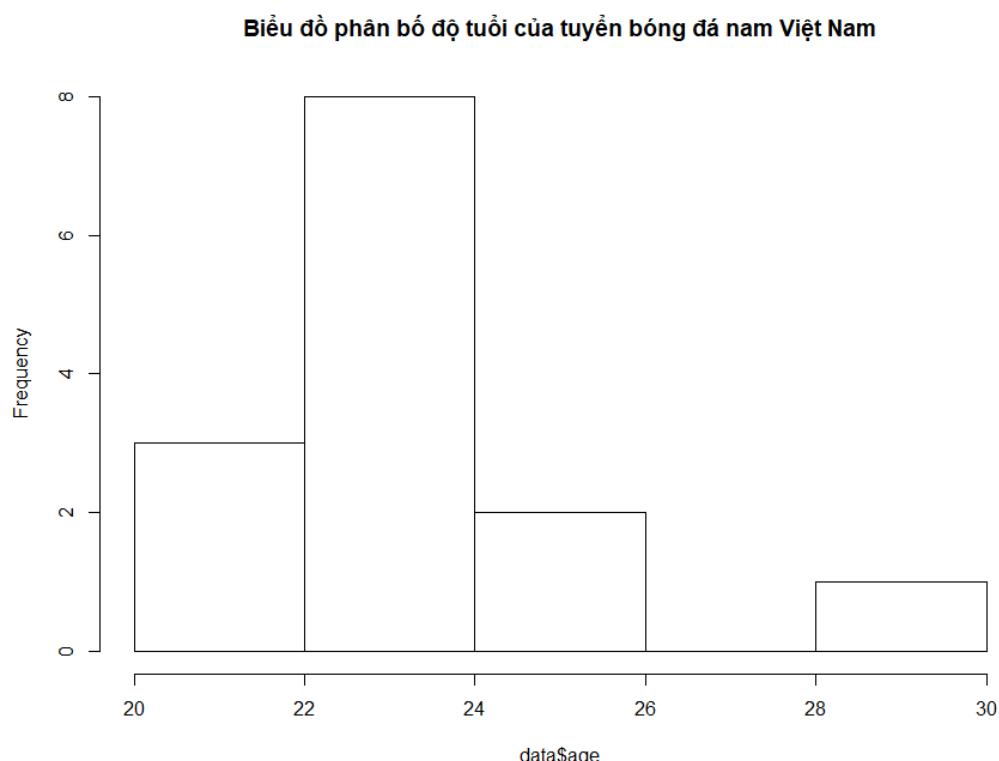
Bảng bên dưới trình bày mục đích hoặc chủ định của các bạn muốn làm gì ở cột **bên trái**. Tương ứng với chủ định thì cột **bên phải** sẽ cho biết nên dùng biểu đồ gì.

| Mục đích | Biểu đồ |
|--|--|
| Cần nhìn thấy bức tranh tổng thể về phân số dữ liệu | |
| Biểu diễn phân bố dữ liệu | Histogram |
| 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) | Boxplot |
| 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. | Bar chart/Bar plot : biểu đồ cột (tên khác là Column chart) |
| So sánh giá trị một biến thay đổi theo thời gian . | 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 biến 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). |

| | |
|---|----------------------------------|
| | Scatterplot. |
| Tương quan giữa 2 biến trong đó có chia theo nhóm | 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 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

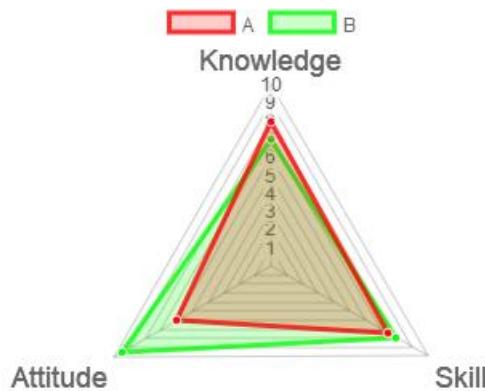
Radar chart có nhiều tên khác như spider chart (biểu đồ mạng nhện – do nó giống cái mạng nhện), hoặc web chart. Radar chart thể hiện nhiều biến trên một đa giác. Mỗi giá trị của biến tương ứng với mỗi đỉnh của đa giác.

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

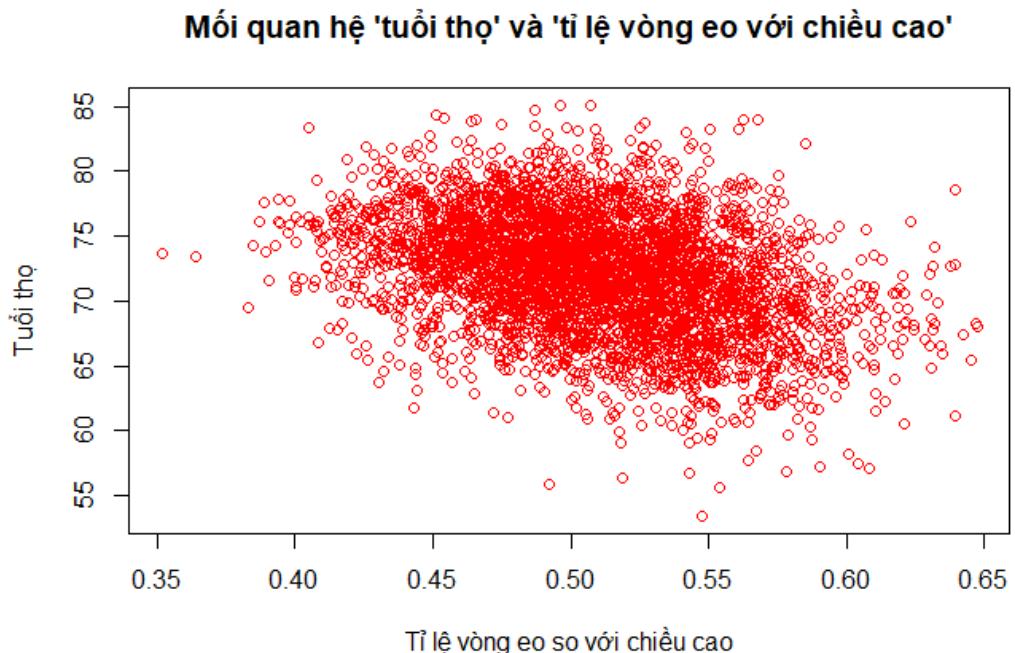
Scatter plot thể hiện các điểm (point) của hai biến số, hoặc nói cách khác là giá trị của một biến được thể hiện trong mối tương quan của hai trục x và y. Biểu đồ này cũng rất hữu ích khi cần quan sát mối tương quan giữa hay nhiều biến số trong nhiều nhóm.

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. Như 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 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)
```

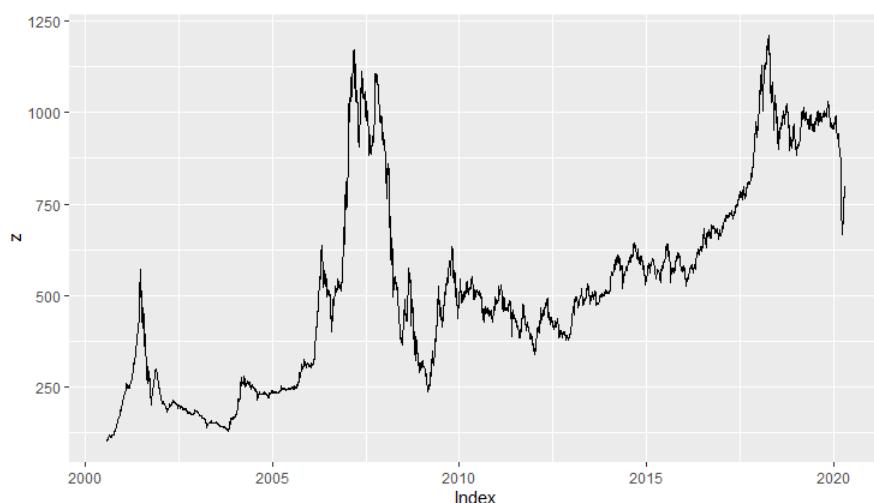
0.9642857142857141

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 9: Vẽ biểu đồ trong R

Bài này trình bày mã nguồn R để giúp các bạn vẽ được các loại biểu đồ trong Bài 8. Mục đích của bài này là giúp bạn làm quen với kỹ thuật vẽ biểu đồ với R thôi chứ không đi sâu vào phân tích và giải thích.

Vài khái niệm cơ bản

Trong R, biểu đồ được xem như là một bức tranh do cách họa sĩ vẽ. Một biểu đồ gồm nhiều thành phần được xếp đặt theo một ý nghĩa nào đó giống như họa sĩ sắp đặt bức tranh.

Ngoài thành phần chính là các loại biểu đồ và số liệu tương ứng thì các thành phần mang tính chất trang trí sau cũng là một phần rất quan trọng:

Trang trí

| Chú pháp | Ghi chú |
|--|--|
| grid(nx , ny) | Thêm lưới cho biểu đồ |
| axis(side n ,) | Thêm trực x, y cho biểu đồ |
| box(which= ,) | Thêm box xung quanh biểu đồ |
| legend | Thêm ghú thíc nhän |
| arrows(x , y) lines(x , y) points(x , y) | Thêm mũi tên, đường thẳng, kiểu của điểm ảnh (p, b, l, ...) |
| abline(a , b) abline(h= or v=) | Thêm đường biểu diễn (a: intercept; b: slope). h: horizontal; v: vertical |
| segments(x0 , x1 , y0 , y1) | Thêm đoạn thẳng giữa 2 điểm (x_0, y_0) và (x_1, y_1) |
| polygon(x , y) | Thêm đa giác xác định bởi vector x và y |
| text(x , y , "note") | Thêm chữ trong biểu đồ |

Plot characters (pch)

pch dùng để thể hiện kí hiệu tại điểm dữ liệu trên biểu đồ.

R cung cấp 26 hình ảnh tương ứng với số thứ tự từ 0 đến 25 và có thể dùng thêm các kí tự (xem bảng bên phải)

| | | | |
|-------|-------|-------|-------|
| 0: □ | 10: ⊕ | 20: ● | A: A |
| 1: ○ | 11: ✖ | 21: ● | a: a |
| 2: △ | 12: ▨ | 22: ■ | B: B |
| 3: + | 13: ✖ | 23: ♦ | b: b |
| 4: ✖ | 14: ▨ | 24: ▲ | S: S |
| 5: ◊ | 15: ■ | 25: ▼ | `: ` |
| 6: ▽ | 16: ● | @:@ | .: . |
| 7: ✩ | 17: ▲ | +:+ | ,: , |
| 8: *: | 18: ♦ | %:% | ??: ? |
| 9: ✧ | 19: ● | #:# | *:* |

Loại đường kẻ (Line Type)

Tham số lty sẽ chỉ định loại đường kẻ trong biểu đồ.

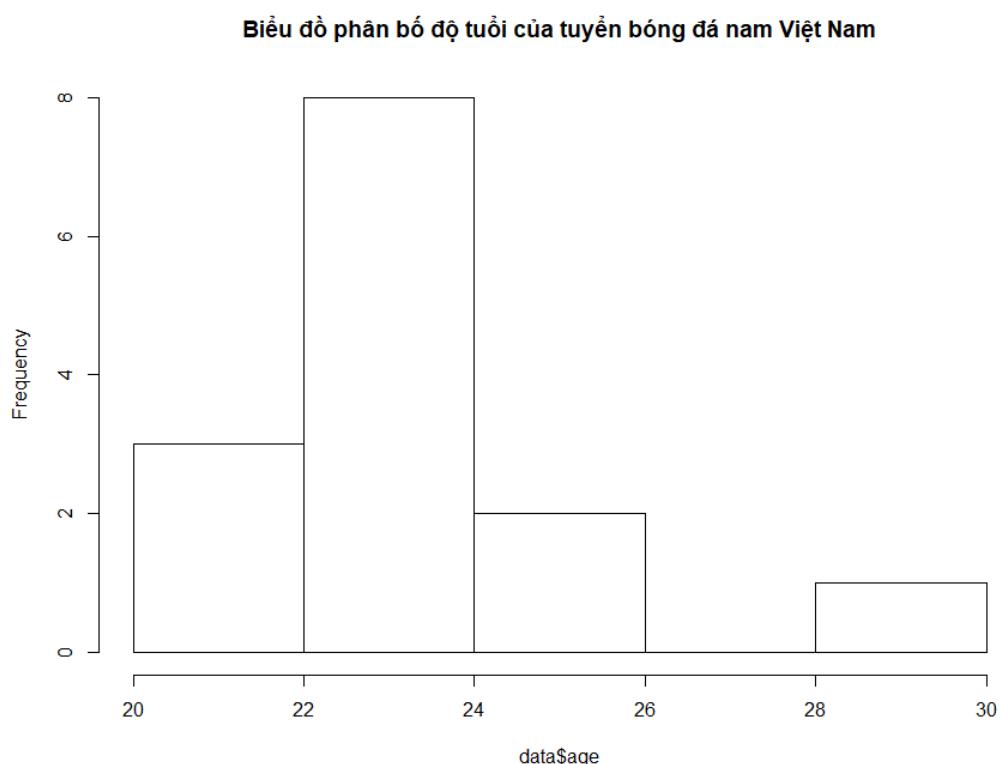
lty có thể được khai báo dạng số thứ tự hoặc bằng tên (character string) như hình bên.

- Ity = 1 hoặc 'solid'
- - - Ity = 2 hoặc 'dashed'
- Ity = 3 hoặc 'dotted'
- - Ity = 4 hoặc 'dotdash'
- - - Ity = 5 hoặc 'longdash'
- Ity = 6 hoặc 'twodash'

Biểu đồ phân bố dữ liệu – Histogram

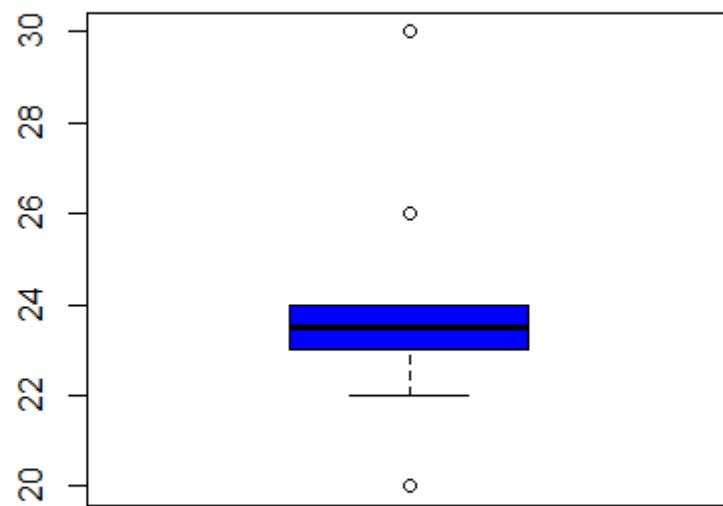
Kết quả hai lệnh bên dưới:

```
df = read.csv('https://thachln.github.io/datasets/TuyenVN.csv', header = T)  
hist(data$age, main = "Biểu đồ phân bố độ tuổi của tuyển bóng đá nam  
Việt Nam")
```



Biểu đồ phân bố dữ liệu – Boxplot

```
df = read.csv('https://thachln.github.io/datasets/TuyenVN.csv', header=T)  
boxplot(data$age, col = 'blue')
```



Biểu đồ so sánh - Line Chart

Ví dụ 1: Thống kê tăng trưởng GDP và CPI (đơn vị: %) của Việt Nam từ năm 2010 đến 2018 như sau

| Year | GDP | CPI |
|------|------|-------|
| 2010 | 6.78 | 11.75 |
| 2011 | 5.89 | 18.13 |
| 2012 | 5.03 | 6.81 |
| 2013 | 5.42 | 6.04 |
| 2014 | 5.98 | 4.09 |
| 2015 | 6.68 | 0.6 |
| 2016 | 6.21 | 4.74 |
| 2017 | 6.81 | 3.53 |
| 2018 | 7.08 | 3.54 |
| 2019 | 7.02 | 2.79 |

Để làm quen với các lệnh vẽ biểu đồ đơn giản trong R bạn cần tìm hiểu lệnh **plot**.

Trong R để tìm hiểu một lệnh hoặc hàm thì dùng cú pháp

```
? < lệnh >
```

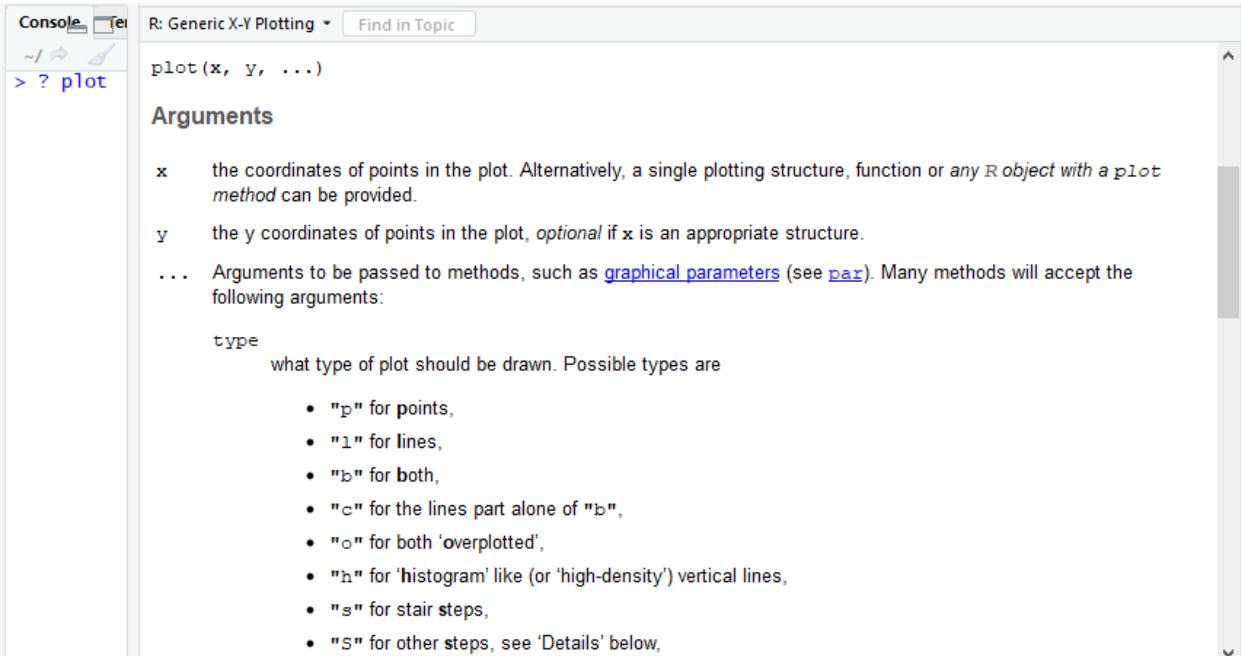
Sau dấu hỏi có hoặc không có khoảng trắng.

Ví dụ:

Chạm tới AI trong 10 ngày

```
> ? plot
```

Sẽ cho hướng dẫn như sau:



The screenshot shows the R help window for the 'plot' function. The title bar says 'R: Generic X-Y Plotting'. The search bar says 'Find in Topic'. The main text area starts with 'plot(x, y, ...)'. Below it is a section titled 'Arguments' which includes three parameters: 'x', 'y', and '...'. 'x' is described as the coordinates of points or a plotting structure. 'y' is the y coordinates of points. '...' is for graphical parameters. A detailed description of the 'type' argument follows, listing nine possibilities: 'p' (points), 'l' (lines), 'b' (both), 'c' (lines part alone of 'b'), 'o' (overplotted), 'h' (histogram-like vertical lines), 's' (stair steps), and 'S' (other steps). The help window has a standard Windows-style interface with icons for file, edit, and help.

Bạn chưa quen với cách đọc hướng dẫn này thì cũng không sao. Cứ cố gắng đọc và dịch để hiểu nghĩa từ từ. Sau này quen dần sẽ thấy rất tiện lợi.

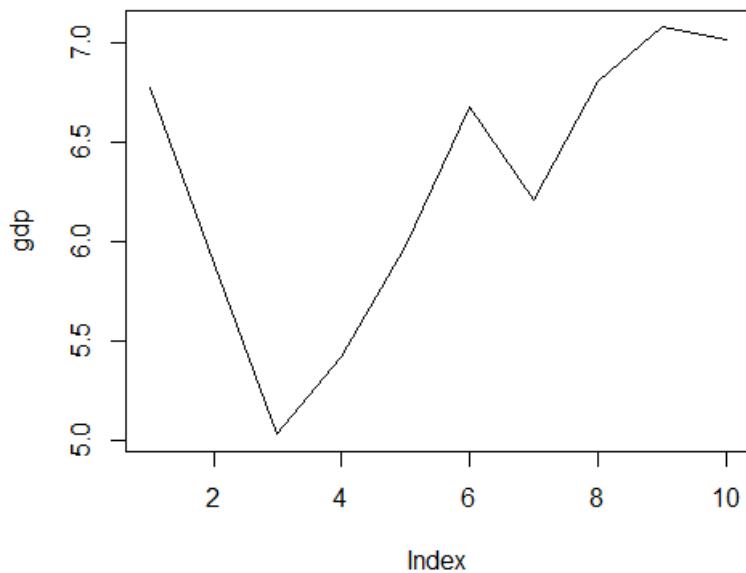
Quay lại ví dụ về mức tăng trưởng GPD của chúng ta, để chuẩn bị dữ liệu thì chúng ta cần khai báo một biến **gdp** để chứa danh sách các giá trị:

```
> gdp = c(6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02)  
> year = c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019)
```

Sau đó dùng hàm **plot** với tham số **type = 'l'** (chữ l trong chữ **line** chứ không phải số 1 nhé, chú ý bao đóng chữ l trong cặp dấu ngoặc ‘ ’ hoặc “ ”).

```
> plot(gdp, type = 'l')
```

Kết quả sẽ có biểu đồ sau:



Do chúng ta không ghi chú trực hoành (trục x) nên mặc định hàm plot sẽ lấy chỉ số (số thứ tự) của các giá trị làm trực hoành. Trục tung (trục y) chính là giá trị của biến đang vẽ, ở đây là biến **gdp**.

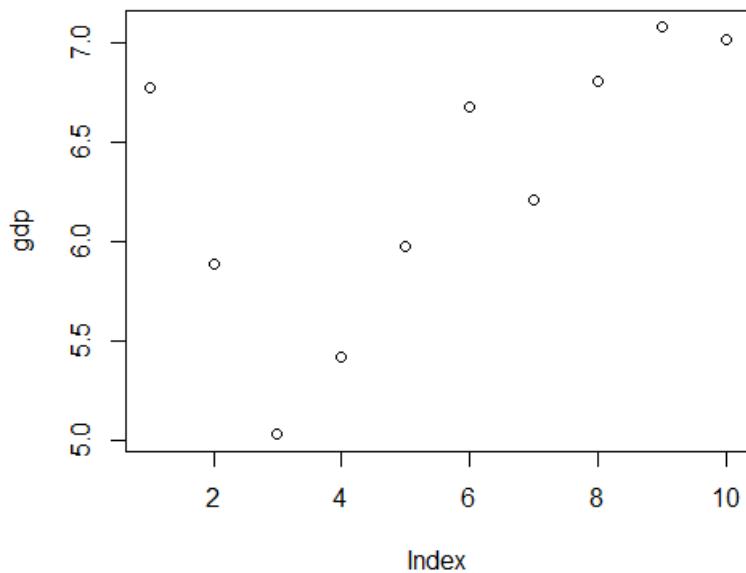
Như vậy với một lệnh plot với type = ‘l’ thì chúng ta có ngay một biểu đồ line. Tuy nhiên, biểu đồ này còn rất thô sơ và chưa được trang trí gì cả. Ở đây chúng ta chỉ làm quen và bạn hãy cố gắng gõ lệnh vào R Studio để cảm nhận và quen tay.

Thử dùng lệnh plot không có tham số type xem sao:

```
> plot(gdp)
```

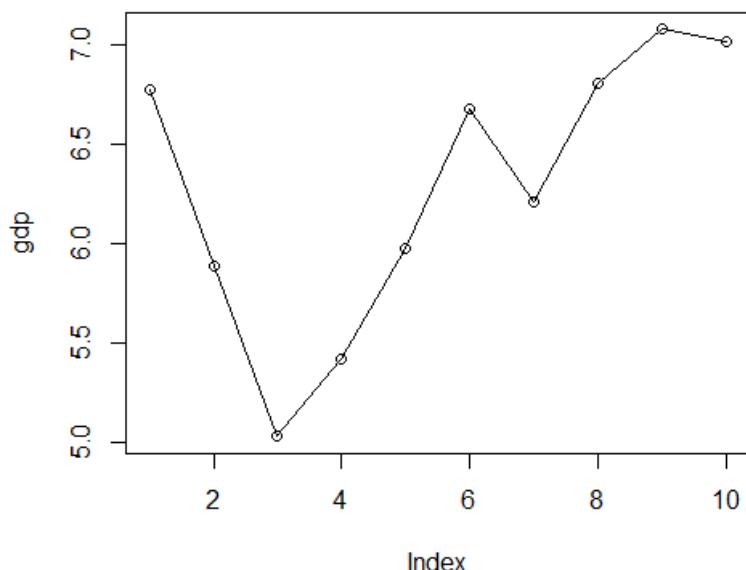
Kết quả sẽ ra các điểm tròn như sau:

Chạm tới AI trong 10 ngày



Hãy xem lại tài liệu hướng dẫn bằng lệnh “`? plot`”. Sau đó thử plot lại gdp với `type = 'o'` xem sao nhé!

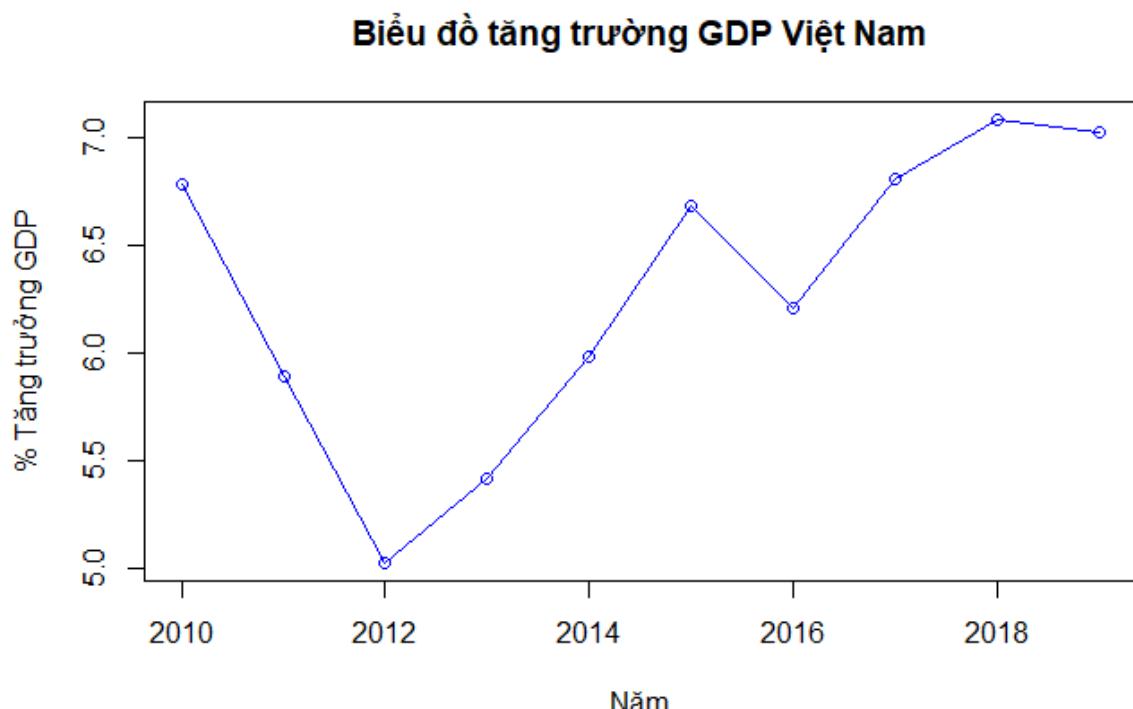
```
plot(gdp, type = 'o')
```



Nâng cấp thêm lệnh `plot` và trang trí một chút:

Chạm tới AI trong 10 ngày

```
plot(gdp ~ year, type = 'o', col = 'blue', xlab = 'Năm', ylab = '% Tăng trưởng GDP', main = 'Biểu đồ tăng trưởng GDP Việt Nam')
```



Học thêm các tham số:

- `gdp ~ year`: yêu cầu R plot giá trị gdp theo trực tung (y) theo các giá trị năm trên trực hoành (x).
- `type = 'o'`: Type có giá trị là kí tự o thường yêu cầu R plot dấu tròn tại các giá trị của y.
- `col = 'blue'`: (col viết tắt của color) vẽ màu xanh
- `xlab, ylab`: (lab viết tắt của label) cho biết nhãn của trực x và trực y.
- `main`: cho biết tiêu đề của biểu đồ.



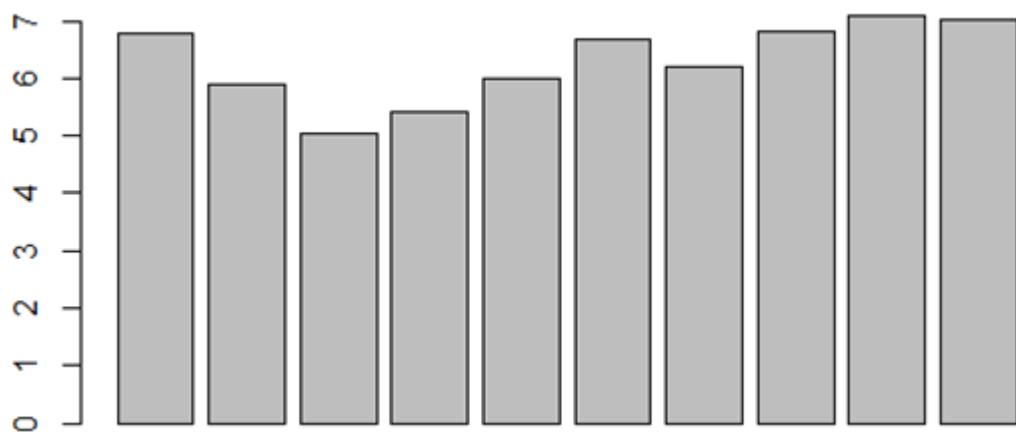
| Thử thách | |
|-----------|--|
| ① | Hãy tạo dữ liệu cho chỉ số tăng trưởng CPI và dùng lệnh plot để vẽ điều đồ line. |
| ② | Dùng lệnh ? để tìm hiểu thêm lệnh plot và thử dùng tham số type với các giá trị khác nhau. |

Biểu đồ so sánh - Bar Chart

Ví dụ:

Quay lại ví dụ chỉ số tăng trưởng GDP ở trên, chúng ta có thể vẽ biểu đồ thanh bằng lệnh **barplot**:

```
> barplot(gdp)
```



Tương tự lệnh plot ở trên, thêm vài tham số cho lệnh boxplot:

```
barplot(gdp ~ year, col = 'blue', xlab = 'Năm', ylab = '% Tăng trưởng GDP', main = 'Biểu đồ tăng trưởng GDP Việt Nam')
```



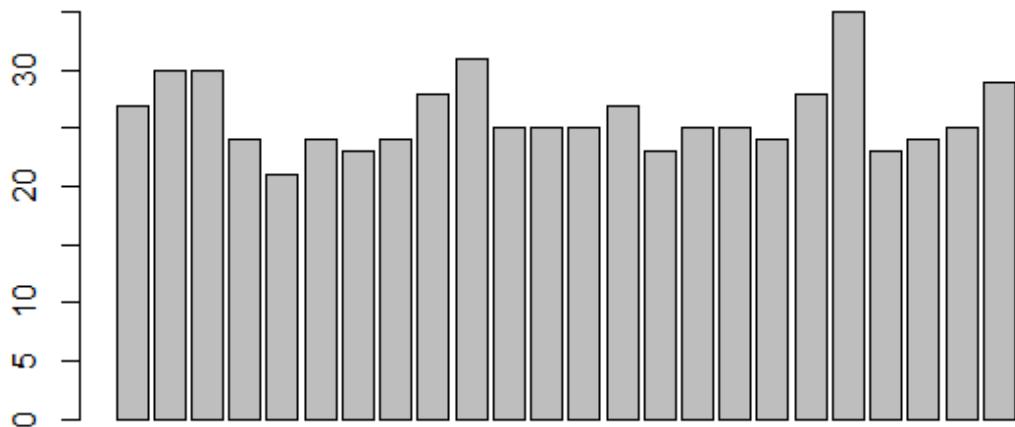
Chạm tới AI trong 10 ngày

Ví dụ thống kê năm sinh (BirthYear), chiều cao (Height) và cân nặng (Weight) của các cầu thủ của đội tuyển nam Việt Nam trong năm 2019 như sau (Xem ví dụ trong Bài 6):

| Name | BirthYear | Height | Weight |
|-----------------------|-----------|--------|--------|
| Đặng Văn Lâm | 1993 | 186 | 76 |
| Nguyễn Tuấn Mạnh | 1990 | 177 | 72 |
| Phạm Văn Cường | 1990 | 186 | 70 |
| Đỗ Duy Mạnh | 1996 | 180 | 70 |
| Đoàn Văn Hậu | 1999 | 185 | 70 |
| Trần Văn Kiên | 1996 | 168 | 64 |
| Nguyễn Thành Chung | 1997 | 180 | 70 |
| Vũ Văn Thanh | 1996 | 173 | 67 |
| Nguyễn Hữu Tuấn | 1992 | 179 | 70 |
| Nguyễn Trọng Hoàng | 1989 | 169 | 67 |
| Bùi Tiến Dũng | 1995 | 176 | 75 |
| Quế Ngọc Hải | 1995 | 180 | 77 |
| Phạm Đức Huy | 1995 | 173 | 65 |
| Đỗ Hùng Dũng | 1993 | 170 | 67 |
| Nguyễn Quang Hải | 1997 | 168 | 65 |
| Nguyễn Tuấn Anh | 1995 | 176 | 65 |
| Lương Xuân Trường | 1995 | 178 | 72 |
| Nguyễn Phong Hồng Duy | 1996 | 168 | 67 |
| Nguyễn Huy Hùng | 1992 | 174 | 69 |
| Nguyễn Anh Đức | 1985 | 181 | 72 |
| Nguyễn Tiến Linh | 1997 | 178 | 67 |
| Nguyễn Văn Toàn | 1996 | 170 | 61 |
| Nguyễn Công Phượng | 1995 | 168 | 65 |
| Hà Minh Tuấn | 1991 | 178 | 71 |

Lệnh R để đọc file excel và vẽ biểu đồ cột đơn giản bằng barplot như sau:

```
# install.packages('readxl')
library('readxl')
d = read_excel('D:/Temp/tuyenvn.xlsx')
head(d)
currYear = as.numeric(format(Sys.time(), "%Y"))
currYear
barplot(currYear - d$BirthYear)
```



Lệnh `install.packages('readxl')` được đặt sau dấu # (đọc là **thăng**) có nghĩa là chú thích (R sẽ không thực thi lệnh này). Nếu bạn chưa cài thư viện `readxl` thì hãy bỏ dấu thăng đi. Lệnh `barplot` được minh họa ở đây đơn giản gồm một biến tuổi. Tuổi được tính bằng cách lấy năm hiện tại trừ cho năm sinh.

Để lấy thời gian hiện tại của máy thì dùng hàm `Sys.time()`. Sau đó dùng hàm `format(..., "%Y")` để lấy ra 4 ký tự của năm. Tiếp theo dùng hàm `as.numeric` để chuyển ký tự thành số. Kết quả năm hiện tại được gán vào biến currYear bởi lệnh sau:

```
currYear = as.numeric(format(Sys.time(), "%Y"))
```

Dấu đô la “\$” trong biểu thức `d$BirthYear` có nghĩa là truy xuất cột dữ liệu `BirthYear` trong biến d (d ở đây là data frame).

Biểu đồ so sánh - Radar Chart

Code R sau hiển thị Radar chart để điểm ASK (Attitude, Skill, Knowledge – Thái độ, Kỹ năng, Kiến thức) của 2 ứng viên A và B.

```
packages <- c('radarchart')
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
```

Ghi nhớ

Để truy xuất một trường dữ liệu (field) hoặc cột (column) của data frame thì dùng dấu \$ đặt giữa biến data frame và column.

```
}
```

```
library('radarchart')
```

```
labs = c('Knowledge', 'Skill', 'Attitude')
```

```
scores = list(
```

```
  'A' = c(8, 7.5, 6),
```

```
  'B' = c(7, 8, 9.5)
```

```
)
```

```
chartJSRadar(scores = scores, labs = labs, maxScale = 10)
```

Biểu đồ tương quan đơn giản

Đọc dữ liệu:

```
df =
```

```
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
```

```
header = T)
```

Xem vài dòng dữ liệu:

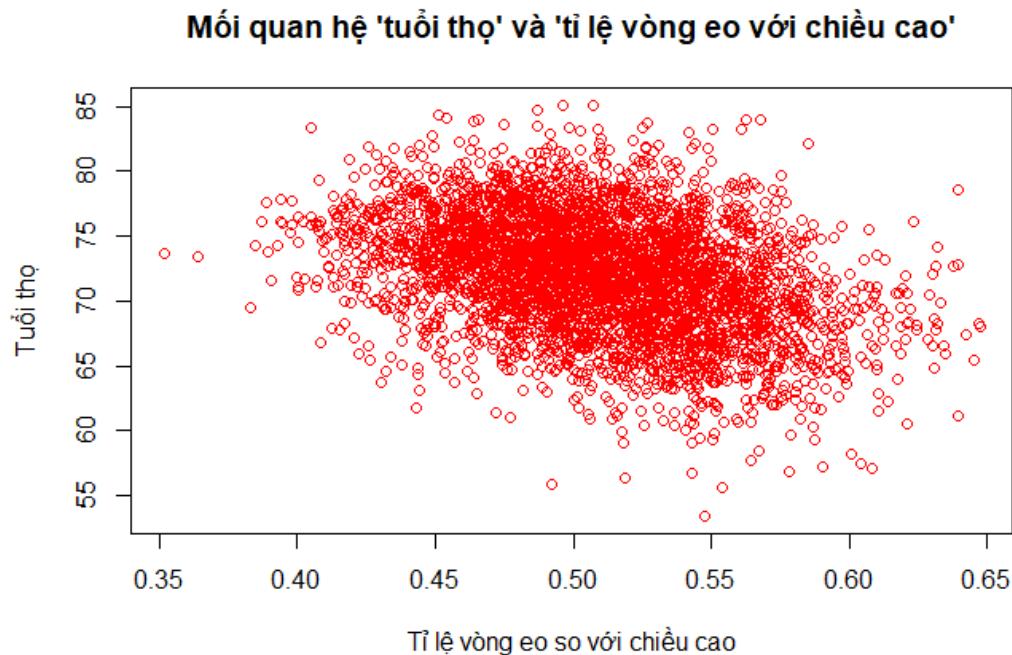
```
head(df)
```

| | id | age | sex | height | waist | risk | weight | hit | life |
|---|----|-----|-----|--------|-------|------|--------|-----|-------|
| 1 | 1 | 39 | 1 | 168 | 70 | 1 | 64 | 85 | 72.05 |
| 2 | 2 | 31 | 0 | 159 | 84 | 0 | 67 | 78 | 68.16 |
| 3 | 3 | 29 | 1 | 163 | 93 | 0 | 54 | 95 | 74.49 |
| 4 | 4 | 55 | 1 | 161 | 84 | 1 | 46 | 82 | 76.93 |
| 5 | 5 | 62 | 1 | 163 | 86 | 0 | 59 | 94 | 72.53 |
| 6 | 6 | 33 | 0 | 158 | 80 | 0 | 40 | 82 | 78.71 |

Chúng ta tự tính cột **whtr** rồi vẽ biểu đồ bằng hàm **plot**:

```
df$whtr = df$waist / df$height
```

```
plot(df$whtr, df$life, col = 'red', ylab = "Tuổi thọ", xlab = "Tỉ lệ  
vòng eo so với chiều cao", main = "Mối quan hệ 'tuổi thọ' và 'tỉ lệ  
vòng eo với chiều cao'")
```



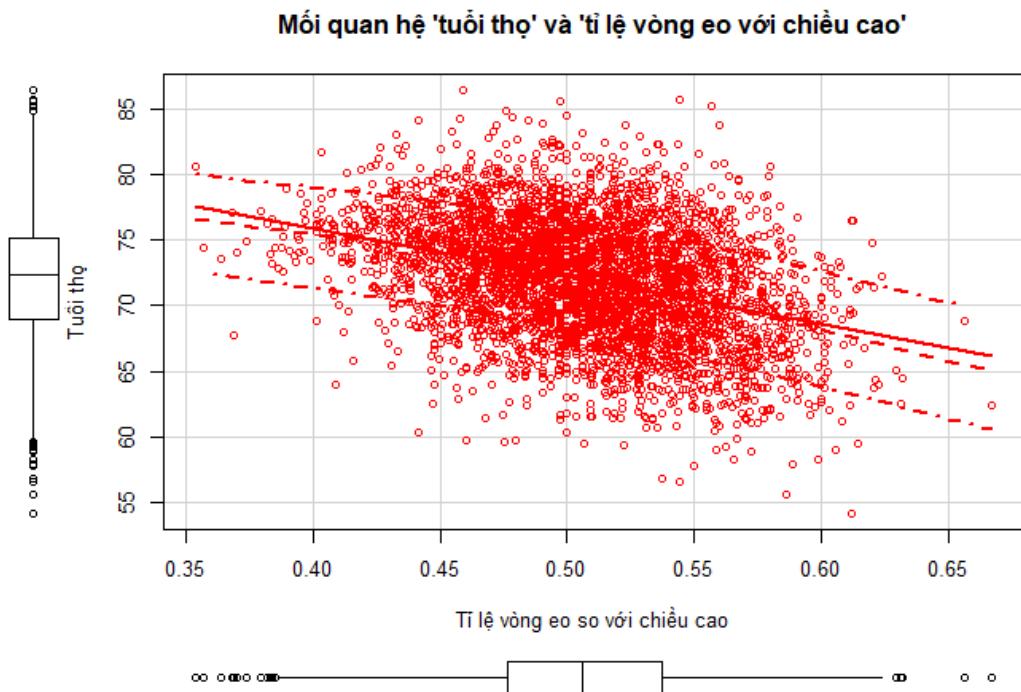
Biểu đồ tương quan scatterplot với thư viện car

Chuyên nghiệp hơn một chút thì sử dụng hàm scatterplot trong thư viện “car” của R. Đoạn code bên dưới thực hiện cài thư viện “car” nếu máy bạn chưa có rồi đọc dữ liệu, vẽ biểu đồ:

```
packages <- c('car')
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

df =
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
header = T)
df$whtr = df$waist / df$height

library(car)
scatterplot(df$life ~ df$whtr, col = 'red', ylab = "Tuổi thọ", xlab =
"Ti lệ vòng eo so với chiều cao", main = "Mối quan hệ 'tuổi thọ' và
'ti lệ vòng eo với chiều cao'")
```

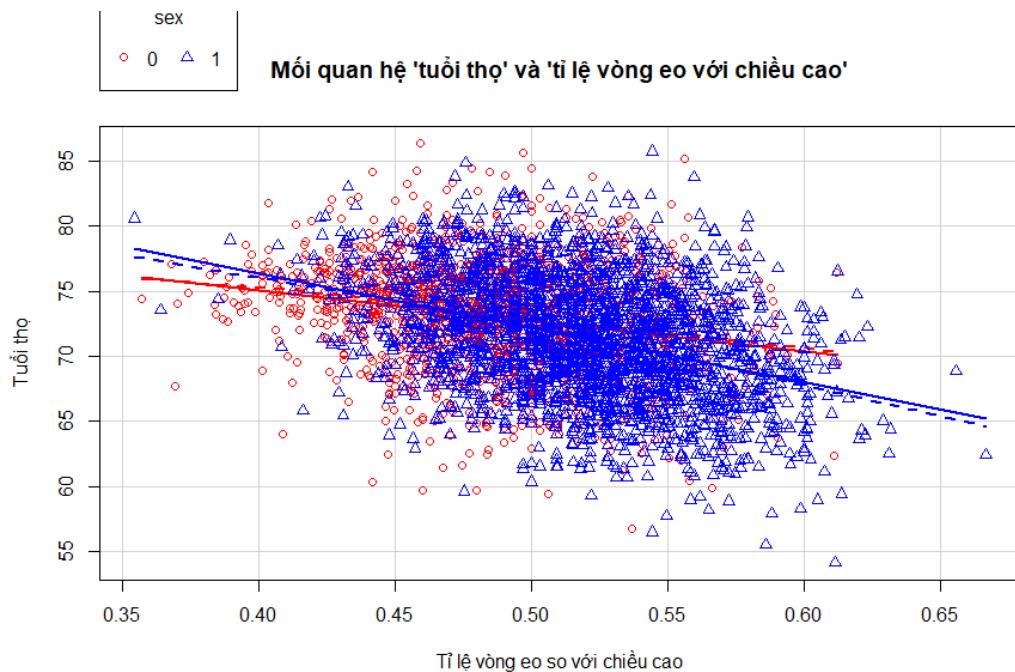


Biểu đồ tương quan có phân nhóm

Để phân nhóm theo giới tính thì thêm cú pháp “| sex” (dấu | là kí tự sọc đứng, phía trên phím Enter) như lệnh bên dưới. Cú pháp sử dụng dấu sọc đứng (broken bar) kết tiếp là một hoặc nhiều biến gọi là cú pháp tương tác.

```
scatterplot(df$life ~ df$whtr | sex, col = c('red', 'blue'), ylab = "Tuổi thọ", xlab = "Tỉ lệ vòng eo so với chiều cao", main = "Mối quan hệ 'tuổi thọ' và 'tỉ lệ vòng eo với chiều cao'")
```

Chạm tới AI trong 10 ngày



Biểu đồ tương quan đa biến

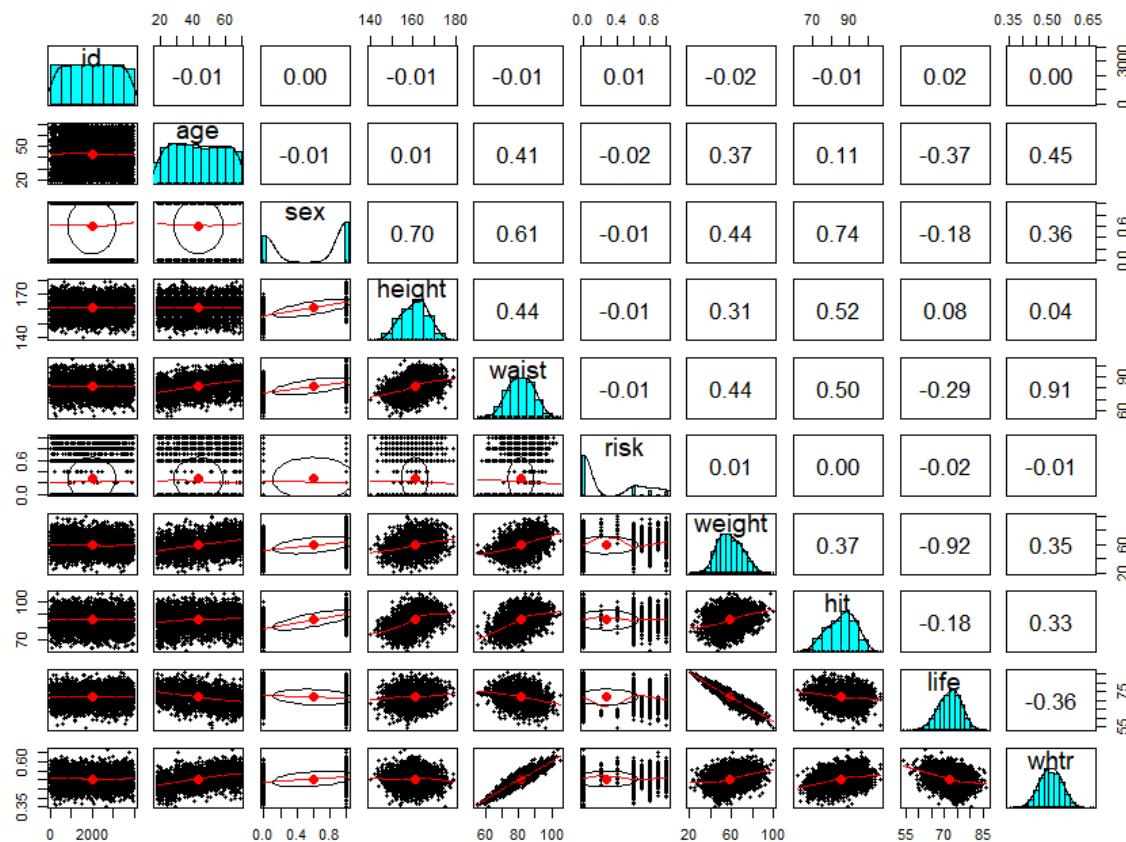
Scatterplot ở trên chỉ cung cấp biểu đồ thể hiện tương quan giữa 2 biến.

Phức tạp hơn một chút, nên chúng ta muốn nhìn sự tương quan của nhiều biến trong một bộ dữ liệu thì dùng lệnh **pairs.panels** trong thư viện **psych**.

```
packages <- c('dplyr', 'psych')
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

library(dplyr)
# Lệnh bên dưới loại bỏ cột id
df = df %>% select(-one_of(c('id')))
library(psych)
pairs.panels(df)
```

Chạm tới AI trong 10 ngày



Biểu đồ dữ liệu theo thời gian

Để minh họa về biểu đồ theo thời gian thì tôi lấy dữ liệu chứng khoán Việt Nam làm ví dụ. Cụ thể là mã nguồn R bên dưới vẽ biểu đồ của chỉ số VNIndex cao nhất trong mỗi ngày từ năm 2000 đến thời điểm viết phần này.

```
packages <- c('zoo', 'ggfortify')

if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

df =
read.csv('https://thachln.github.io/datasets/vnindex_20200424.txt',
header = T)

# Thêm cột strDate bằng cách lấy dữ liệu cột "X.DTYYYYMMDD." chuyển
# thành kiểu kí tự (chuỗi).
data$strDate = as.character(data$X.DTYYYYMMDD.)

# Thêm cột data bằng cách lấy dữ liệu cột "strDate" vừa thêm chuyển
# thành kiểu ngày bằng hàm as.Date(strDate, '%Y%m%d')
```

Chạm tới AI trong 10 ngày

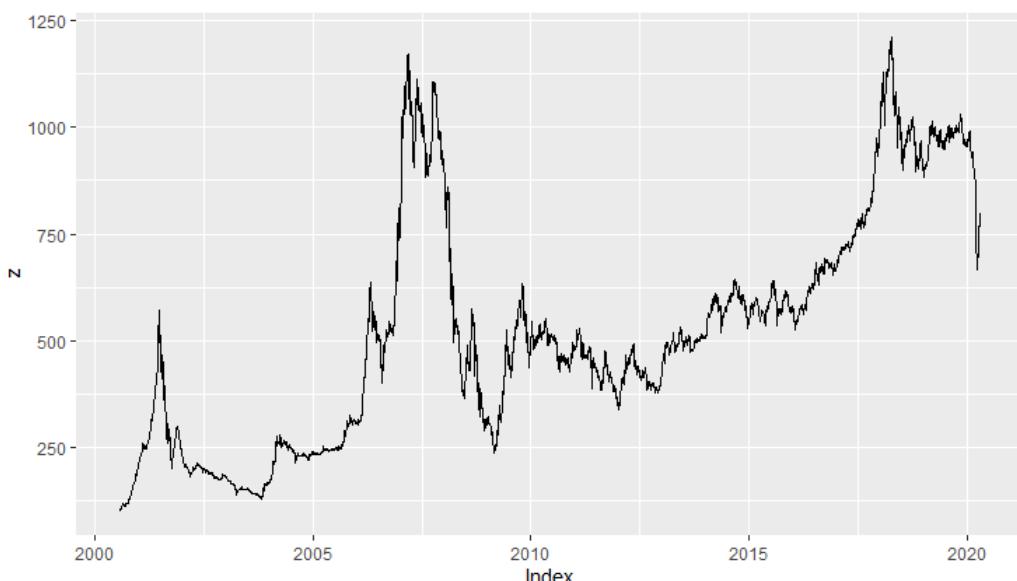
```
data$date = as.Date(data$strDate, format = '%Y%m%d')

library(zoo)

head(data$X.High.)

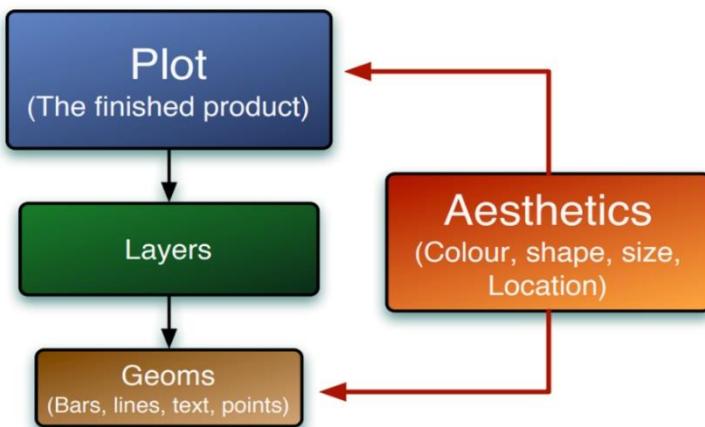
# Tạo dữ liệu x theo thời gian của giá trị cao nhất của VNIndex
z = zoo(x = cbind(data$X.High.), order.by = data$date)

# Vẽ biểu đồ VNIndex cao nhất theo ngày
library(ggfortify)
autoplot(z)
```



Vẽ biểu đồ với ggplot2

Hadley Wickham, người New Zealand, là tác giả của ggplot2. Giới khoa học đánh giá ggplot2 như là cuộc cách mạng trong việc hiển thị dữ liệu. Phần này sẽ giúp các bạn hiểu được triết lý ggplot2 để tạo ra các biểu đồ chất lượng cao.



Hadley Wickham đưa ra triết lý biểu đồ được xem như là bức tranh hoàn chỉnh. Để vẽ bức tranh thì đầu tiên phải bày ra canvas, giống như khung vải cho bức tranh. Thứ hai là phải có ý tưởng vẽ cái gì. Thứ ba là chọn màu sắc và ánh sáng, một yếu tố quan trọng.

Cách xây dựng biểu đồ giống như là vẽ bức tranh. Cụ thể gồm các thành phần sau:

| Thành phần | Ghi chú | | | | | | | | | | | | | | | | |
|--|--|------------------|---------|--------------------|-----------------|------------------|-------------|--------------|---------------|--------------------|--------------|-------------------|---------------------|---------------------|----------------|----------------------|-----------------------------|
| Layer - Tầng | <p>Biểu đồ gồm nhiều layer:</p> <ul style="list-style-type: none"> Layer 1: định nghĩa biến phân tích cần vẽ: aes() Layer 2: thể loại biểu đồ: geom_xxx() Layer 3: màu, kích thước, v.v...: theme_xxx() ... | | | | | | | | | | | | | | | | |
| Geometric objects (geom) – đối tượng hình học | <table> <thead> <tr> <th>Geometric object</th> <th>Ý nghĩa</th> </tr> </thead> <tbody> <tr> <td>geom_histogram(x=)</td> <td>Biểu đồ phân bố</td> </tr> <tr> <td>geom_boxplot(x=)</td> <td>Biểu đồ hộp</td> </tr> <tr> <td>geom_bar(x=)</td> <td>Biểu đồ thanh</td> </tr> <tr> <td>geom_point(x=, y=)</td> <td>Biểu đồ điểm</td> </tr> <tr> <td>geom_line(x=, y=)</td> <td>Biểu đồ đường thẳng</td> </tr> <tr> <td>geom_smooth(x=, y=)</td> <td>Biểu đồ smooth</td> </tr> <tr> <td>geom_density(x=, y=)</td> <td>Biểu đồ với xác suất phân b</td> </tr> </tbody> </table> | Geometric object | Ý nghĩa | geom_histogram(x=) | Biểu đồ phân bố | geom_boxplot(x=) | Biểu đồ hộp | geom_bar(x=) | Biểu đồ thanh | geom_point(x=, y=) | Biểu đồ điểm | geom_line(x=, y=) | Biểu đồ đường thẳng | geom_smooth(x=, y=) | Biểu đồ smooth | geom_density(x=, y=) | Biểu đồ với xác suất phân b |
| Geometric object | Ý nghĩa | | | | | | | | | | | | | | | | |
| geom_histogram(x=) | Biểu đồ phân bố | | | | | | | | | | | | | | | | |
| geom_boxplot(x=) | Biểu đồ hộp | | | | | | | | | | | | | | | | |
| geom_bar(x=) | Biểu đồ thanh | | | | | | | | | | | | | | | | |
| geom_point(x=, y=) | Biểu đồ điểm | | | | | | | | | | | | | | | | |
| geom_line(x=, y=) | Biểu đồ đường thẳng | | | | | | | | | | | | | | | | |
| geom_smooth(x=, y=) | Biểu đồ smooth | | | | | | | | | | | | | | | | |
| geom_density(x=, y=) | Biểu đồ với xác suất phân b | | | | | | | | | | | | | | | | |
| Aesthetics (aes) – thẩm mỹ | <p>Data: dữ liệu</p> <p>Color: màu sắc</p> <p>Size: kích thước</p> | | | | | | | | | | | | | | | | |

Shape: Hình dạng**Location: vị trí của biểu đồ**

Một biểu đồ đầu tiên có **nhiều layer**. Tiếp theo là **đối tượng hình học**. Đối tượng hình học có thể là bar, line, point, v.v... Tiếp theo là yếu tố thứ ba liên quan đến **thẩm mỹ** (màu sắc, kích thước, hình dạng, vị trí). Ba yếu tố tên kết hợp với các **biến dữ liệu** sẽ cho ra một biểu đồ như là một bức tranh.

Văn phạm và thành tố của ggplot2:

Để vẽ biểu đồ thì đầu tiên là phải có **dữ liệu** và lựa chọn **biến số**. Tiếp theo là phải suy nghĩ ra **hình thức thể hiện**. Tiếp theo là ghi cái **nhãn** cho **trục x và y**. Tiếp theo nữa là thiết lập cái theme (cánh nền). Cuối cùng là có thể hoán chuyển như dữ liệu % thành log; hoặc hoán chuyển trực tung và trung hoành để thể hiện mối liên quan.

Để minh họa thì chúng ta quay lại bộ dữ liệu nghiên cứu sức khỏe. Chúng ta đọc dữ liệu vào và tính cột tỉ số eo mông (whtr). Tiếp theo là chuyển cột sex thành kiểu yếu tố do biến sex gồm hai giá trị 0, 1 sẽ được lệnh read.csv hiểu là integer. Bạn có thể kiểm tra lại bằng lệnh class (df\$sex) .

```
packages <- c('ggplot2')
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

library(ggplot2)

df =
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
header = T)
df$whtr = df$waist / df$height
df$sex = as.factor(df$sex)

head(df)
```

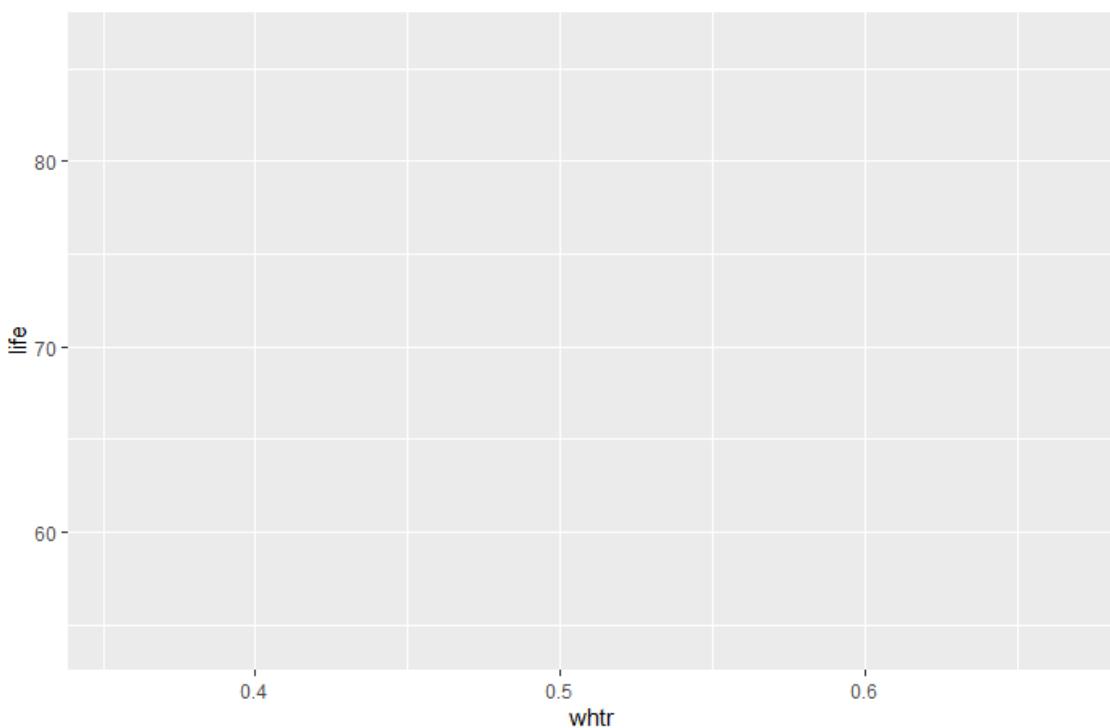
| | id | age | sex | height | waist | risk | weight | hit | life | whtr |
|---|----|-----|-----|--------|-------|------|--------|-----|-------|-----------|
| 1 | 1 | 23 | 0 | 148 | 69 | 0.0 | 38 | 77 | 77.50 | 0.4662162 |
| 2 | 2 | 26 | 1 | 171 | 82 | 0.0 | 57 | 95 | 75.37 | 0.4795322 |
| 3 | 3 | 66 | 1 | 164 | 86 | 0.0 | 77 | 89 | 66.09 | 0.5243902 |
| 4 | 4 | 55 | 1 | 170 | 89 | 0.0 | 73 | 90 | 69.59 | 0.5235294 |
| 5 | 5 | 30 | 0 | 154 | 76 | 0.0 | 59 | 83 | 70.01 | 0.4935065 |
| 6 | 6 | 27 | 1 | 173 | 79 | 0.6 | 66 | 91 | 72.76 | 0.4566474 |

Chúng ta cần thể hiện mối liên quan giữa tỉ số eo mông whtr và tuổi thọ life. Như vậy biến x = whtr, y = life.

Lệnh sau sẽ chuẩn bị canvas gồm trục x là whtr và y là life.

Chạm tới AI trong 10 ngày

```
p = ggplot(data=df, aes(x=whtr, y=life, fill=sex, color=sex))  
p
```

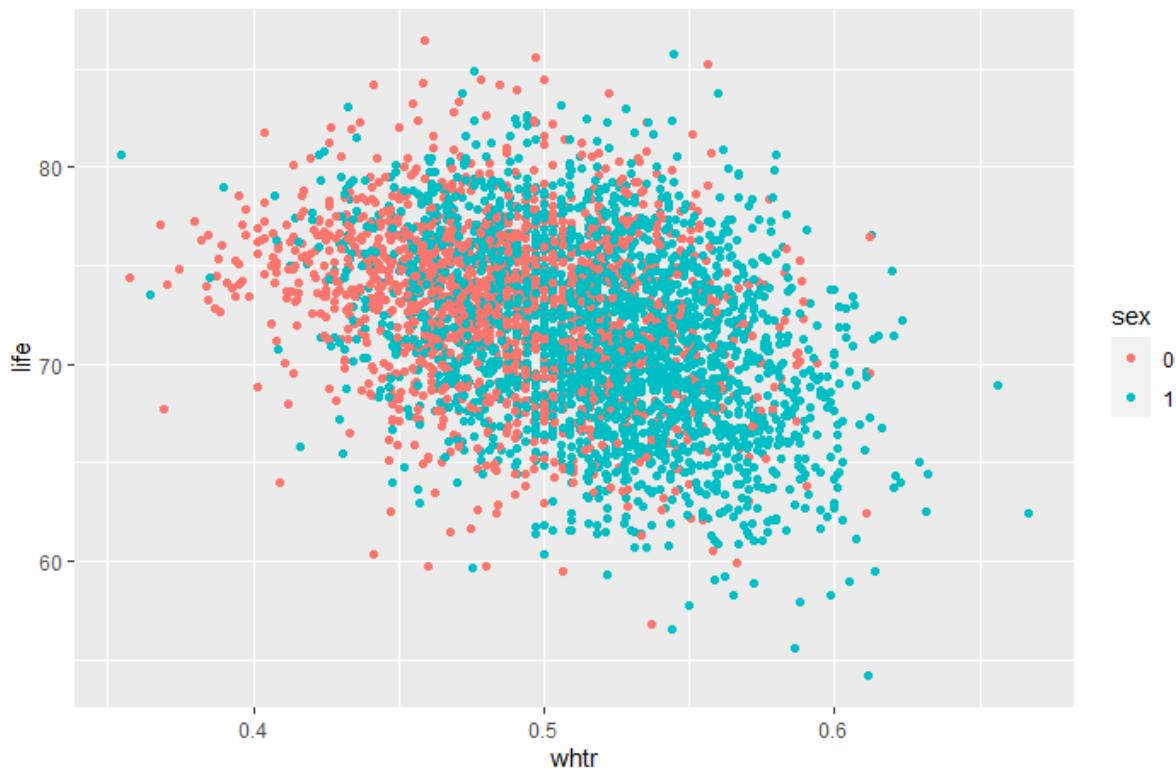


Bước này giống như họa sĩ căng khung vải chuẩn bị cho bức tranh.

Bước kế tiếp là thể hiện đối tượng hình học (geometry, xem danh sách các geometry ở trang trước). Ví dụ chọn đối tượng hình học là “điểm” bằng cách ghép đối tượng p với geom_point() bằng phép toán cộng.

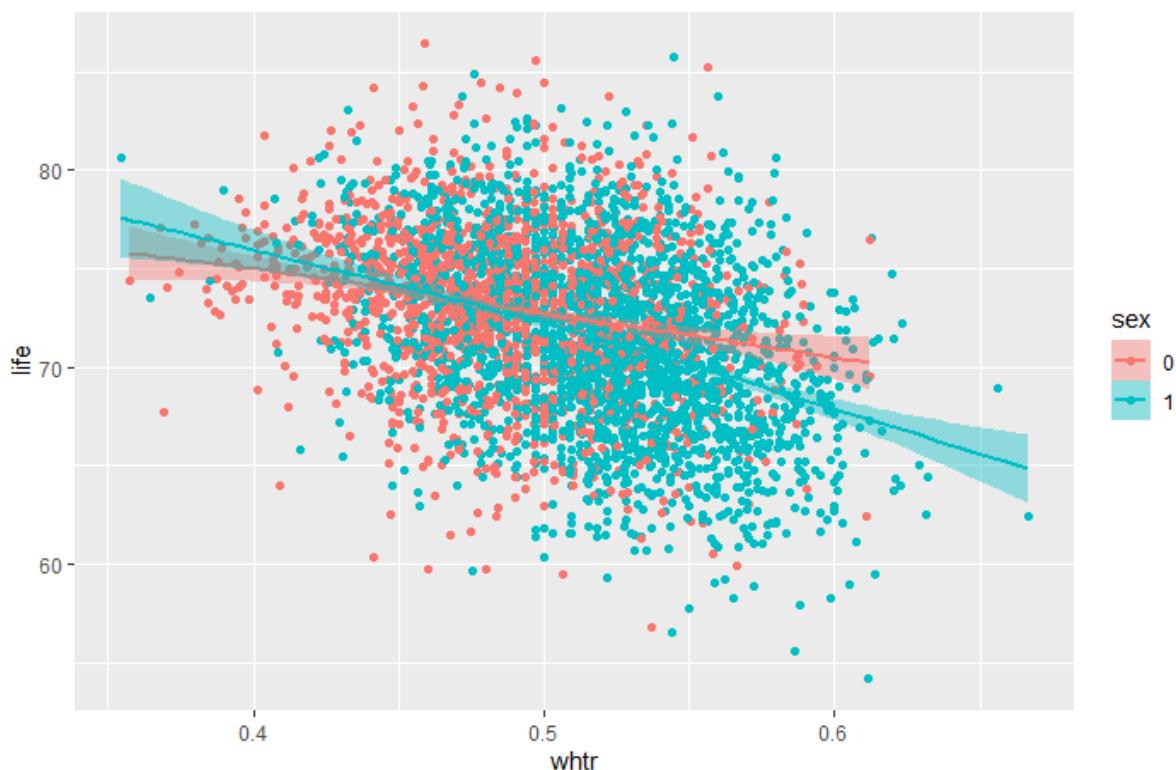
```
p = p + geom_point()  
p
```

Chạm tới AI trong 10 ngày



Tiếp theo, thể hiện thêm biểu đồ bằng mô hình bằng cách vẽ chòng đối tượng `geom_smooth()` như sau:

```
p = p + geom_smooth()  
p
```



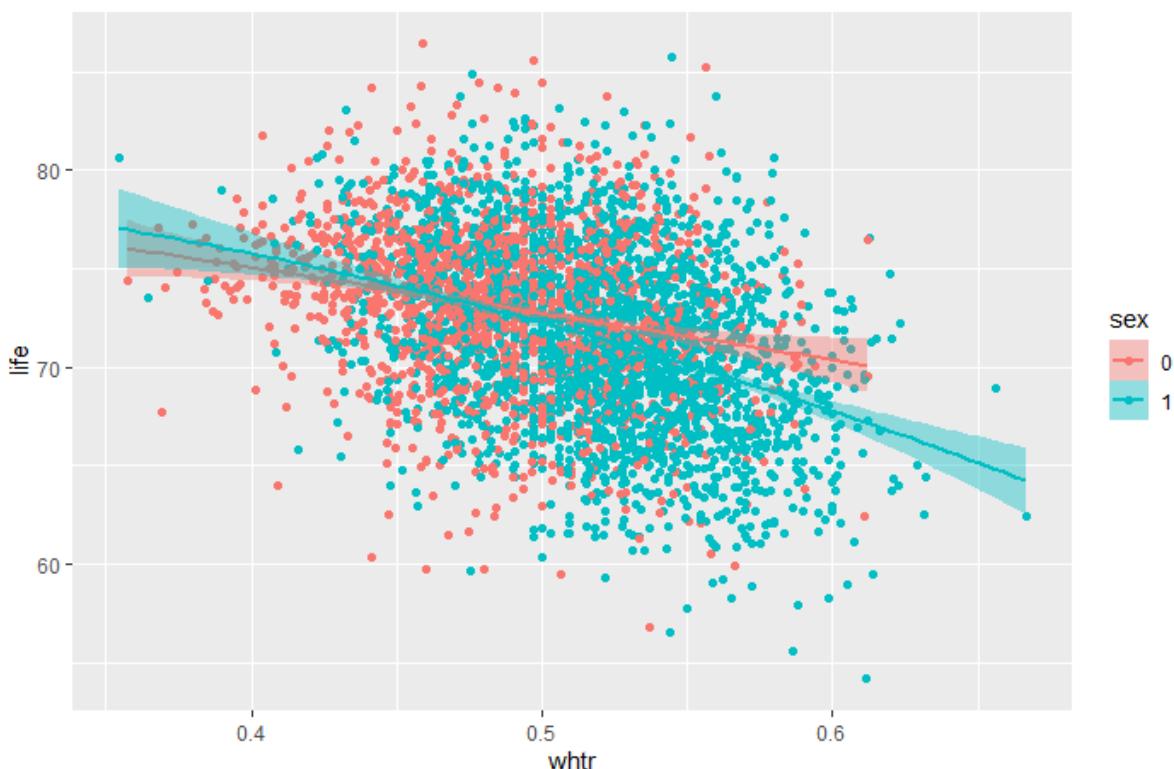
Chạm tới AI trong 10 ngày

Hình trên có vẻ mối tương quan giữa tỉ số eo mông và tuổi thọ theo phương trình tuyến tính. Vùng mờ mờ xung quanh đường “tuyến tính” (giống đám mây) là khoảng tin cậy 95%. KTC 95% cho biết khoảng dao động. Với tỉ số eo mông dưới 0.4 và trên 0.6 thì khoảng tin cậy 95% rộng. Lý do là số đối tượng nghiên cứu ít, nên độ lệch chuẩn lớn. Vì thế khoảng tin cậy 95% cũng rộng ra.

Đường kẻ liền nét là mô hình tuyến tính bậc 1. Nếu thử mô hình tuyến tính bậc 2 thì sao. Chúng ta bổ sung thêm tham số cho geom_smooth như sau:

```
p = p + geom_smooth(method = "lm", formula = y~x + I(x^2))  
p
```

lm là linear model (linear đọc là lin-near, chứ không phải là lai-near)



Quan sát kỹ thì mô hình tuyến tính bậc 2 cũng không khác mấy. Chỉ có nam (sex = 1) thì khi whtr > 0.63 thì tuổi thọ thấp xuống một chút.

Bước 3: Thêm layer liên quan đến nhãn cho trục tung và trục hoành.

Nếu trục tung, trục hoành là biến rời rạc (discrete) dùng dùng hàm scale như sau:

```
scale_x_discrete(name='xxx', limits=c('A', 'B', 'C')  
scale_y_discrete(name='xxx', limits=c('X', 'Y', 'Z'))
```

Nếu trục tung, trục hoành là biến liên tục (continuous) dùng dùng hàm scale như sau:

Chạm tới AI trong 10 ngày

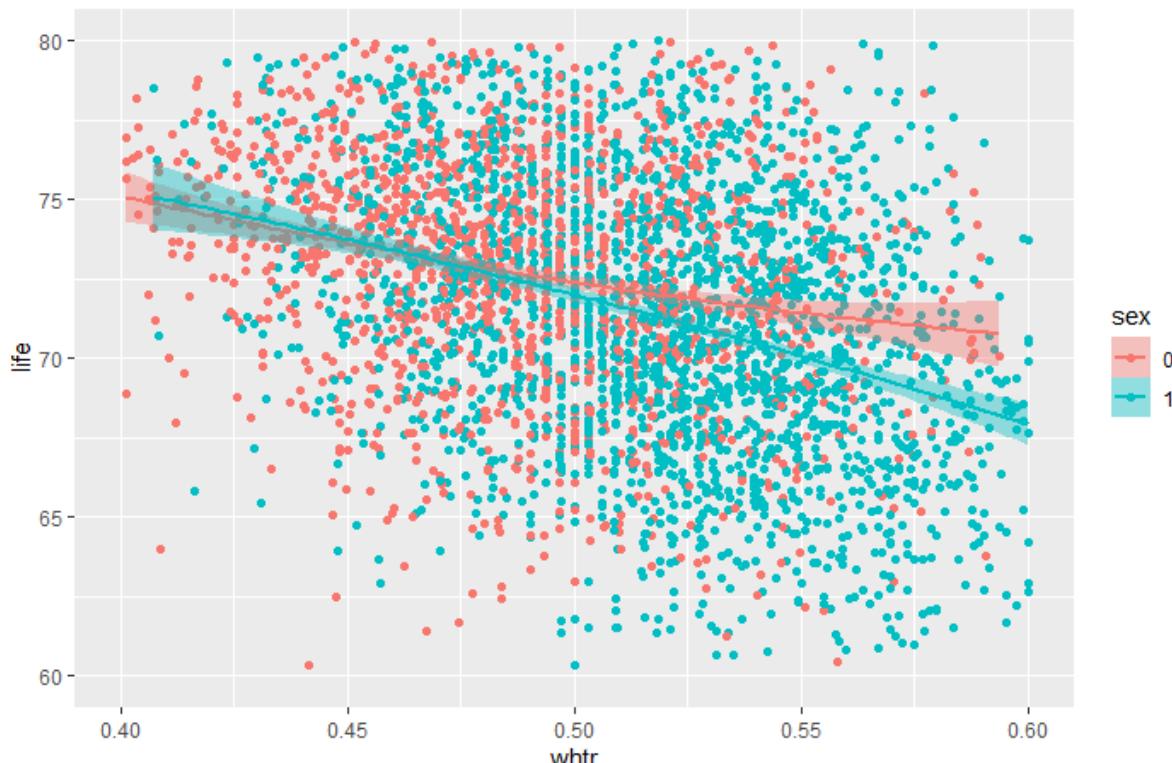
```
scale_x_continuous(name='xxx', limits=c(lower, upper),
break = c(1, 2, 5, 10))
scale_y_continuous(name='xxx', limits=c(lower, upper),
break = c(0, 5, 10, 20))
```

Cụ thể quay lại biểu đồ ở trên, thêm layer scale và code viết lại toàn bộ như sau:

```
library(ggplot2)

df =
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
header = T)
df$whtr = df$waist / df$height

head(df)
df$sex = as.factor(df$sex)
p = ggplot(data=df, aes(x=whtr, y=life, fill=sex, color=sex))
p = p + geom_point()
p = p + geom_smooth(method = "lm", formula = y~x + I(x^2))
p = p + scale_x_continuous(name = "whtr", limits = c(0.4, 0.6)) +
scale_y_continuous(name='life', limits = c(60, 80))
p
```



Chạm tới AI trong 10 ngày

Lúc này x được scale từ 0.4 đến 0.6; x được scale từ 60 đến 80 tuổi.

Bổ sung thêm **breaks**:

```
library(ggplot2)

df =
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
header = T)
df$whtr = df$waist / df$height

head(df)
df$sex = as.factor(df$sex)
p = ggplot(data=df, aes(x=whtr, y=life, fill=sex, color=sex))
p = p + geom_point()
p = p + geom_smooth(method = "lm", formula = y~x + I(x^2))
p = p + scale_x_continuous(name = "whtr", breaks=seq(0.4,0.6, 0.025))
+ scale_y_continuous(name='life', breaks = seq(60, 80, 2))
p
```

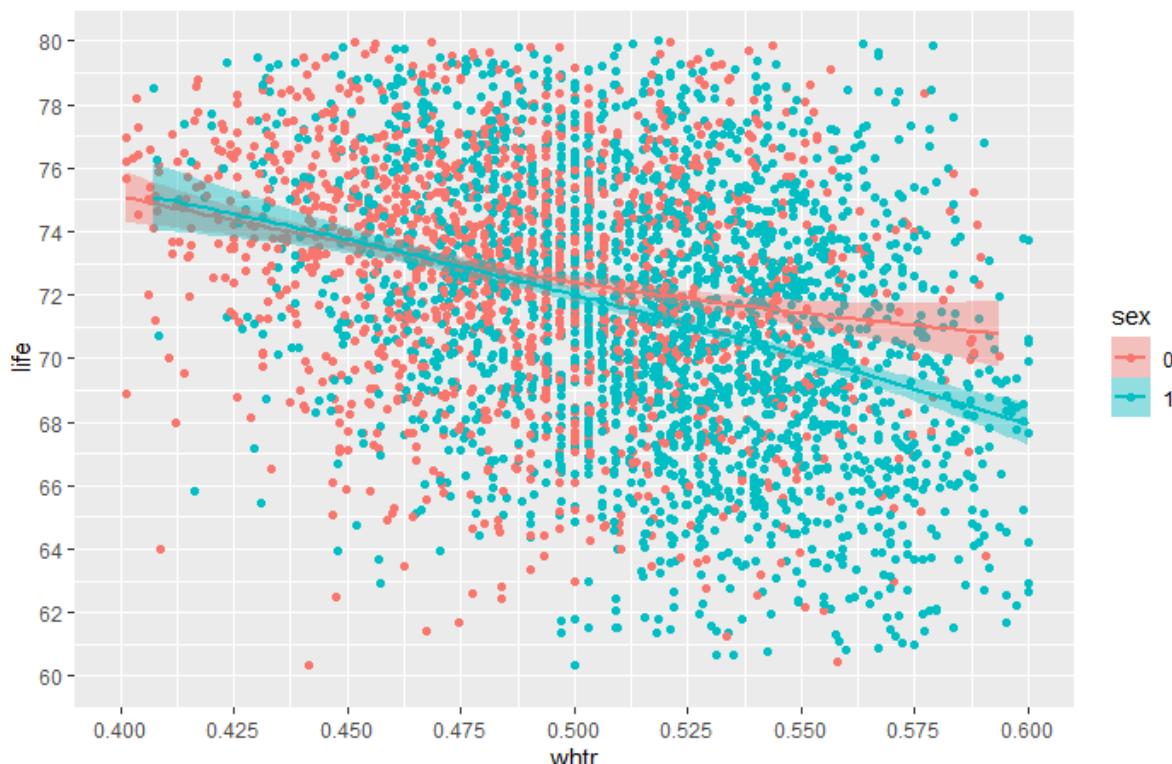


Kết hợp vừa **breaks** và **limits** như sau:

```
library(ggplot2)
```

Chạm tới AI trong 10 ngày

```
df =  
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',  
header = T)  
  
df$whtr = df$waist / df$height  
  
head(df)  
df$sex = as.factor(df$sex)  
p = ggplot(data=df, aes(x=whtr, y=life, fill=sex, color=sex))  
p = p + geom_point()  
p = p + geom_smooth(method = "lm", formula = y~x + I(x^2))  
p = p + scale_x_continuous(name = "whtr", breaks=seq(0.4,0.6, 0.025),  
limits = c(0.4, 0.6)) + scale_y_continuous(name='life', breaks =  
seq(60, 80, 2), limits = c(60, 80))  
p
```



Bước 4: Nhãn và tiêu đề

- Nhãn cho trục x và y:

xlab('Tỉ số eo mông') + **ylab**('Tuổi thọ')

- Tiêu đề

ggtitle("Biểu đồ tương quan giữa tỉ số eo mông và tuổi thọ.")

- Vị trí của legends

theme(legend.position = 'xxx') xxx = 'top', 'bottom', 'none'

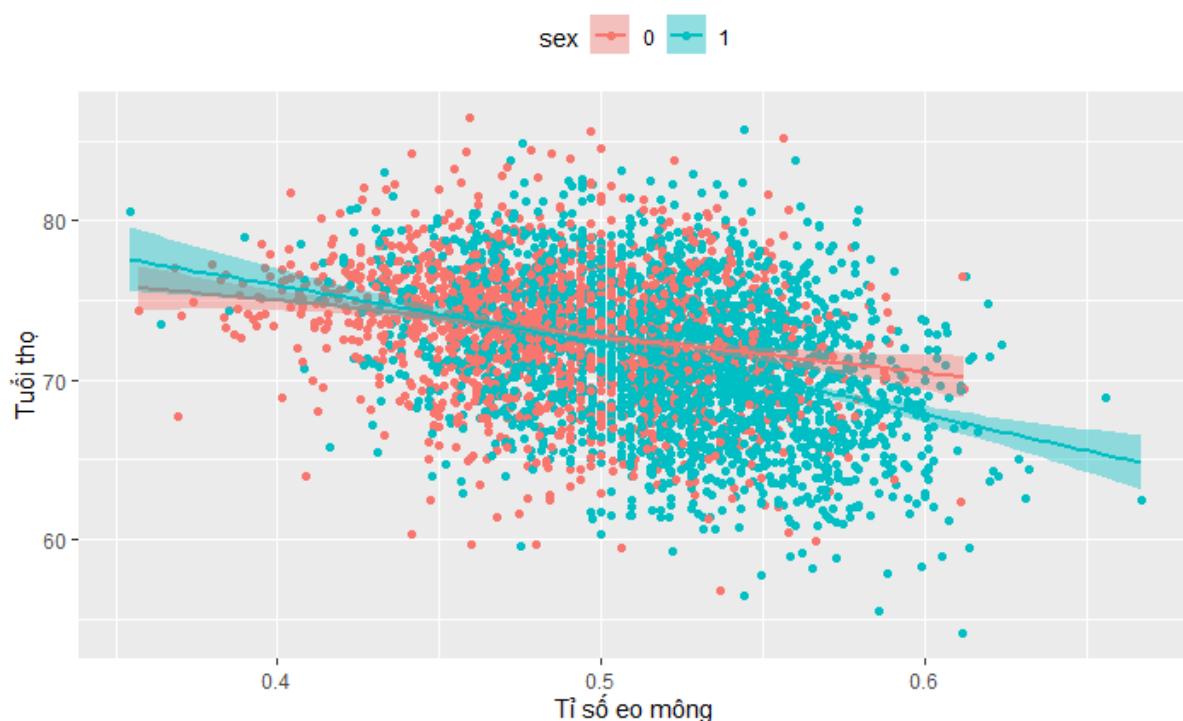
Code mới như sau:

```
library(ggplot2)

df =
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
header = T)
df$whtr = df$waist / df$height

head(df)
df$sex = as.factor(df$sex)
p = ggplot(data=df, aes(x=whtr, y=life, fill=sex, color=sex))
p = p + geom_point() + geom_smooth()
p = p + theme(legend.position = 'top')
p = p + xlab('Tỉ số eo mông') + ylab('Tuổi thọ')
p = p + ggtitle("Biểu đồ tương quan giữa tỉ số eo mông và tuổi thọ.")
p
```

Biểu đồ tương quan giữa tỉ số eo mông và tuổi thọ.



Lần này cái legend giải thích giới tính được chuyển lên trên để không gian vẽ biểu đồ rộng hơn. Ngoài ra, Nhãn của trục x, trục y và tiêu đề của biểu đồ cũng được thêm vào.

Bước 4: Thêm text cho trục x và trục y

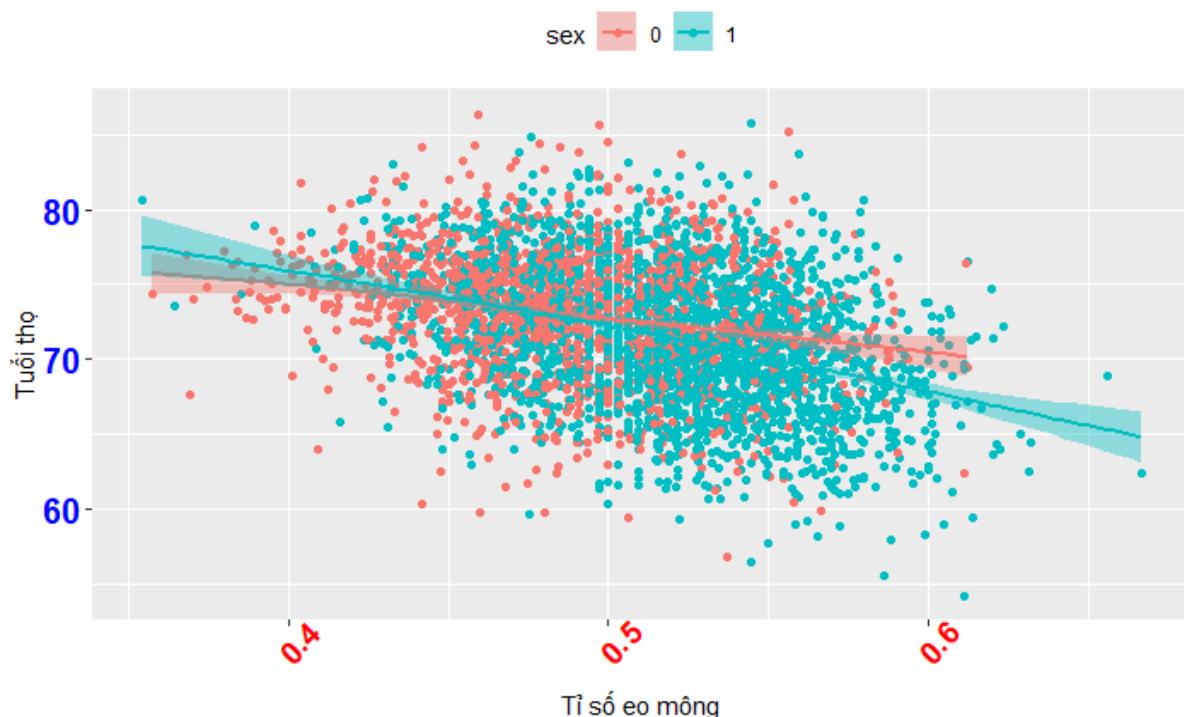
Thêm định dạng màu sắc, kích thước chữ, góc nghiêng chữ cho nhãn của trục x và y như sau:

```
library(ggplot2)

df =
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
header = T)
df$whtr = df$waist / df$height

head(df)
df$sex = as.factor(df$sex)
p = ggplot(data=df, aes(x=whtr, y=life, fill=sex, color=sex))
p = p + geom_point() + geom_smooth()
p = p + theme(legend.position = 'top', axis.text.x = element_text(face = 'bold', color = 'red', size = 14, angle = 45),
axis.text.y = element_text(face = 'bold', color = 'blue', size = 15))
p = p + xlab('Tỉ số eo mông') + ylab('Tuổi thọ')
p = p + ggtitle("Biểu đồ tương quan giữa tỉ số eo mông và tuổi thọ.")
p
```

Biểu đồ tương quan giữa tỉ số eo mông và tuổi thọ.



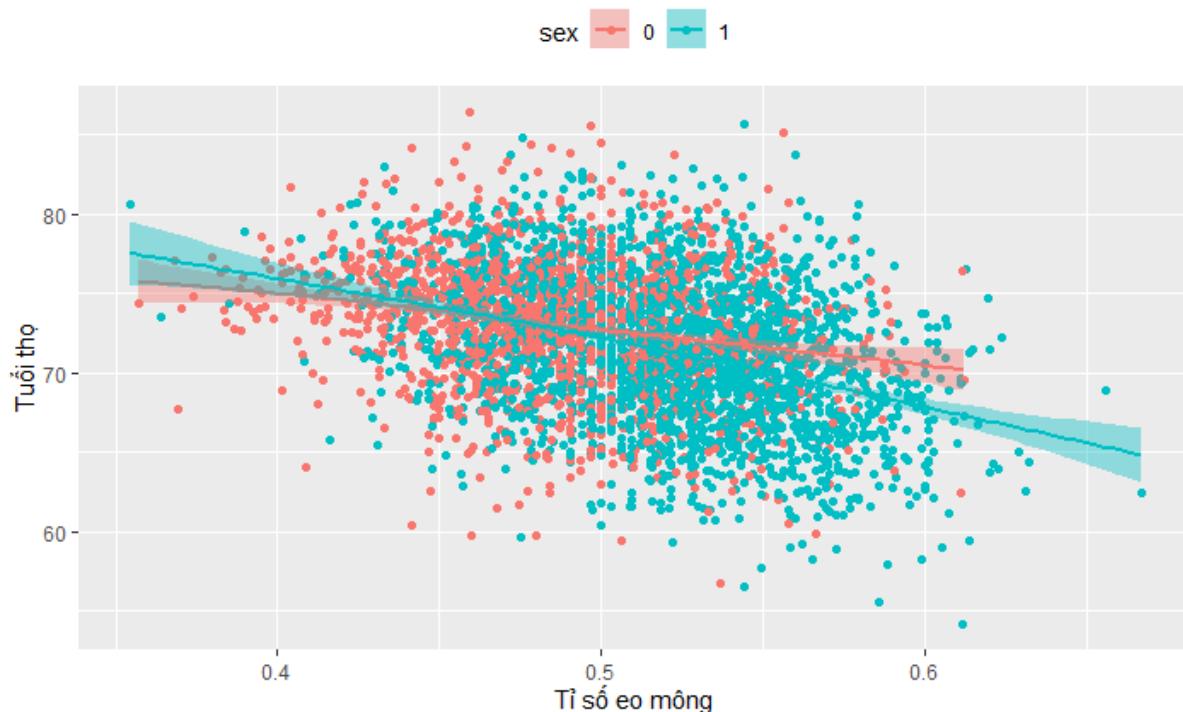
Trang trí thêm cho tiêu đề:

```
packages <- c("ggplot2", "ggthemes")
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}
library(ggplot2)
library(ggthemes)

df =
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
header = T)
df$whtr = df$waist / df$height

head(df)
df$sex = as.factor(df$sex)
p = ggplot(data=df, aes(x=whtr, y=life, fill=sex, color=sex))
p = p + geom_point() + geom_smooth()
p = p + xlab('Tỉ số eo mông') + ylab('Tuổi thọ')
p = p + ggtitle("Biểu đồ tương quan giữa tỉ số eo mông và tuổi thọ.")
p = p + theme(legend.position = 'top', plot.title=element_text(hjust = 0.5, color='red', face = 'bold', size = 15))
p
```

Biểu đồ tương quan giữa tỉ số eo mông và tuổi thọ.

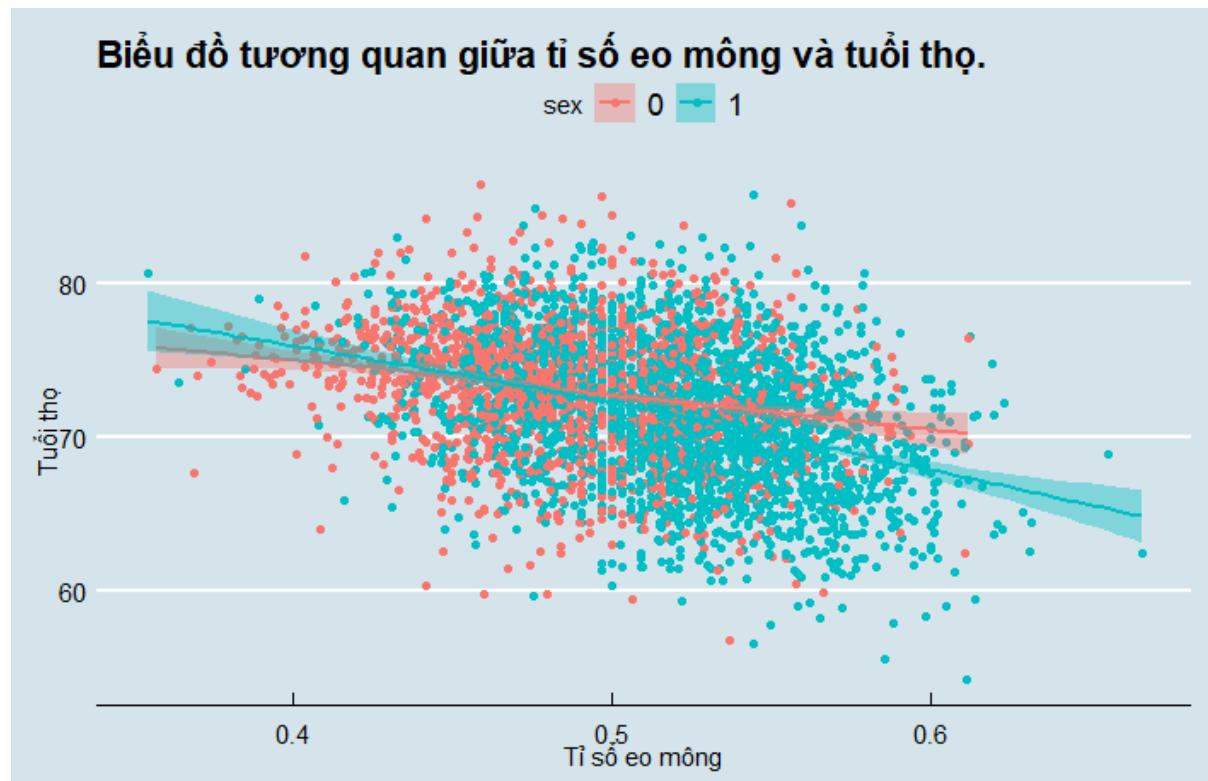


Sử dụng theme có sẵn của tờ báo Time Economist:

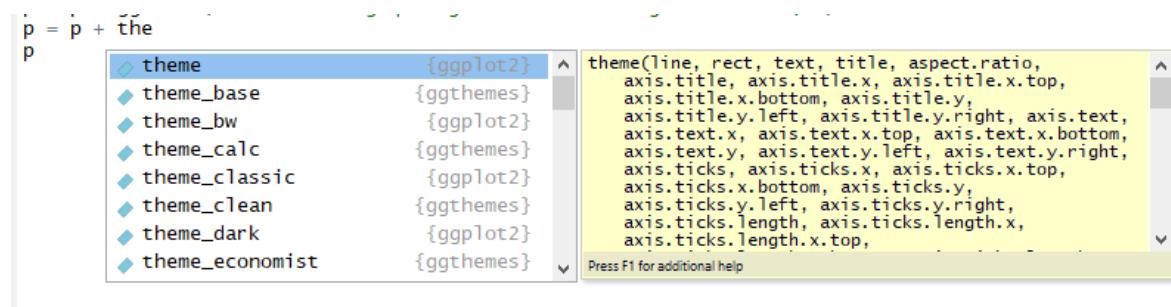
```
packages <- c("ggplot2", "ggthemes")
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}
library(ggplot2)
library(ggthemes)

df =
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
header = T)
df$whtr = df$waist / df$height

head(df)
df$sex = as.factor(df$sex)
p = ggplot(data=df, aes(x=whtr, y=life, fill=sex, color=sex))
p = p + geom_point() + geom_smooth()
p = p + xlab('Tỉ số eo mông') + ylab('Tuổi thọ')
p = p + ggtitle("Biểu đồ tương quan giữa tỉ số eo mông và tuổi thọ.")
p = p + theme_economist()
p
```



Trong Rstudio thì nó đoán là bạn gõ cái gì thì nó sẽ hiện ra các theme tương ứng có sẵn trong thư viện đã cài. Bạn có thể chọn các theme khác để thử thêm như hình bên dưới:



Nếu khó tính hơn nữa thì cần phải nhìn thêm histogram của dữ liệu tỉ số eo mông và tuổi thọ. Dùng thêm phần mềm ggExtra.

```
packages <- c("ggplot2", "ggthemes", "ggExtra")
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

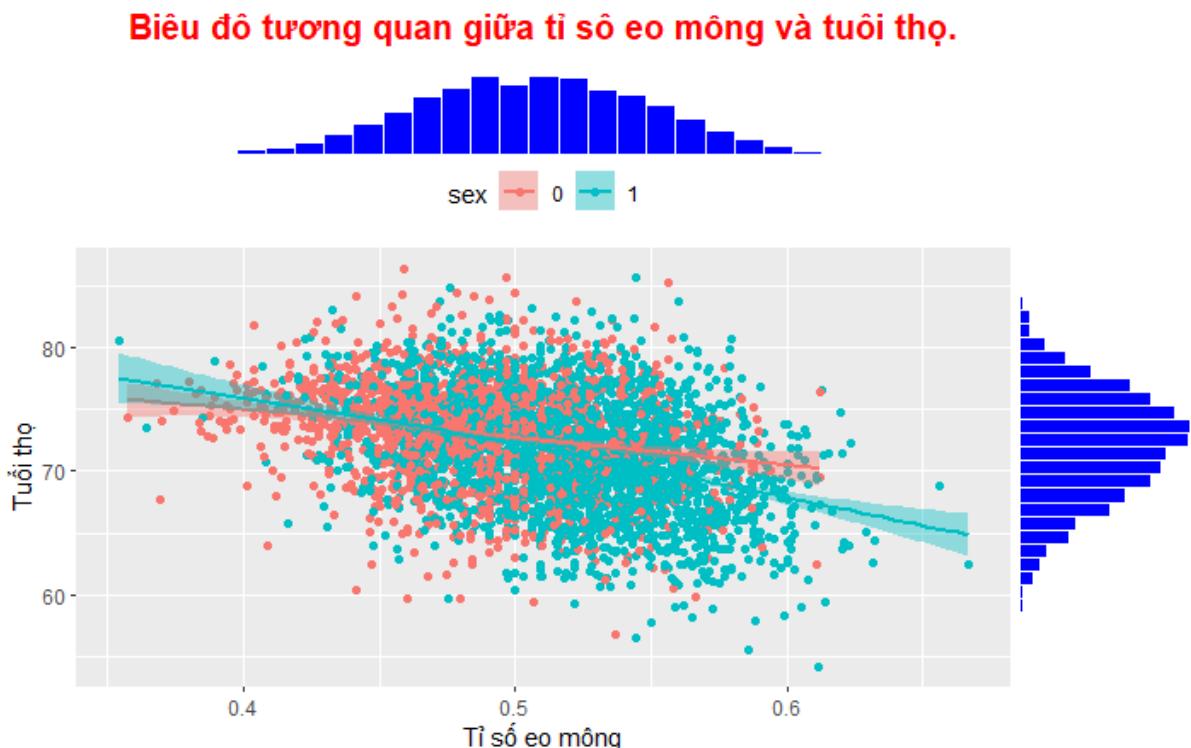
packages <- c("ggplot2", "ggthemes")
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}
```

Chạm tới AI trong 10 ngày

```
library(ggplot2)
library(ggthemes)
library(ggExtra)

df =
read.csv('https://thachln.github.io/datasets/sample_health_vn.csv',
header = T)
df$whtr = df$waist / df$height

head(df)
df$sex = as.factor(df$sex)
p = ggplot(data=df, aes(x=whtr, y=life, fill=sex, color=sex))
p = p + geom_point() + geom_smooth()
p = p + xlab('Tỉ số eo mông') + ylab('Tuổi thọ')
p = p + ggtitle("Biểu đồ tương quan giữa tỉ số eo mông và tuổi thọ.")
p = p + theme(legend.position = 'top', plot.title=element_text(hjust =
0.5, color='red', face = 'bold', size = 15))
ggMarginal(p, type='histogram', col = 'white', fill = 'blue')
```



Chú ý đoạn code trên có sự khác biệt một chút là lệnh **ggMarginal** sẽ vẽ luôn biểu đồ. Trong khi các đoạn code phía trước bạn phải gõ biến b ở dòng cuối cùng để vẽ biểu đồ trong đối tượng p.

Chạm tới AI trong 10 ngày

Một điểm chú ý nữa là title của biểu đồ không hiển thị hết dấu tiếng Việt. Cụ thể là các chữ có 2 dấu như ê, ô, ô thì hiển thị mất một dấu. Đây là vấn đề xử lý chữ của thư viện ggplot2. Chúng ta khắc phục trong một bài khác.

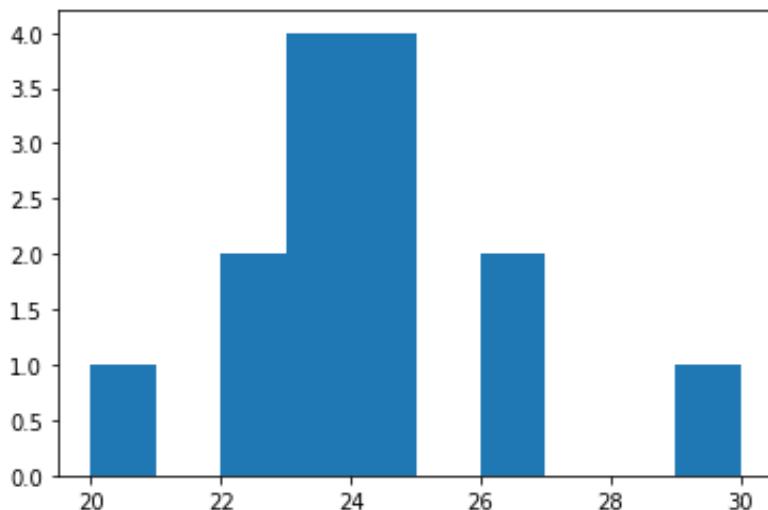
Bài 10: Vẽ biểu đồ trong Python

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:

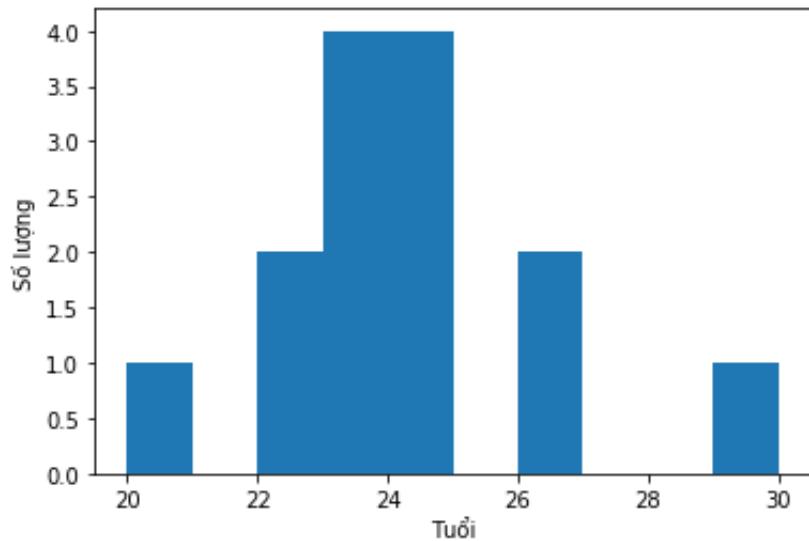
- Trong Python, cú pháp để truy xuất cột dữ liệu trong data frame là dùng cặp dấu ngoặc vuông [], bên trong cặp [] là tên cột bao đóng bởi cặp dấu nháy (nháy đơn hoặc đôi đều được). Ví dụ data: **df["age"]**
- Còn trong R, thì cú pháp là dùng dấu \$ ngay sau tên biến của data frame, tiếp theo là tên của cột mà không cần bao đóng bởi dấu nháy. Ví dụ: **data\$age**

Trang trí thêm trục x và y:

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

Chạm tới AI trong 10 ngày

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



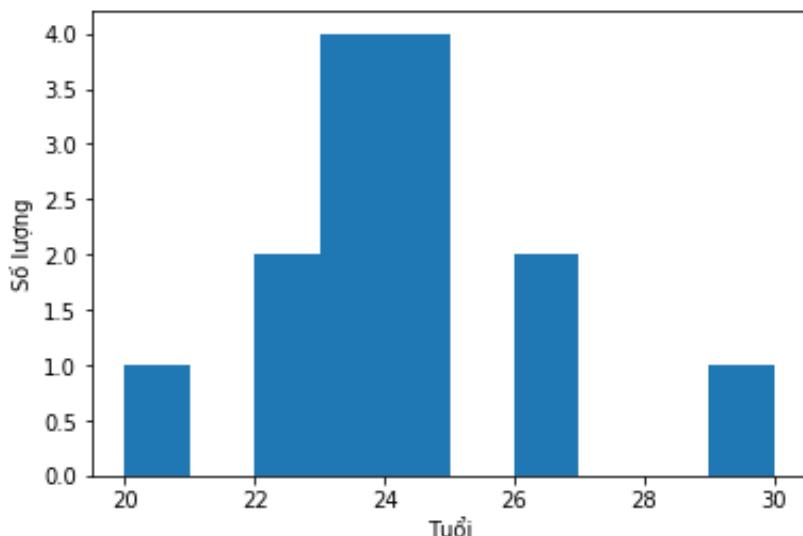
Để trang trí thêm tiêu đề cho biểu đồ thì trong thư viện **matplotlib** cung cấp đối tượng **Figure**.

```
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'])
```

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



Biểu đồ phân bố dữ liệu – Boxplot

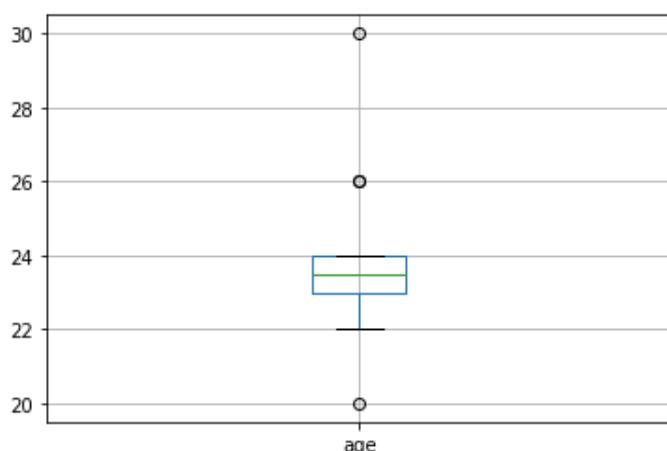
Thư viện pandas ngoài việc cung cấp chức năng đọc dữ liệu từ file vào data frame nó có luôn hàm xem dữ liệu dạng boxplot luôn. Dưới đây là ví dụ:

```
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 data) luôn chúa 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:

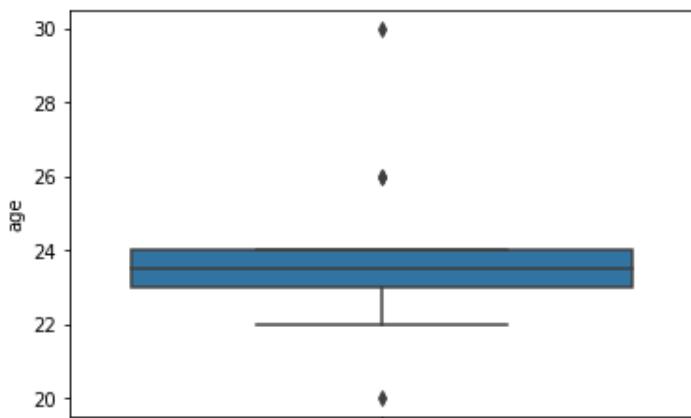
```
import pandas as pd
```

Chạm tới AI trong 10 ngày

```
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

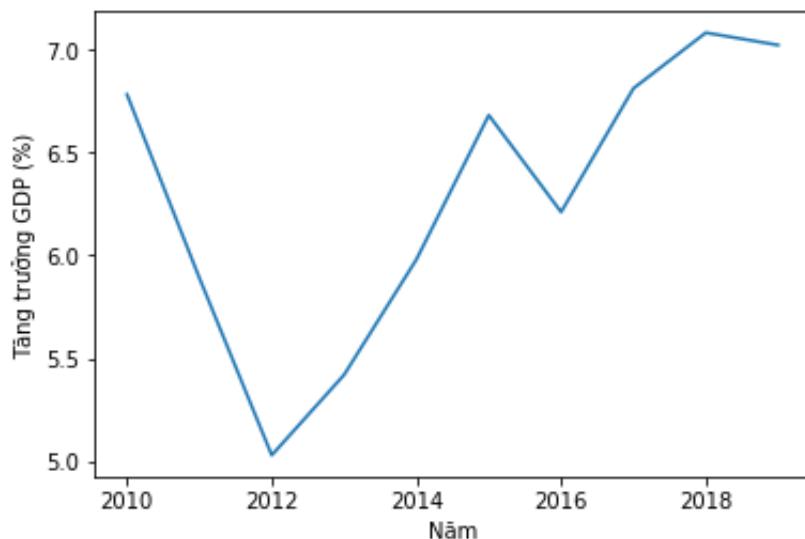
```
import pandas as pd
import matplotlib.pyplot as plt

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.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(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

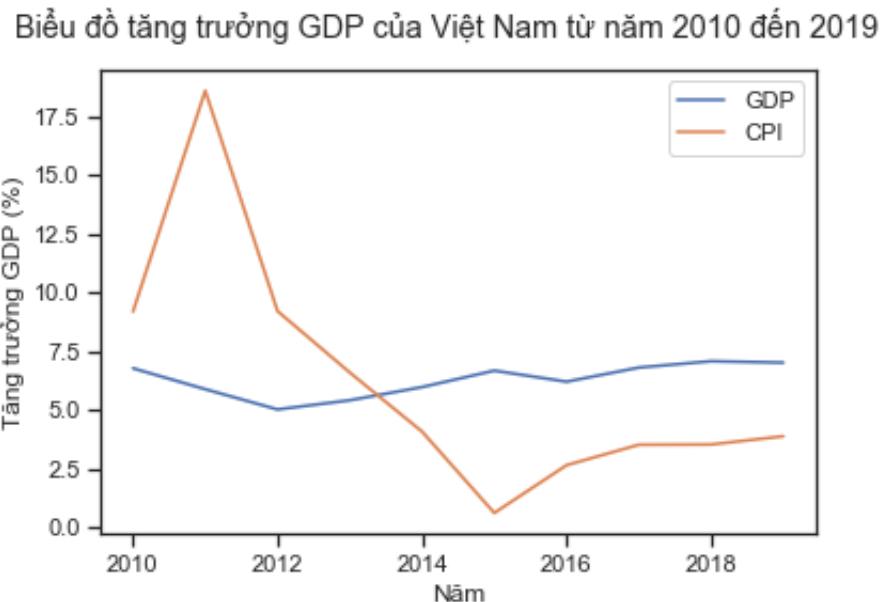
Biểu đồ so sánh - Line Chart – 2 biến

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



Biểu đồ 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 Matlib 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

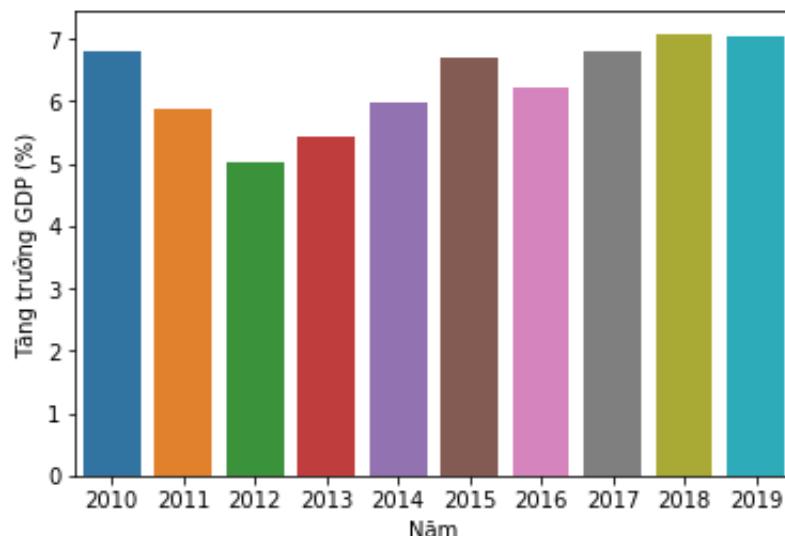
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019]

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

plt.figure().suptitle('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 (%)')
```

Chạm tới AI trong 10 ngày

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



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

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

values = np.asarray(values)
```

Chạm tới AI trong 10 ngày

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

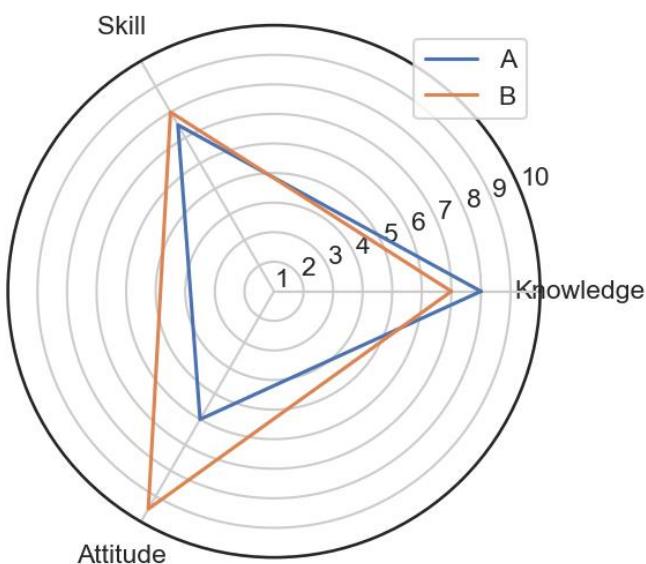
# Create figure
plt.figure(figsize=(4, 4), dpi=150).suptitle('Biểu đồ so sánh tam giác ASK của 2 người.')

# 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.

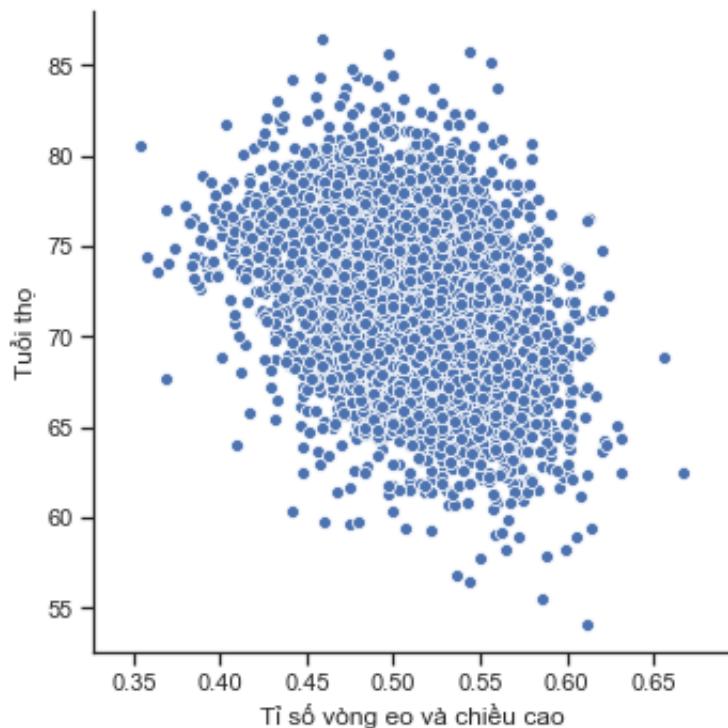


Biểu đồ tương quan đơn giản

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

Chạm tới AI trong 10 ngày

```
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')
```

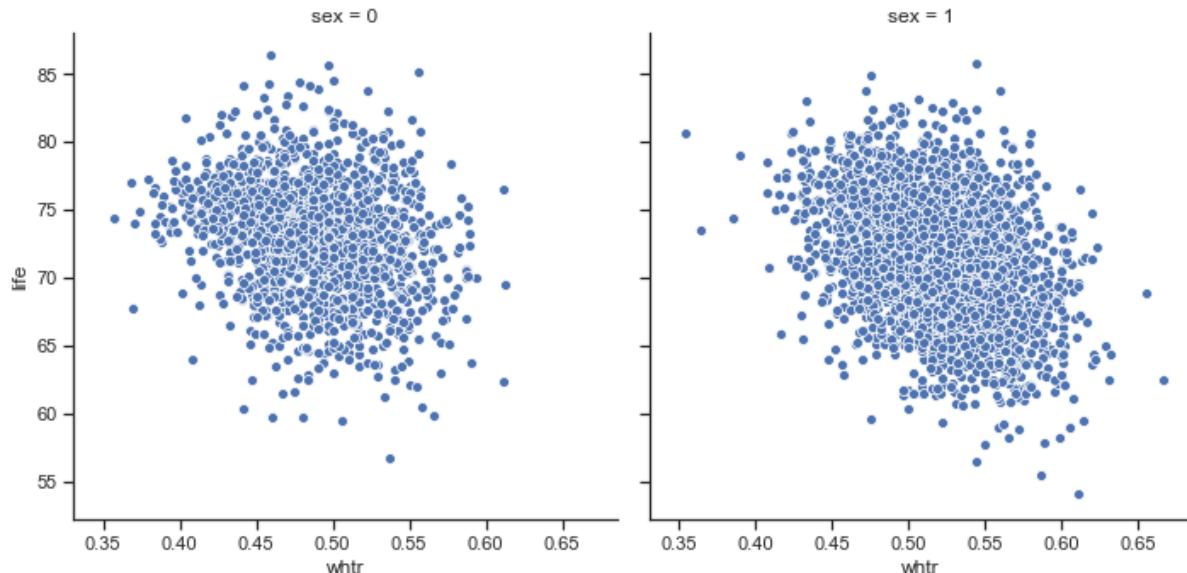


Biểu đồ tương quan có phân nhóm

```
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'")
```

Chạm tới AI trong 10 ngày

```
sns.set(style="ticks")
sns.relplot(x="whtr", y="life", col="sex", data=df)
plt.xlabel('Tí số vòng eo và chiều cao')
plt.ylabel('Tuổi thọ')
```



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.

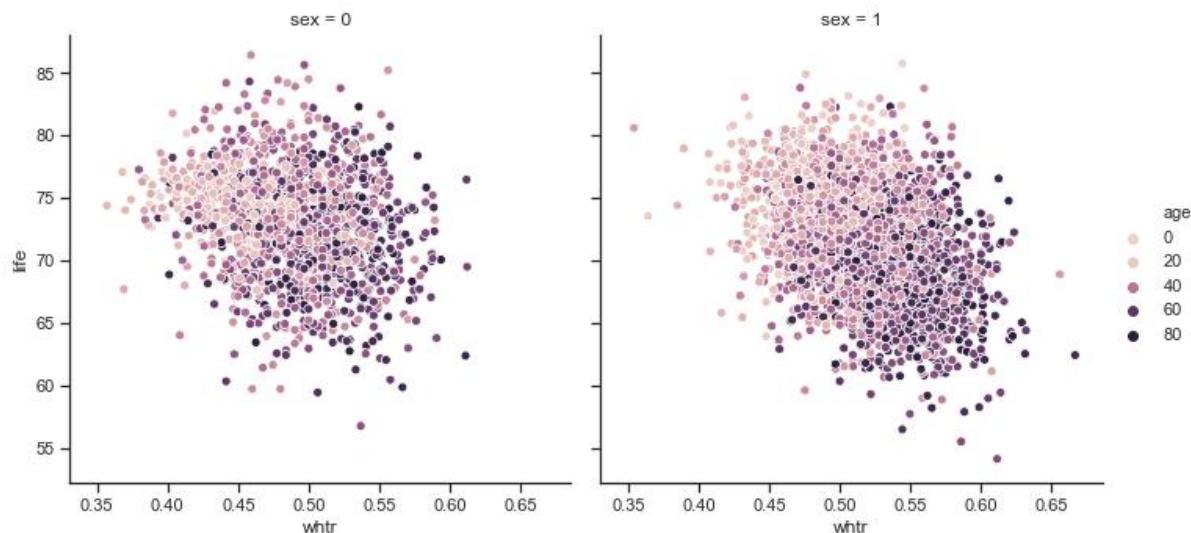
Lệnh bên dưới dùng thêm tham số hue = 'age':

```
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)
```

Chạm tới AI trong 10 ngày



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 tần số 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

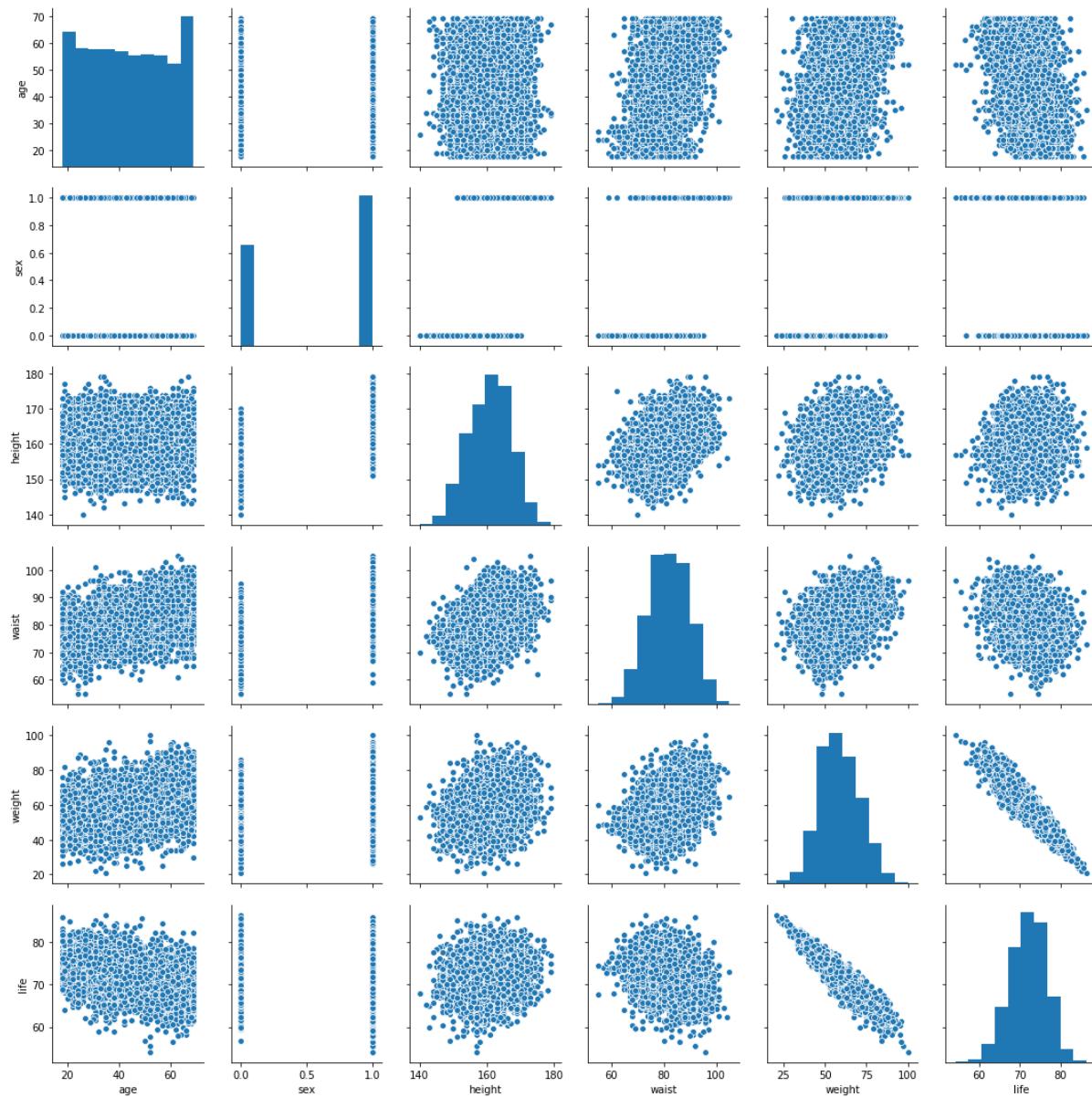
Để vẽ biểu đồ trình bày mối tương quan của các biến trong data frame thì thư viện Seaborn cung cấp hàm **pairplot**.

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

```
import seaborn as sns
import pandas as pd

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)
```

Chạm tới AI trong 10 ngày

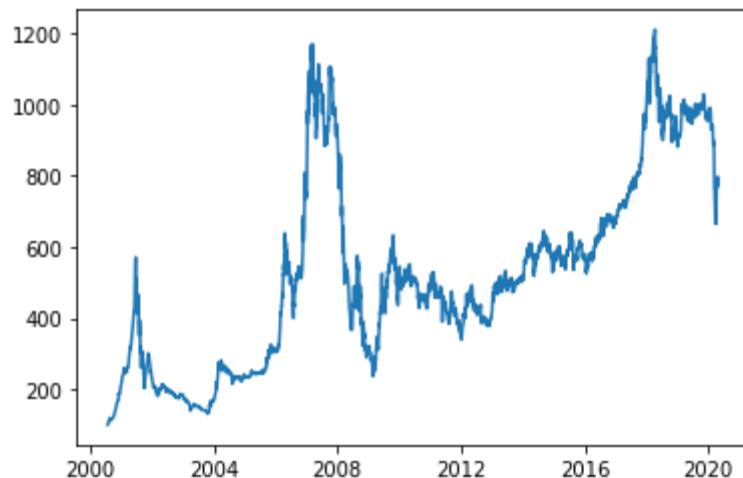


Biểu đồ dữ liệu theo thời gian

```
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 ư 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
```

```
Anaconda Prompt (Anaconda3)
r-purrr-0.3.2          | 430 KB  | #####| 100%
r-memoise-1.1.0         | 45 KB   | #####| 100%
r-digest-0.6.18          | 154 KB  | #####| 100%
r-bh-1.69.0_1            | 10.9 MB | #####| 100%
r-rlang-0.3.4             | 1.1 MB  | #####| 100%
tbb-2018.0.5              | 150 KB  | #####| 100%
m2w64-libxml2-2.9.3       | 2.1 MB  | #####| 100%
r-tibble-2.1.1            | 336 KB  | #####| 100%
r-plogr-0.2.0              | 20 KB   | #####| 100%
conda-4.8.3                | 2.8 MB  | #####| 100%
r-crayon-1.3.4             | 757 KB  | #####| 100%
r-cli-1.1.0                 | 189 KB  | #####| 100%
r-blob-1.1.1                 | 40 KB   | #####| 100%
r-pkgconfig-2.0.2            | 25 KB   | #####| 100%
r-base-3.6.1                  | 55.3 MB | #####| 100%
rpy2-2.9.4                   | 342 KB  | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

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

Chạm tới AI trong 10 ngày

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()

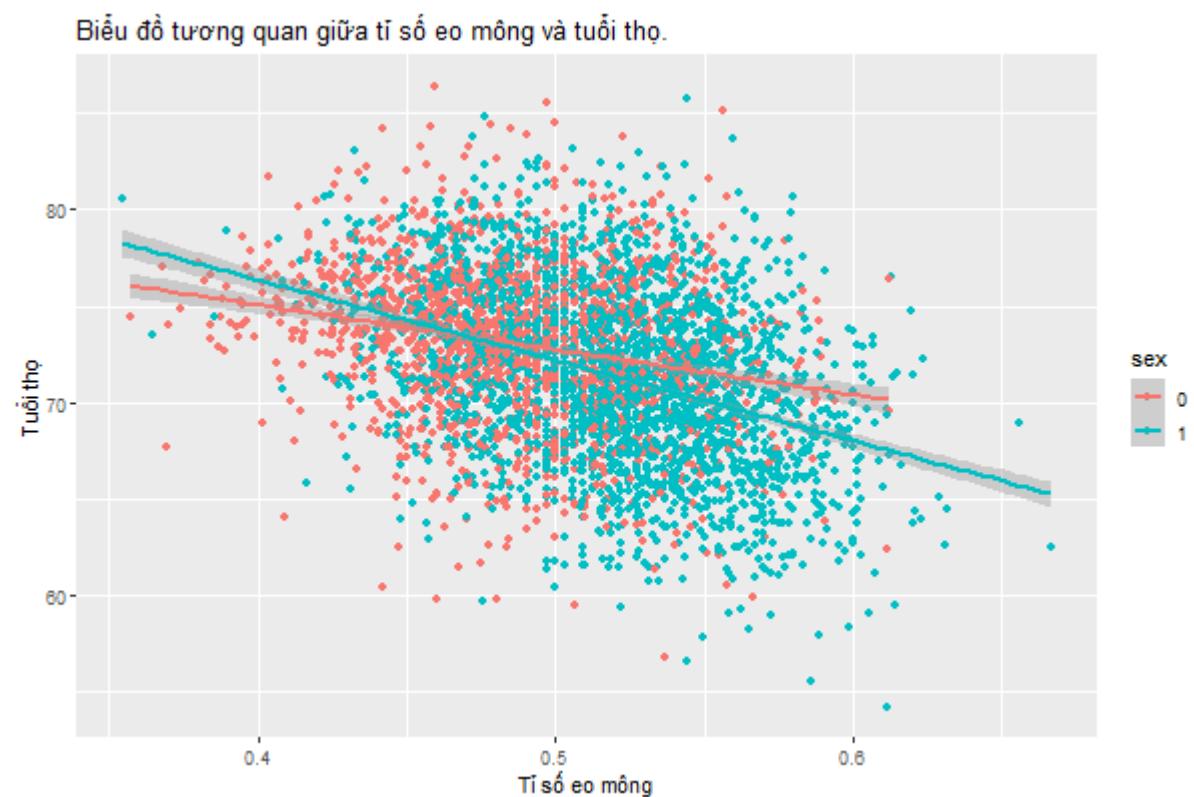
pandas2ri.activate()
r_dataframe = pandas2ri.py2ri(df)

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()
```

Chạm tới AI trong 10 ngày

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



Bài 11: 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

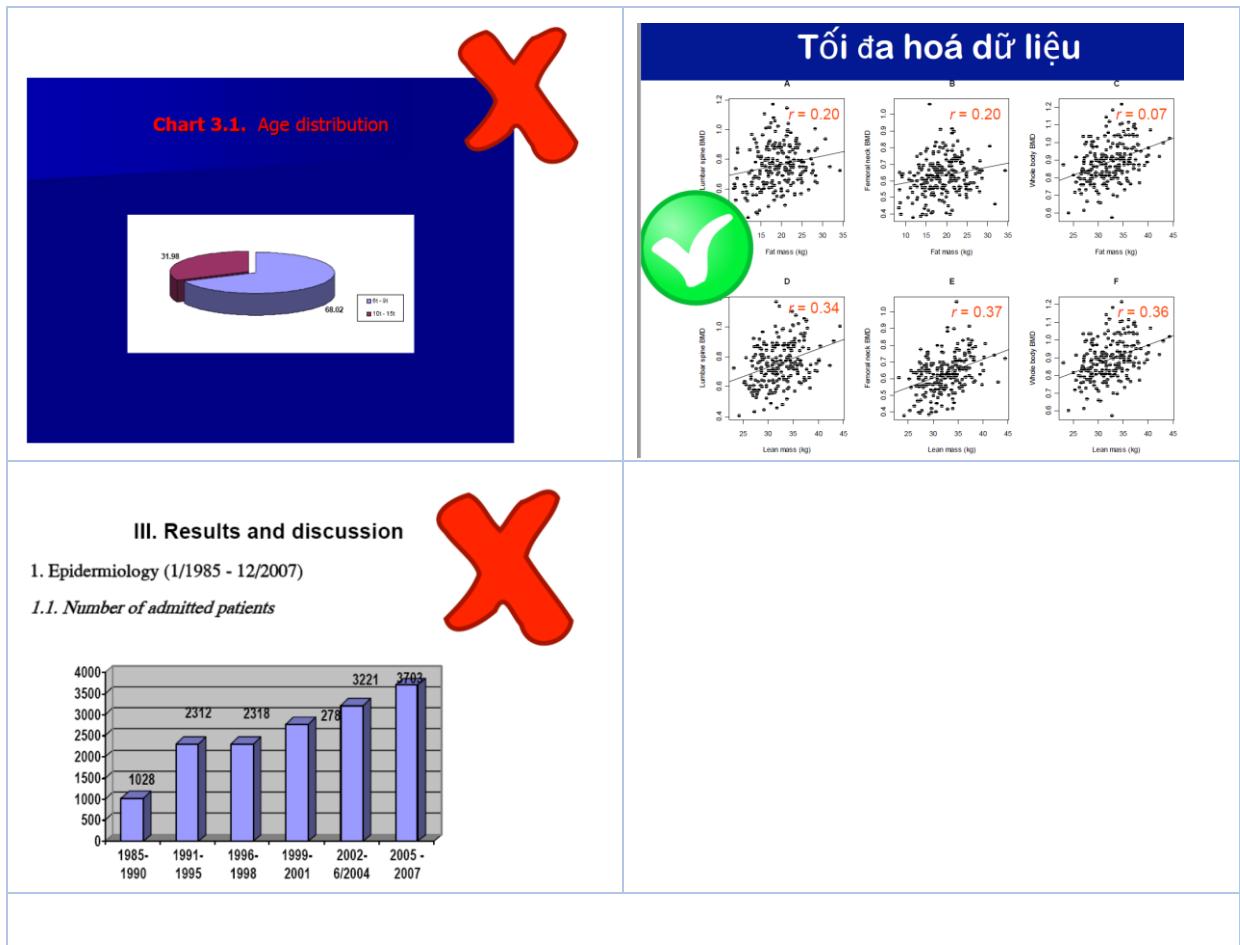
Có thể đây đơn thuần là bài toán kinh tế. Tức là trên một trang giấy nếu bạn dùng ít mực in mà cung cấp được nhiều thông tin nhất thì sẽ có lợi nhất. Vì vậy chúng ta cần chọn biểu đồ, màu sắc sao cho cung cấp cho người đọc nhiều thông tin nhất có thể. Nếu in ra giấy thì tiết kiệm mực nhất có thể.

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



Bài 12: Giới thiệu Matplotlib

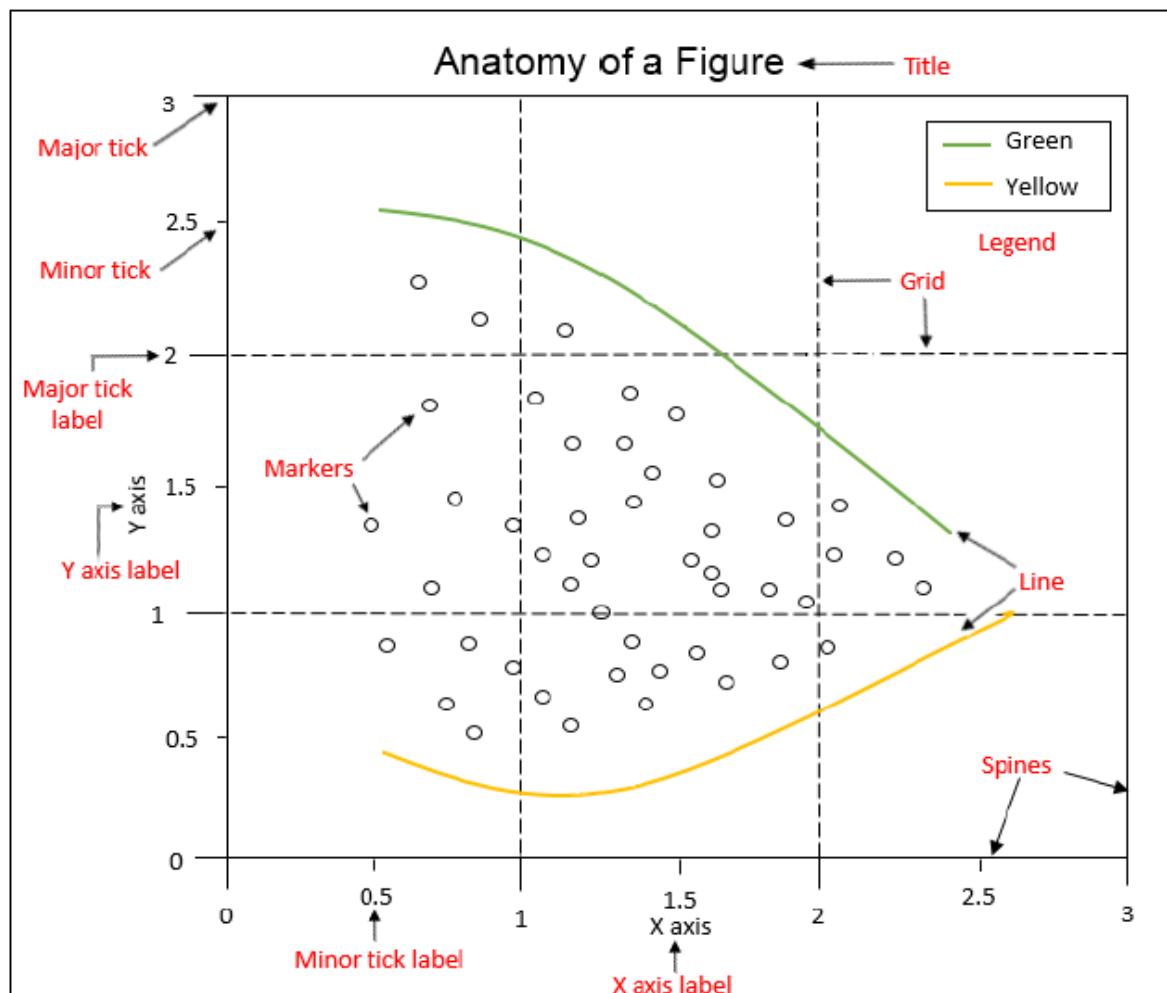
Trong ngày đầu tiên bạn đã làm quen với các loại biểu đồ và đã biết thư viện matplotlib. Tuy nhiên bài này sẽ dành riêng để tìm hiểu về thư viện vẽ biểu đồ phổ biến trong Python này. Matplotlib rất phổ biến trong giới data science (khoa học dữ liệu) và machine learning (máy học). Matplotlib được John 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 trực x, trực y, các giá trị cần thể hiện trong không gian x,y (Markers, Lines, Grid); các thành phần khác để trang trí như: tên trực x (x axis label), tên trực y (y axis label), tiêu đề bức tranh (title), ghi chú (Legend).

Trên hai trực 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 thì giờ cho việc trang trí (sử dụng Figure và Axes).

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

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

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

Tạo figure ≈ tạo khung tranh

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

```
plt.figure()
```

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

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

```
# Thiết lập bề rộng và chiều cao  
plt.figure(figsize=(10, 5))  
  
# Thiết lập dpi: Số điểm ảnh trên một đơn vị Inch  
plt.figure(dpi=300)  
  
Out: <Figure size 1800x1200 with 0 Axes><Figure size 720x360 with 0 Axes>  
<Figure size 1800x1200 with 0 Axes>
```

Đóng figure ≈ tạo 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))  
  
# Thiết lập dpi: Số điểm ảnh trên một đơn vị Inch  
plt.figure(dpi=300)  
  
plt.close()
```

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

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

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

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

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

Chạm tới AI trong 10 ngày

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

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'

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/3.1.1/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" | Ү | tri_down |
| "2" | ѧ | tri_up |
| "3" | Յ | tri_left |
| "4" | Յ | tri_right |
| "8" | ● | octagon |

| marker | symbol | description |
|--------|--------|---------------|
| "s" | ■ | square |
| "p" | ◆ | pentagon |
| "P" | ✚ | plus (filled) |
| "*" | ★ | star |
| "h" | ⬡ | hexagon1 |
| "H" | ⬢ | hexagon2 |
| "+" | + | plus |
| "x" | ✗ | x |
| "X" | ✖ | x (filled) |
| "D" | ◆ | diamond |
| "d" | ◆ | thin_diamond |
| " " | | vline |

Chạm tới AI trong 10 ngày

| marker | symbol | description |
|--------|--------|-------------|
| "_" | — | hline |

Định dạng màu sắc

| Kí hiệu | Màu |
|---------|---------|
| b | blue |
| r | red |
| g | green |
| m | magenta |

| Kí hiệu | Màu |
|---------|--------|
| c | cyan |
| b | black |
| w | white |
| 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 | · · · · · · · · |

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')
```

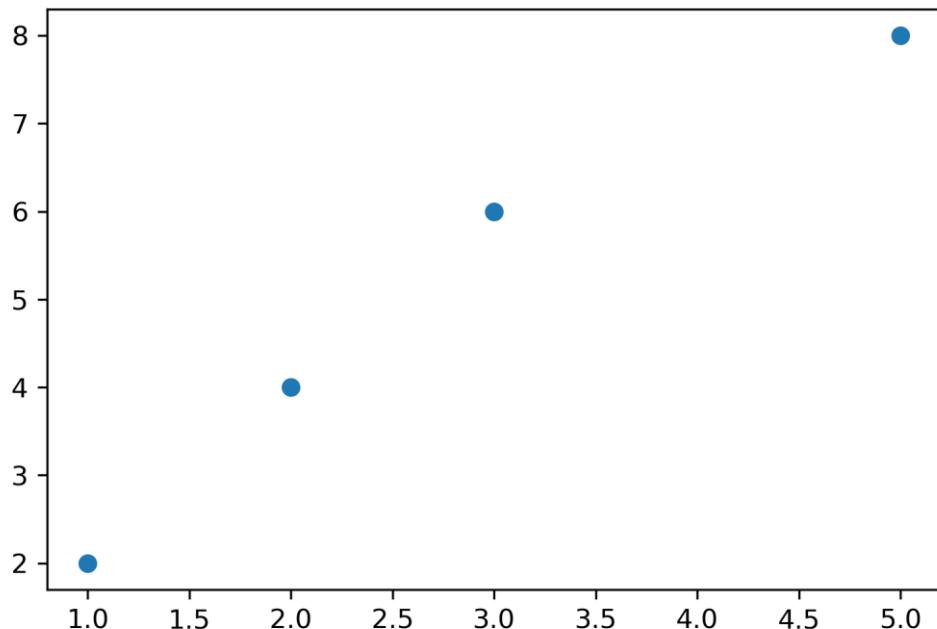
Nhắc lại: kí hiệu [] bao đóng tham số cho biết là không bắt buộc được chỉ định (sẽ sử dụng giá trị mặc định).

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

```
# Tạo Figure với số 1. Thiết lập bề rộng và chiều cao
```

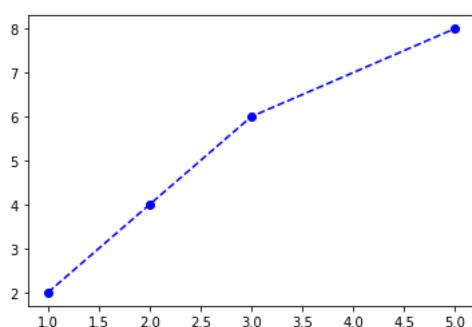
Chạm tới AI trong 10 ngày

```
plt.figure(num=1, figsize=(10, 5))  
# Thiếp 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')
```



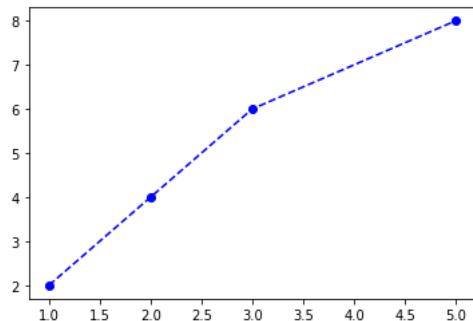
Thử plot với format khác nhau để tự khám phá ý nghĩa các kí hiệu:

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

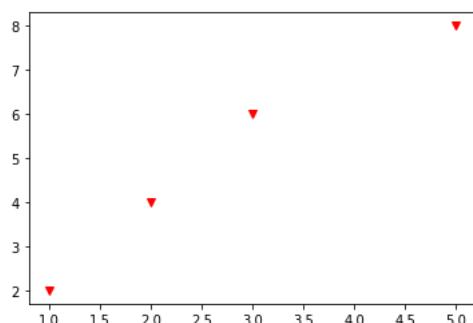


```
plt.plot(x, y, color='blue', marker='o', linestyle='dashed')
```

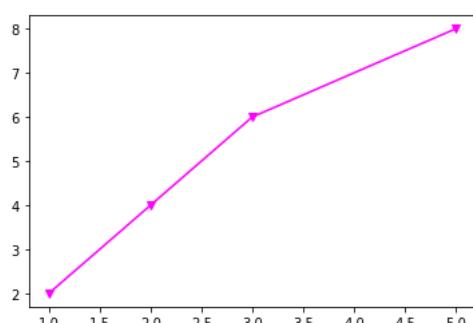
Chạm tới AI trong 10 ngày



```
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

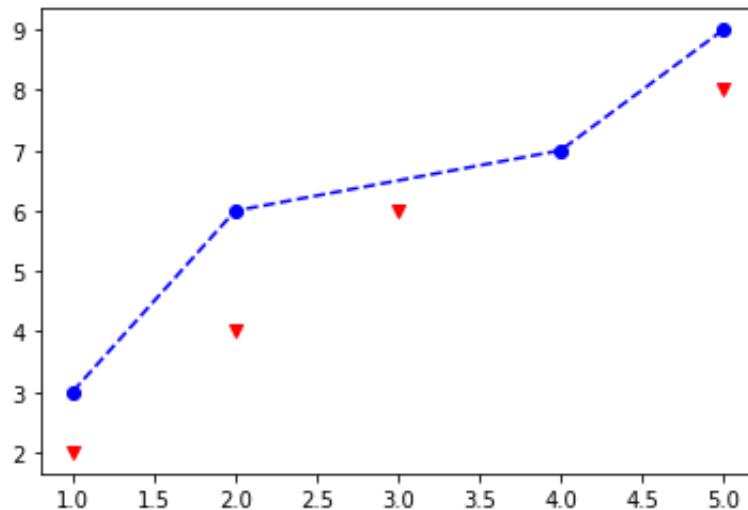
Có thể mở rộng gồm 2 bộ tham số:

```
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--')
```

Chạm tới AI trong 10 ngày



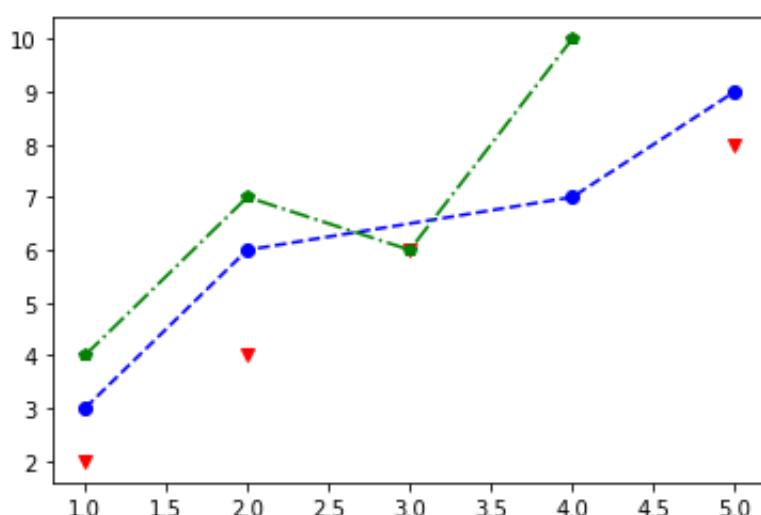
Với 3 bộ tham số:

```
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 với data frame

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

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

Quay lại ví dụ trong Bài 10, vẽ biểu đồ tăng trưởng GDP và CPI

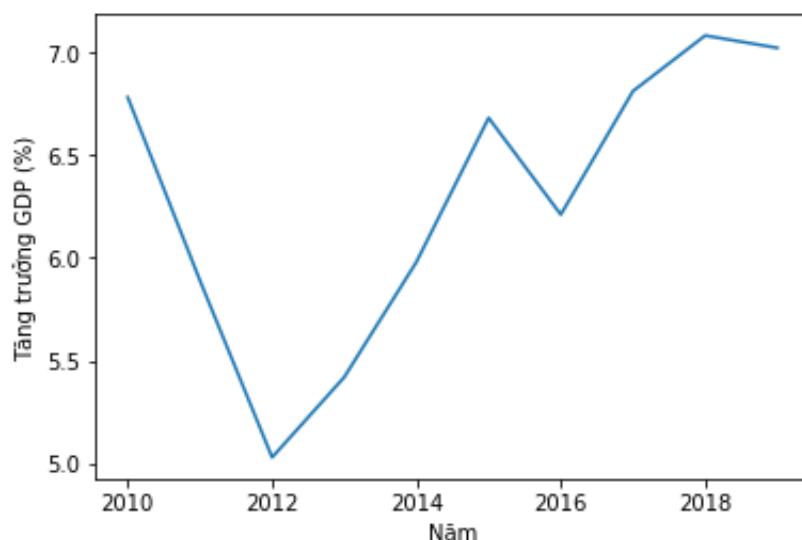
```
import pandas as pd
import matplotlib.pyplot as plt

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 trục x là giá trị cột tên '**year**', trục y là giá trị cột tên '**gdp**', với data frame là **df**.

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
```

Chạm tới AI trong 10 ngày

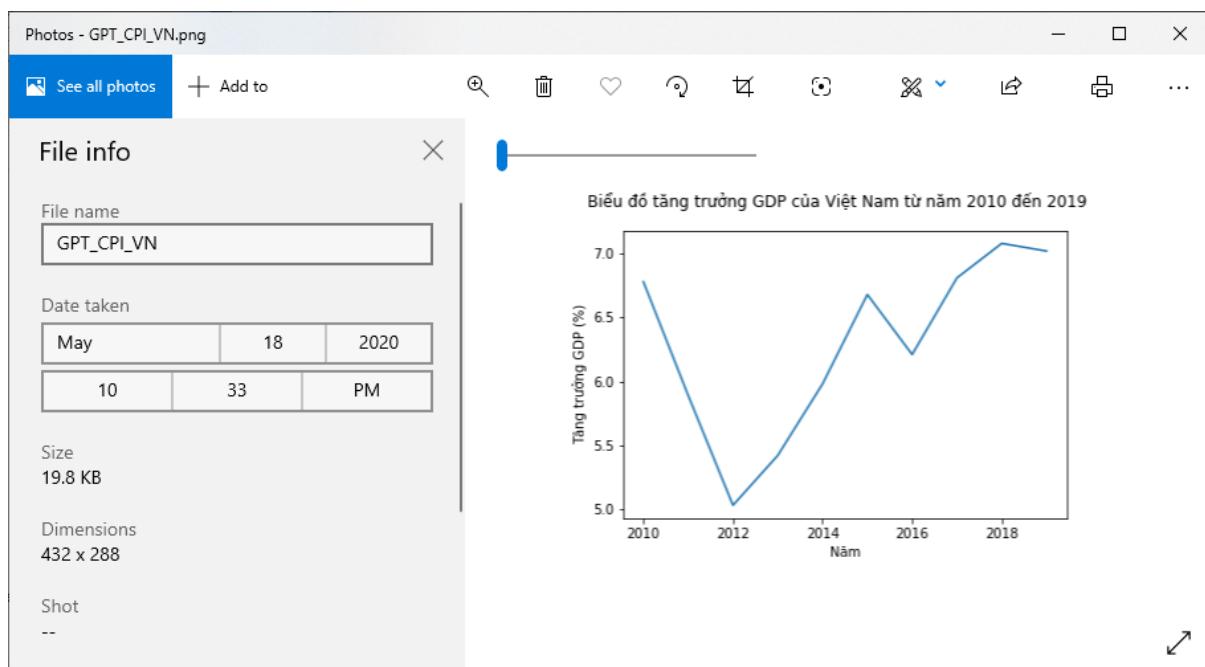
```
import matplotlib.pyplot as plt

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")
```

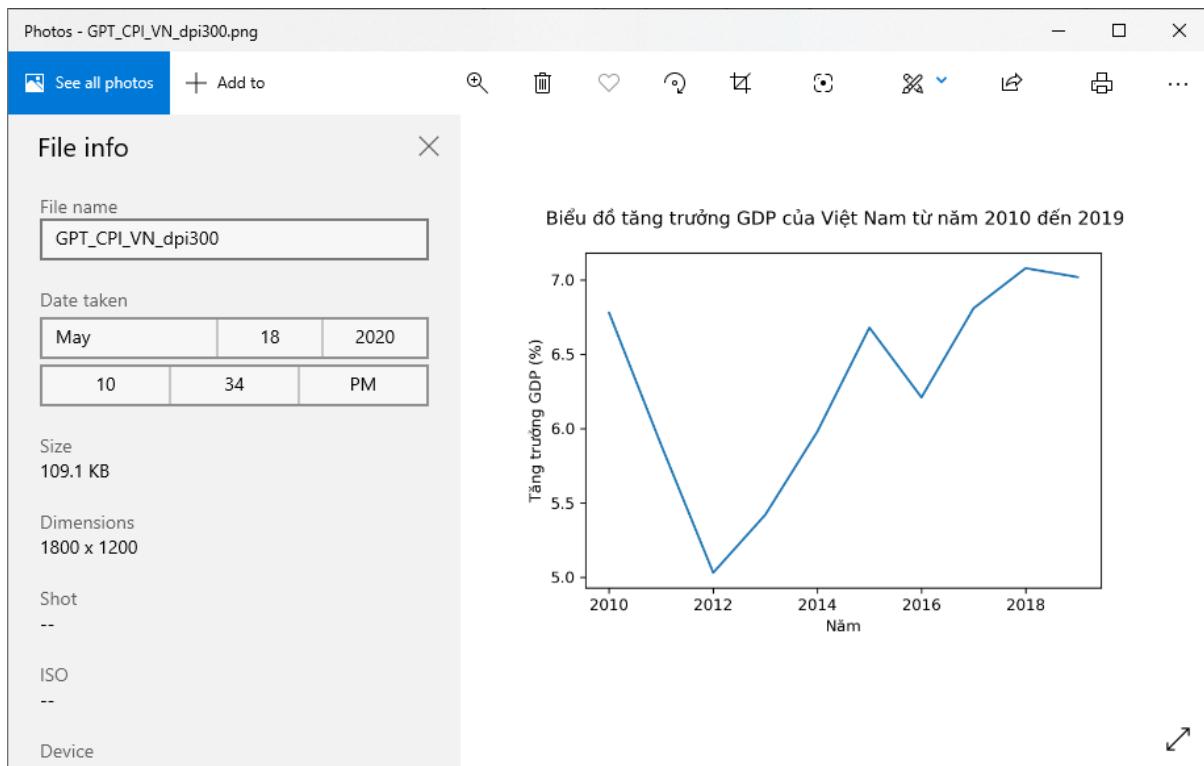
Kết quả:



Thử sửa lại dòng cuối cùng

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

Chạm tới AI trong 10 ngày



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)
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)
```

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 13: 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 14: Giúp bạn nắm được cách so sánh hai tỉ lệ.

Bài 15: Tóm tắt lại lý thuyết về Mô hình kiểm định giả thuyết.

Bài 16: 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 14, 15, 16 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 13: 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 UCI⁶. 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 antechinus (anteater) 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", "Contact", "Search", "Repository", "Web", and "View ALL Data Sets". A Google search bar is also present. The main content area is titled "Bank Marketing Data Set" with a sub-section "Download: Data Folder, Data Set Description". It includes an abstract: "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) |
|----|-------------|---|
| 1 | age | tuổi (numeric) |
| 2 | job | nghề nghiệp |
| 3 | marital | |
| 4 | education | |
| 5 | default | |
| 6 | housing | |
| 7 | loan | |
| 8 | contact | |
| 9 | month | |
| 10 | day_of_week | |
| 11 | duration | |
| 12 | campaign | Số lần liên lạc với khách hàng này. Giá trị số có nghĩa là lần liên lạc gần nhất) |
| 13 | pdays | |

⁶ Link: <https://archive.ics.uci.edu/ml/datasets/bank+marketing>

| | | |
|----|----------------|--|
| 14 | previous | |
| 15 | poutcome | |
| 16 | emp.var.rate | |
| 17 | cons.prive.idx | |
| 18 | cons.conf.idx | |
| 19 | euribor3m | |
| 20 | nr.employed | |

Biến kết quả

| | | |
|----|---|--|
| 22 | 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 mô tả Bank Marketing với R

Thông qua phân tích dữ liệu Bank Marketing này tôi mong đợi là bạn sẽ làm quen tiếp các lệnh R. Cụ thể là đọc số liệu từ file csv, thực hiện mã hóa (thêm cột với kiểu dữ liệu mới) dữ liệu đơn giản. Trong phần phân tích này bạn sẽ được luyện tập thêm kỹ thuật pipeline của thư viện **dplyr**.

Xem cột dữ liệu và làm quen với vài dữ liệu

```
df = read.csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', sep=';')

names(df)
head(df)

> names(df)
[1] "age"      "job"       "marital"    "education"   "defau
1t"        "housing"   "loan"       "contact"     "education"   "defau
[9] "month"    "day_of_week" "duration"   "campaign"   "pdays"
"           "previous"  "poutcome"   "emp.var.rate" "pdays"
[17] "cons.price.idx" "cons.conf.idx" "euribor3m"  "nr.employed" "y"
> head(df)
  age      job marital education default housing loan contact month day_of
 _week duration campaign pdays previous poutcome
1 56 housemaid married basic.4y no      no no telephone may
mon 261      1 999      0 nonexistent
2 57 services married high.school unknown no      no no telephone may
mon 149      1 999      0 nonexistent
3 37 services married high.school no      yes no telephone may
mon 226      1 999      0 nonexistent
```

Chạm tới AI trong 10 ngày

| | | | | | | | | | |
|--------------|----------------|---------------|-----------|-------------|---------|-------------|-----|-----------|-----|
| 4 | 40 | admin. | married | basic.6y | no | no | no | telephone | may |
| mon | 151 | | 1 | 999 | 0 | nonexistent | | | |
| 5 | 56 | services | married | high.school | no | no | yes | telephone | may |
| mon | 307 | | 1 | 999 | 0 | nonexistent | | | |
| 6 | 45 | services | married | basic.9y | unknown | no | no | telephone | may |
| mon | 198 | | 1 | 999 | 0 | nonexistent | | | |
| emp.var.rate | cons.price.idx | cons.conf.idx | euribor3m | nr.employed | y | | | | |
| 1 | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no | | | |
| 2 | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no | | | |
| 3 | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no | | | |
| 4 | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no | | | |
| 5 | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no | | | |
| 6 | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no | | | |

Bằng hai lệnh **names(df)** và **head(df)** bạn sẽ phần nào biết được chút ít về bộ dữ liệu này.

Một lệnh khác khá hữu ích trong thư viện dplyr là `glimpse`.

```
glimpse(df)
```

Chạm tới AI trong 10 ngày

Lệnh `glimpse` sẽ giúp bạn quan sát được tổng thể số dòng, số cột, tên các cột, kiểu dữ liệu tương ứng và vài dữ liệu được trình bày theo hàng ngang.

Quan sát một số chỉ số cơ bản

| Lệnh | Kết quả |
|--|---------------------------------------|
| <code>mean(df\$age)</code> | 40.02406 |
| <code>median((df\$age))</code> | 38 |
| <code>findmode <- function(x) { uniqx = unique(x) uniqx[which.max(tabulate(match(x, uniqx)))] }</code> | 31 |
| <code>findmode(df\$age)</code> | |
| <code>quantile(df\$age)</code> | 0% 25% 50% 75% 100% 17 32 38 47 98 |
| <code>range(df\$age)</code> | 17 98 |
| <code>var(df\$age)</code> | 108.6025 |
| <code>sd(df\$age)</code> | 10.42125 |

Trong R theo tôi biết đến thời điểm hiện tại thì chưa có sẵn hàm `mode` để tìm giá trị lặp lại nhiều nhất nên chúng ta phải viết code phức tạp một chút:

```
findmode = function(x) {  
  uniqx = unique(x)  
  uniqx[which.max(tabulate(match(x, uniqx)))]  
}
```

Hàm `findmode` được viết theo cấu trúc như bên dưới để định nghĩa hàm tên là `findmode` có tham số là x:

```
findmode = function(x) {  
  # Các lệnh bên trong hàm findmode  
}
```

Như tôi đã nhắc ở các bài trước thì phép gán có thể dùng dấu mũi tên trái `<-`. Hầu hết các tài liệu do nhà lập trình viết thì dùng dấu mũi tên này `<-` nên nếu bạn đọc sách thì sẽ đừng thấy lạ. Với quan điểm làm sao cho đơn giản và gọn, hiệu quả nhất có thể thì tôi thích dấu `=` hơn.

Chạm tới AI trong 10 ngày

Bàn về cách viết hàm (function) hoặc các thuật ngữ khác trong giới lập trình thường dùng là method, procedure. Các khái niệm function, procedure method thì các bạn học chuyên ngành về lập trình thì chắc phân biệt được. Ở đây với góc độ là người đi tìm ý nghĩa của dữ liệu thì bạn chỉ cần nắm được ý tưởng cốt lõi của nó là: vì tôi muốn đi tìm một kết quả nào đó mà trong R không có sẵn cái hàm để tôi gọi, hoặc chưa có ai viết sẵn cái hàm làm đúng mục đích tôi cần nên tôi phải tự viết láy. Trong trường hợp là chúng ta tự viết hàm **findmode** để đi tìm số được lặp lại nhiều nhất trong cột dữ liệu. Cấu trúc hàm như sau:

```
<tên hàm> = function(cá tham số cách nhau bởi dấu phẩy) {  
}
```

Cách định nghĩa hàm trong R cũng hơi lạ so với các ngôn ngữ lập trình khác. Chủ đề này không thuộc phạm vi tài liệu này nên tôi nghĩ là một bài tập thú vị cho các bạn muốn đi sâu về kỹ thuật lập trình.

Bàn về các lệnh bên trong hàm **findmode** để hy vọng bạn thấy thú vị một chút.

Đầu tiên là bạn thấy lệnh **unique(x)**. Nếu bạn chịu khó đọc hướng dẫn bằng cách gõ lệnh:

```
?unique
```

thì sẽ được tài liệu hướng dẫn như sau:

The screenshot shows the R Documentation page for the `unique` function. At the top, there is a search bar with the text `R: Extract Unique Elements` and a button labeled `Find in Topic`. Below the search bar, the function name `unique {base}` is displayed. To the right, it says `R Documentation`. The main title is **Extract Unique Elements**. Under the title, there is a **Description** section which states: `unique` returns a vector, data frame or array like `x` but with duplicate elements/rows removed. Below the description is a **Usage** section containing the following R code:

```
unique(x, incomparables = FALSE, ...)  
  
## Default S3 method:  
unique(x, incomparables = FALSE, fromLast = FALSE,  
      nmax = NA, ...)  
  
## S3 method for class 'matrix'  
unique(x, incomparables = FALSE, MARGIN = 1,  
      fromLast = FALSE, ...)  
  
## S3 method for class 'array'  
unique(x, incomparables = FALSE, MARGIN = 1,  
      fromLast = FALSE, ...)
```

Hàm `unique(df$age)` sẽ cung cấp cho chúng ta các tuổi không trùng nhau. Tức là mỗi giá tuổi trong cột `age` sẽ xuất hiện duy nhất một lần trong kết quả trả về của hàm. Cụ thể:

Chạm tới AI trong 10 ngày

```
unique(df$age)
```

```
[1] 56 57 37 40 45 59 41 24 25 29 35 54 46 50 39 30 55 49 34 52 58 32 38 44 42  
[6] 60 53 47 51 48 33 31 43 36 28 27 26 22 23 20 21 61 19 18 70  
[46] 66 76 67 73 88 95 77 68 75 63 80 62 65 72 82 64 71 69 78 85 79 83 81 74 17  
[87] 87 91 86 98 94 84 92 89
```

Chú ý trong kết quả trả về, Rstudio có hiển thị [1], rồi [46]. Cái này đánh dấu số thứ tự của số được hiển thị. Tức là trong dãy tuổi trả về này thì tuổi 66 ở vị trí 46.

Kết quả được lưu vào biến uniqx thông qua phép gán.

```
uniqx = unique(df$age)
```

Lệnh tiếp theo:

```
uniqx[which.max(tabulate(match(x, uniqx)))]
```

Lệnh này phức tạp nên chúng ta sẽ phân tích từng phần.

Một lệnh nhỏ match (x, uniqx) sẽ đi tìm giá trị x có trùng với phần tử nào trong dãy số uniqx hay không? Hãy thử lệnh sau:

```
match(df$age, uniqx)
```

Tương tự bạn có thể tách nhỏ từng lệnh ra và gán vào một cái biến để xem giá trị của lệnh, kèm theo đọc tài liệu thì sẽ hiểu được. Các lệnh tôi tách ra như sau:

```
uniqx = unique(df$age)  
matchx = match(df$age, uniqx)  
tabulatex = tabulate(matchx)  
posx = which.max(tabulatex)  
uniqx[posx]
```

Kết quả là:

31

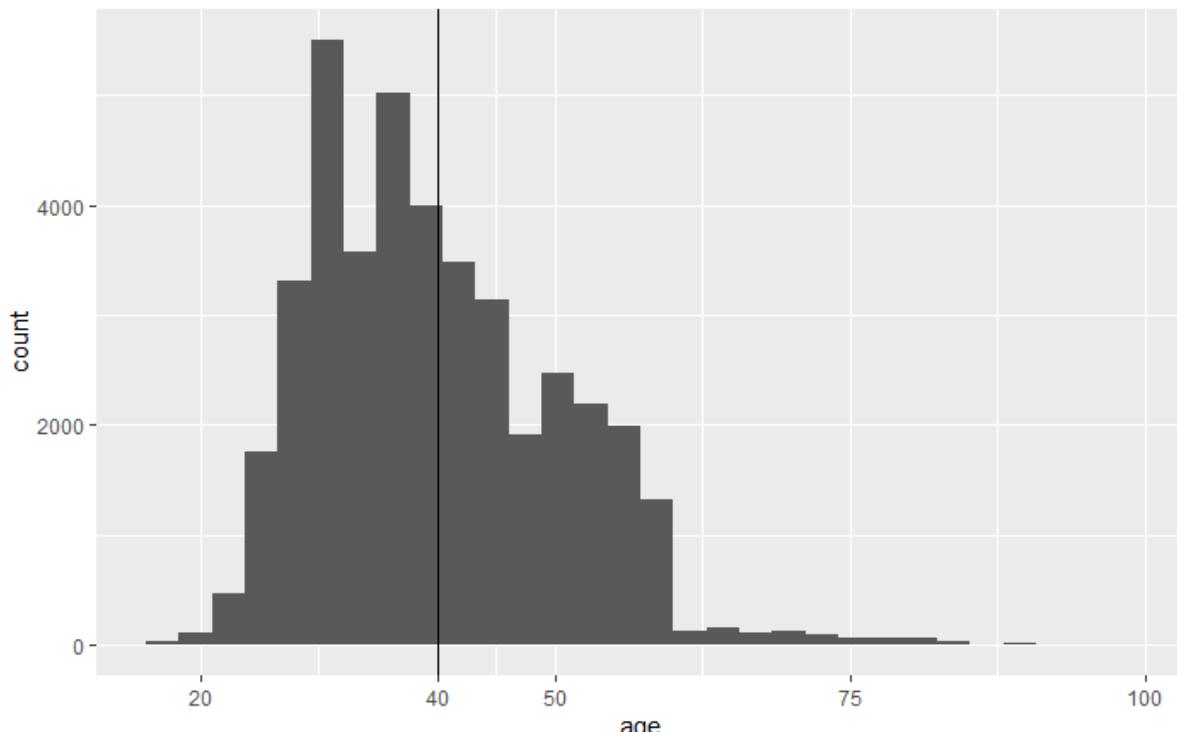
Có hai chỉ số mà chúng ta nên bàn sâu một chút là **độ lệch chuẩn và phương sai**. Cụ thể độ lệch chuẩn và phương sai của tuổi trong bộ dữ liệu Bank Marketing này như sau:

| Lệnh | Kết quả |
|---------------|----------|
| mean(df\$age) | 40.02406 |
| sd(df\$age) | 10.42125 |
| var(df\$age) | 108.6025 |

Để hình dung phương sai và độ lệch chuẩn, chúng ta vẽ phân bố tuổi và đường trung bình xem sao:

Chạm tới AI trong 10 ngày

```
library(ggplot2)
p = ggplot(data = df, aes(x = age))
p = p + geom_histogram()
p = p + geom_vline(xintercept = mean(df$age))
p = p + scale_x_continuous(name = 'age', breaks = c(20,
as.integer(mean(df$age)), 50, 75, 100))
p
```



Biểu đồ trên cho thấy rất nhiều khách hàng tiềm năng được gọi điện trong chiến dịch marketing này lệch so với tuổi trung bình (40).

Nếu tính khoảng cách tuổi của **từng khách hàng** với đường trung bình rồi cộng lại. Sau đó chia đều (chia cho số lượng khách hàng) thì chúng ta sẽ hình dung được **độ chênh lệch của tuổi** trong tập dữ liệu. Chênh lệch ở đây là so với tuổi trung bình.

Phương sai (Variable)

Gọi hàm **var (df\$age)** sẽ cho ra phương sai của tuổi là 108.6025. Bạn hình dung cách mô tả độ chênh lệch của tuổi ở trên. Khi tính hiệu của tuổi so với tuổi trung bình thì sẽ có số âm. Vì thế người ta bình thường nó lên. Như vậy để hình dung độ lệch tuổi thì người ta bình phương giá trị hiệu của tuổi từng người và tuổi trung bình. Sau đó tính tổng hết lại. Cuối cùng chia đều để lấy số trung bình. Số này gọi là **phương sai**. (Xem lại ví dụ và công thức trong bài 1 nếu bạn vẫn chưa hình dung được).

Vì là **tổng trung bình bình phương** của chênh lệch tuổi từng người so với tuổi trung bình (nhắc lại là **tổng trung bình bình phương**) cho nên bạn thấy số 108.6025 rất lớn trong sự tưởng tượng về độ lệch?

Độ lệch chuẩn (SD – Standard deviation)

Một cách ngắn gọn, SD dùng để mô tả sự khác biệt giữa các cá nhân (của một biến liên tục) trong một mẫu. Cụ thể SD của age là 10.4 trong ví dụ đang bàn nói lên điều gì?

Quay lại khái niệm phuong sai ở trên. Có câu hỏi là con số 108.6025 rất lớn trong sự tưởng tượng về độ lệch? Vì nó là **là tổng trung bình bình phuong** nên người ta phải căn bậc hai của nó mới đúng nghĩa là độ lệch.

Vì vậy $SD = \text{căn bậc hai (phuong sai)}$.

Bạn thử gõ 2 lệnh sau sẽ cho kết quả như nhau: **10.42125**

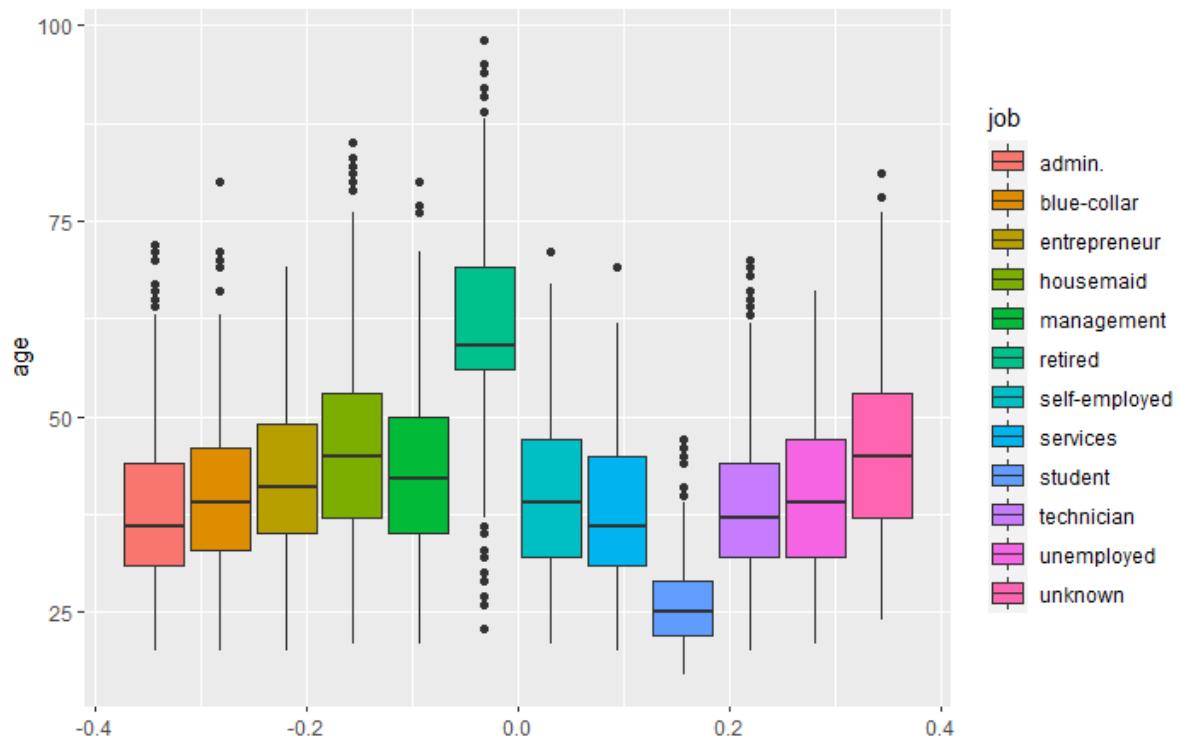
```
sqrt(108.6025)  
sd(df$age)
```

Các khách hàng tiềm năng trong chiến dịch Marketing này có sự khác biệt nhau tầm 10 tuổi xung quanh tuổi trung bình.

Xem phân bố tuổi theo công việc

```
df = read.csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', sep=';')  
  
library(ggplot2)  
p = ggplot(data = df, aes(y=age, fill = job))  
p = p + geom_boxplot()  
p
```

Chạm tới AI trong 10 ngày



Legend bên phải có thứ tự tương ứng với các box từ trái sang phải.

Biểu đồ cho thấy người về hưu (retired) có tuổi rất khác biệt với nhóm còn lại. Điều này là đương nhiên rồi. Tiếp theo là nhóm nội trợ (housemaid) cũng là nhóm có tuổi cao tiếp theo (gần gần với nhóm “unknown”). Thấp nhất là nhóm sinh viên (student). Điều này cho thấy là bình thường và hợp lý.

Tính tỉ lệ chuyển đổi – conversion rate

Khái niệm: Trong kinh tế khi một công ty thực hiện một chương trình để thu hút được người chưa mua hàng hoặc chưa sử dụng dịch vụ của công ty (tức là người chưa phải là khách hàng) trở thành người mua hàng hoặc dùng dịch vụ của mình (tức là khách hàng) thì gọi là chuyển đổi (conversion). Tỉ liệu chuyển đổi dùng để đo mức độ mua hàng hoặc sử dụng dịch vụ lần đầu

Đoạn code R sau sẽ đọc dữ liệu từ website và thực hiện tính tỉ lệch chuyển đổi.

```
df = read.csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', sep=';')

names(df)
head(df)

df$conversion[df$y == 'yes'] = 1
df$conversion[df$y == 'no'] = 0

# total number of conversions
sum(df$conversion)
```

Chạm tới AI trong 10 ngày

```
nrow(df)

sprintf('Tỉ số chuyển đổi: %0.2f%%', sum(df$conversion) / nrow(df) * 100)
```

Conversion rate theo độ tuổi

```
library(dplyr)
conversionsByAge = df %>% group_by(Age=age) %>%
  summarise(TotalCount=n(), NumConversions=sum(conversion)) %>%
  mutate(ConversionRate=NumConversions/TotalCount*100.0)

head(conversionsByAge, 10)

# A tibble: 10 x 4
  Age TotalCount NumConversions ConversionRate
  <dbl>     <dbl>        <dbl>      <dbl>
1 17         5            2        40
2 18        28           12       42.9
3 19        42           20       47.6
4 20        65           23       35.4
5 21       102           29       28.4
6 22       137           36       26.3
7 23       226           48       21.2
8 24       463           86       18.6
9 25       598           93       15.6
10 26      698          122      17.5
```

Code **tỉ lệ chuyển đổi theo độ tuổi** thì phức tạp hơn một chút.

Cú pháp `%>%` trong thư viện `dplyr` giúp ta xử lý dữ liệu liên tục rất hiệu quả theo trình tự như sau:

Bước 1: Lệnh bên dưới sẽ đầy từng dòng dữ liệu của `df` đi qua một đường ống (bạn hình dung cú pháp `%>%` giống như một đường ống, thuật ngữ gọi là **pipeline**). Trong đường ống này sẽ được hàm `group_by(Age = age)` gom nhóm dữ liệu theo độ tuổi. Tức là các dòng dữ liệu có `age` bằng nhau sẽ được gom lại thành một dòng thôi. Và cột `tuổi` được gom này được đặt tên là `Age` (chữ A hoa để phân biệt với cột `age` ban đầu)

```
df %>% group_by(Age=age)
```

Bước 2: Dữ liệu được gom nhóm ở bước một sẽ được tiếp tục xử lý bằng cách cho chạy qua một **pipeline** nữa (phần in đậm trong lệnh bên dưới).

Pipeline trong bước 2 này sẽ làm nhiệm vụ **tổng hợp** (`summarise`) dữ liệu bằng cách tạo ra một biến mới là `NumConversions` có giá trị là **tổng** của biến `conversion`.

Chạm tới AI trong 10 ngày

```
df %>% group_by(Age=age) %>%
summarise(NumConversions=sum(conversion))
```

| # A tibble: 78 x 2 | Age | NumConversions |
|-------------------------|-------|----------------|
| | <int> | <dbl> |
| 1 | 17 | 2 |
| 2 | 18 | 12 |
| 3 | 19 | 20 |
| 4 | 20 | 23 |
| 5 | 21 | 29 |
| 6 | 22 | 36 |
| 7 | 23 | 48 |
| 8 | 24 | 86 |
| 9 | 25 | 93 |
| 10 | 26 | 122 |
| # ... with 68 more rows | | |

Hãy nhớ là conversion là cột có giá trị 1 hoặc 0 (1 tức là người được gọi điện đã trở thành khách hàng; 0 là chưa thành khách hàng). Như vậy sum (conversion) sẽ cho ra tổng số người được gọi điện đã trở thành khách hàng.

Quan sát kết quả bạn sẽ thấy dữ liệu của lệnh trên có 2 cột. Cột thứ nhất Age gồm các tuổi không lặp lại. Cột thứ hai NumConversions là tổng số **chuyển đổi** như đã giải thích ở trên. Đến đây chắc là bạn hình dung được lệnh %>% lợi hại phần nào rồi chứ? Nhưng chưa hết. Để tôi giải thích tiếp phần còn lại của bước 2. Trong bước 2 ở trên tôi chưa đề cập đến tham số **TotalCount=n()** trong hàm summarise. Hàm summarise cho phép bạn tổng hợp dữ liệu và tạo ra biến mới để chứa dữ liệu tổng hợp đó. Đặc biệt là cho phép tổng hợp nhiều dữ liệu được viết theo cú pháp sau:

```
summarise(biến mới = hàm_tổng_hợp(), biến mới = hàm_tổng_hợp(), ...)
```

Trong lệnh bên dưới tổng hợp hai biến mới là TotalCount và NumConversions.

Hàm n() sẽ tính tổng các dòng tương ứng của cùng độ tuổi. Tức là tính tổng theo biến age đã được gom nhóm bởi pipeline trước đó.

```
df %>% group_by(Age=age) %>%
summarise(TotalCount=n(), NumConversions=sum(conversion))
```

Kết quả lệnh này như sau:

```
# A tibble: 78 x 3
  Age  TotalCount NumConversions
  <int>      <int>        <dbl>
1    17          5            2
2    18         28           12
3    19         42           20
4    20         65           23
5    21        102           29
6    22        137           36
7    23        226           48
8    24        463           86
9    25        598           93
10   26        698          122
# ... with 68 more rows
```

Chạm tới AI trong 10 ngày

Bạn thấy gồm có 3 cột cho biết theo từng độ tuổi, tổng số dòng dữ liệu tương ứng, tổng số dữ liệu “**chuyển đổi**”.

Với bảng kết quả trên bạn hoàn toàn có thể tự tính tỉ lệ chuyển đổi theo độ tuổi. Sức mạnh của cú pháp pipeline %>% sẽ được phát huy tiếp trong bước 3 cho việc này.

Bước 3: Dữ liệu sẽ được tiếp tục được đưa qua pipeline để tạo ra biến mới ConversionsRate như sau (phản in đậm)

```
df %>% group_by(Age=age) %>%
  summarise(TotalCount=n(), NumConversions=sum(conversion)) %>%
  mutate(ConversionRate=NumConversions/TotalCount*100.0)
```

Kết quả của 3 bước xử lý qua kỹ thuật pipeline ở trên sẽ được lưu vào một data frame mới tên là **conversionsByAge** bởi phép gán như bạn đã biết:

```
conversionsByAge = df %>% group_by(Age=age) %>%
  summarise(TotalCount=n(), NumConversions=sum(conversion)) %>%
  mutate(ConversionRate=NumConversions/TotalCount*100.0)
```

Nhắc lại, phép gán ngoài cú pháp dấu bằng thì có thể dùng dấu <- (2 ký tự < và – ghép lại giống như mũi tên) nên bạn có thể viết:

```
conversionsByAge <- df %>% group_by(Age=age) %>%
  summarise(TotalCount=n(), NumConversions=sum(conversion)) %>%
  mutate(ConversionRate=NumConversions/TotalCount*100.0)
```

Đặc biệt, với cú pháp pipeline (tôi tạm gọi là kỹ thuật đường ống cho bạn dễ mường tượng) thì bạn đang hình dung là dữ liệu đang được xử lý và chạy qua các pipeline. Như thế kết quả dữ liệu sẽ được chạy tiếp vào cái biến luôn cho thuận suy nghĩ. Để cho dữ liệu chạy vào biến thì dùng dấu mũi tên thuận sang phải ->.

Lệnh trên có thể viết lại như sau:

```
df %>% group_by(Age=age) %>%
  summarise(TotalCount=n(), NumConversions=sum(conversion)) %>%
  mutate(ConversionRate=NumConversions/TotalCount*100.0) ->
  conversionsByAge
```

Tùy theo thói quen hoặc quy ước trong nhóm, trong tổ chức thì bạn chọn cách viết cho tiện với số đông.

Xem dòng kết quả trong data frame conversionsByAge:

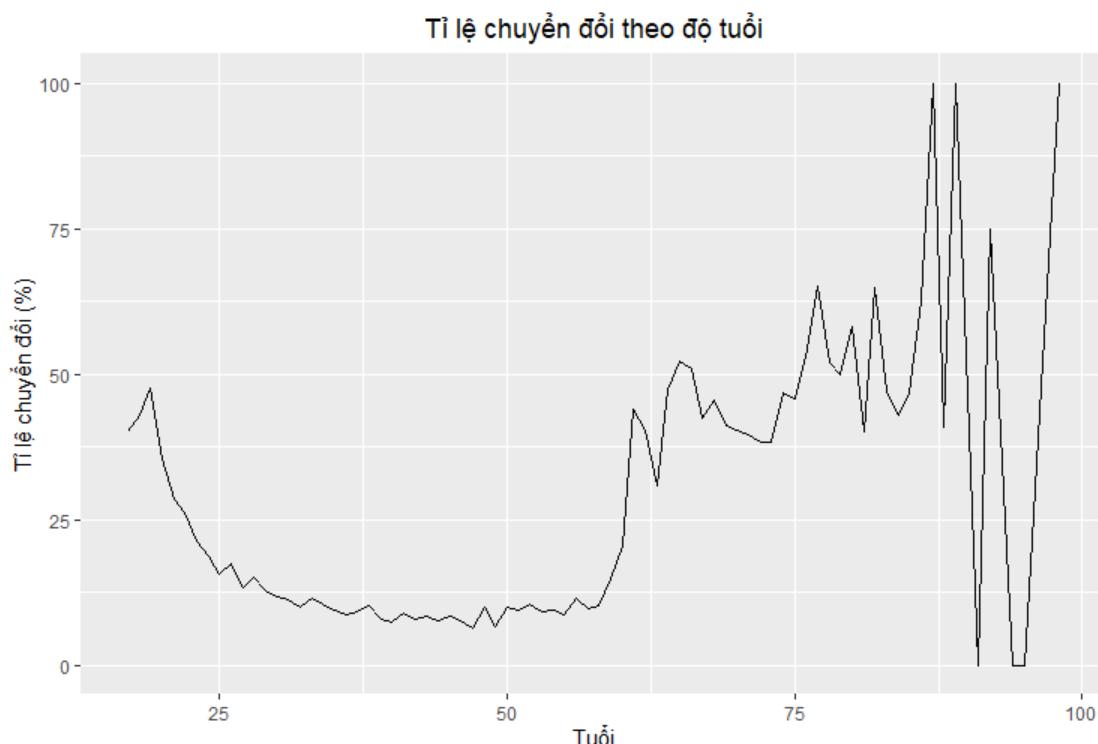
Chạm tới AI trong 10 ngày

```
head(conversionsByAge, 5)
```

| # A tibble: 5 x 4 | Age | TotalCount | NumConversions | ConversionRate |
|-------------------|-------|------------|----------------|----------------|
| <int> | <int> | <dbl> | <dbl> | |
| 1 | 17 | 5 | 2 | 40 |
| 2 | 18 | 28 | 12 | 42.9 |
| 3 | 19 | 42 | 20 | 47.6 |
| 4 | 20 | 65 | 23 | 35.4 |
| 5 | 21 | 102 | 29 | 28.4 |

Xem biểu đồ

```
# line chart
ggplot(data=conversionsByAge, aes(x=Age, y=ConversionRate)) +
  geom_line() +
  ggtitle('Tỉ lệ chuyển đổi theo độ tuổi') +
  xlab("Tuổi") +
  ylab("Tỉ lệ chuyển đổi (%)") +
  theme(plot.title = element_text(hjust = 0.5))
```



Conversion rate theo số lần liên lạc

Đến lúc này bạn có thể tự gõ, giải thích ý nghĩa của lệnh và khám phá kết quả tổng hợp cho câu hỏi sau “Theo từng độ tuổi thì tỉ lệ chuyển đổi bao nhiêu?”:

```
conversionsByNumContact = df %>%
```

Chạm tới AI trong 10 ngày

```
group_by(NumContact=campaign) %>%
summarise(TotalCount=n(), NumConversions=sum(conversion)) %>%
mutate(ConversionRate=NumConversions/TotalCount*100.0)
```

Xem 10 kết quả đầu tiên:

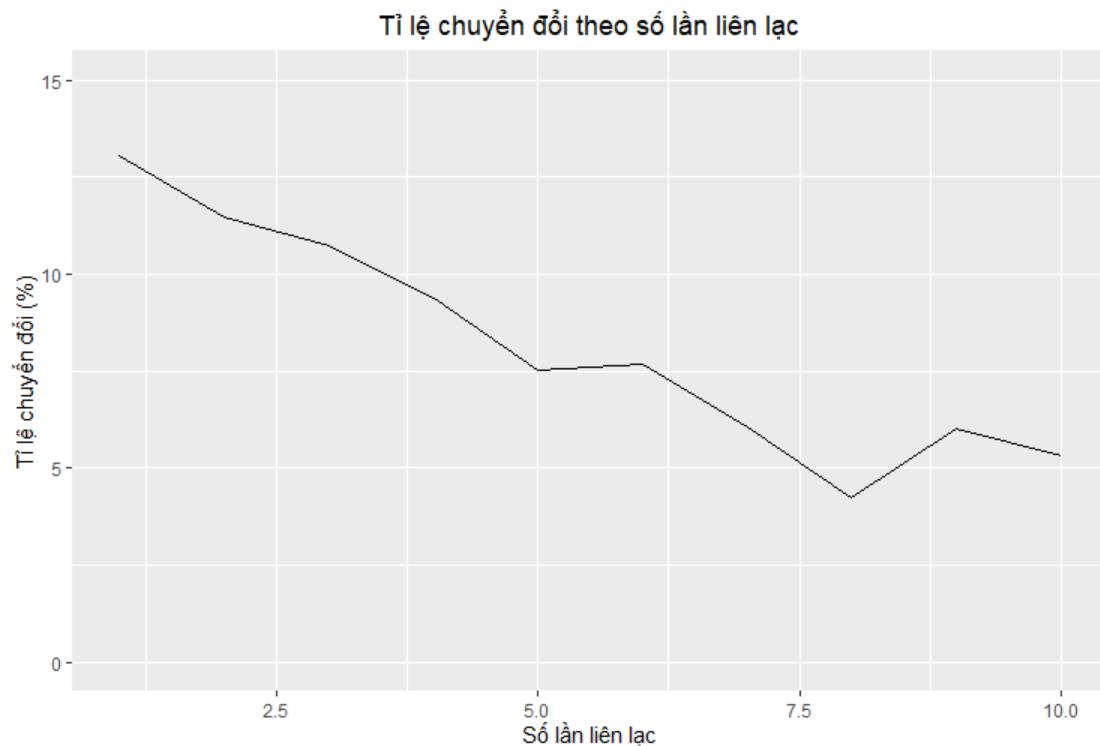
```
head(conversionsByNumContact, 10)
```

| | NumContact | TotalCount | NumConversions | ConversionRate |
|----|------------|------------|----------------|----------------|
| | <int> | <int> | <dbl> | <dbl> |
| 1 | 1 | 17642 | 2300 | 13.0 |
| 2 | 2 | 10570 | 1211 | 11.5 |
| 3 | 3 | 5341 | 574 | 10.7 |
| 4 | 4 | 2651 | 249 | 9.39 |
| 5 | 5 | 1599 | 120 | 7.50 |
| 6 | 6 | 979 | 75 | 7.66 |
| 7 | 7 | 629 | 38 | 6.04 |
| 8 | 8 | 400 | 17 | 4.25 |
| 9 | 9 | 283 | 17 | 6.01 |
| 10 | 10 | 225 | 12 | 5.33 |

Xem biểu đồ

```
ggplot(data=head(conversionsByNumContact, 10), aes(x=NumContact,
y=ConversionRate)) +
geom_line() +
ggtitle('Tỉ lệ chuyển đổi theo số lần liên lạc') +
xlab("Số lần liên lạc") +
ylab("Tỉ lệ chuyển đổi (%)") +
ylim(c(0, 15)) +
theme(plot.title = element_text(hjust = 0.5))
```

Chạm tới AI trong 10 ngày



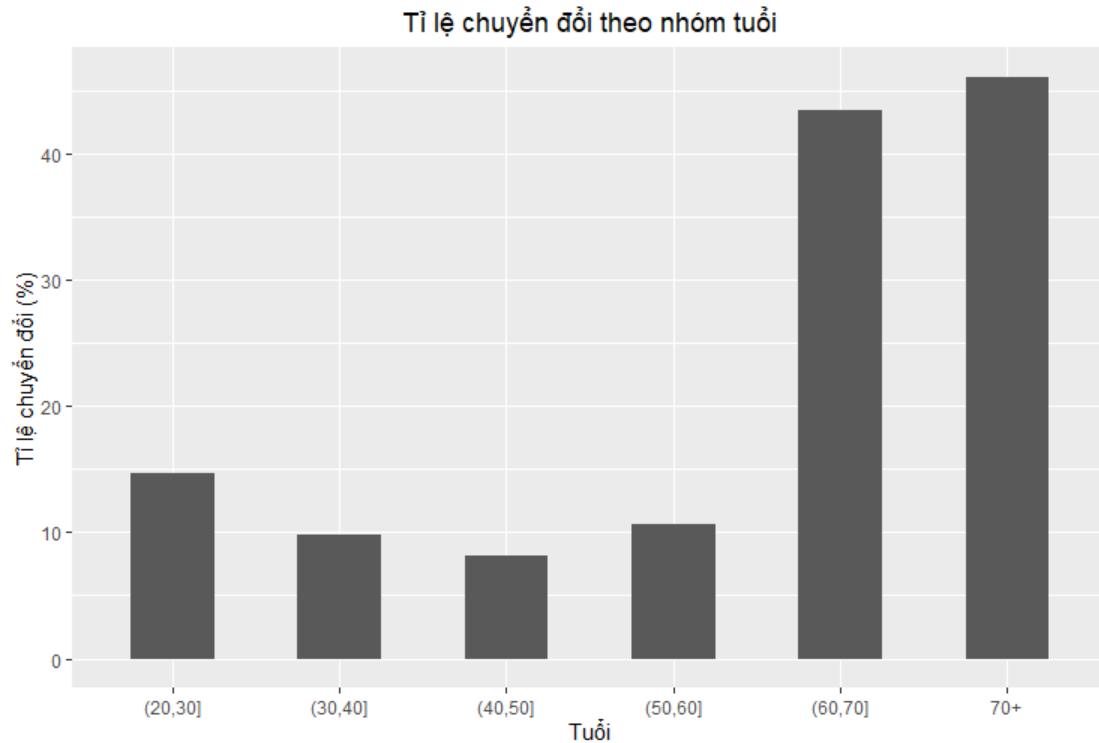
Conversion rate theo nhóm tuổi

```
conversionsByAgeGroup = df %>%
  group_by(AgeGroup=cut(age, breaks= seq(20, 70, by = 10))) %>%
  summarise(TotalCount=n(), NumConversions=sum(conversion)) %>%
  mutate(ConversionRate=NumConversions/TotalCount*100.0)

conversionsByAgeGroup$AgeGroup <-
  as.character(conversionsByAgeGroup$AgeGroup)
conversionsByAgeGroup$AgeGroup[6] <- "70+"

# bar chart
ggplot(conversionsByAgeGroup, aes(x=AgeGroup, y=ConversionRate)) +
  geom_bar(width=0.5, stat="identity") +
  ggtitle('Tỉ lệ chuyển đổi theo nhóm tuổi') +
  xlab("Tuổi") +
  ylab("Tỉ lệ chuyển đổi (%)") +
  theme(plot.title = element_text(hjust = 0.5))
```

Chạm tới AI trong 10 ngày



Sử dụng table1

Bước 1:

Thứ đọc lại dữ liệu trực tiếp từ Internet và xem qua dữ liệu với hàm `glimpse(...)` trong thư viện `dplyr`:

Chạm tới AI trong 10 ngày

```
$ y <chr> "no", "no", "no", "no", "no", "no", "no", "no", "no", "no...
```

Bước 2:

Cài đặt thư viện `table1`:

```
install.packages('table1')
```

Thử phân tích vài số liệu bằng cách gọi hàm `table1` với cú pháp:

- Tham số đầu tiên bắt đầu bằng dấu ~ (đọc là till). Tiếp theo là các biến cần phân tích được ghép với nhau bằng dấu +
- Tham số thứ hai `data=df` cho biết dữ liệu cần phân tích lấy từ biến `df`.

```
library(table1)
table1(~ age + job + education + contact + emp.var.rate +
cons.conf.idx + euribor3m + nr.employed, data=df)
```

Kết quả

| Overall (N=41188) | |
|----------------------|-------------------|
| age | |
| Mean (SD) | 40.0 (10.4) |
| Median [Min, Max] | 38.0 [17.0, 98.0] |
| job | |
| admin. | 10422 (25.3%) |
| blue-collar | 9254 (22.5%) |
| entrepreneur | 1456 (3.5%) |
| housemaid | 1060 (2.6%) |
| management | 2924 (7.1%) |
| retired | 1720 (4.2%) |
| self-employed | 1421 (3.5%) |
| services | 3969 (9.6%) |
| student | 875 (2.1%) |
| technician | 6743 (16.4%) |
| unemployed | 1014 (2.5%) |
| unknown | 330 (0.8%) |
| education | |
| basic.4y | 4176 (10.1%) |
| basic.6y | 2292 (5.6%) |

Chạm tới AI trong 10 ngày

| Overall (N=41188) | |
|----------------------|----------------------|
| basic.9y | 6045 (14.7%) |
| high.school | 9515 (23.1%) |
| illiterate | 18 (0.0%) |
| professional.course | 5243 (12.7%) |
| university.degree | 12168 (29.5%) |
| unknown | 1731 (4.2%) |
| contact | |
| cellular | 26144 (63.5%) |
| telephone | 15044 (36.5%) |
| emp.var.rate | |
| Mean (SD) | 0.0819 (1.57) |
| Median [Min, Max] | 1.10 [-3.40, 1.40] |
| cons.conf.idx | |
| Mean (SD) | -40.5 (4.63) |
| Median [Min, Max] | -41.8 [-50.8, -26.9] |
| euribor3m | |
| Mean (SD) | 3.62 (1.73) |
| Median [Min, Max] | 4.86 [0.634, 5.05] |
| nr.employed | |
| Mean (SD) | 5170 (72.3) |
| Median [Min, Max] | 5190 [4960, 5230] |

Vài diễn giải:

- Tổng số quan sát (số records): N=41188
- Các biến liên tục như age, emp.var.rate, cons.conf.idx, euribor3m, nr.employed thì lệnh table1 báo cáo số Trung bình (Mean), độ lệch chuẩn (SD), Trung vị (Median), Nhỏ nhất (Min), và Lớn nhất (Max).

Ví dụ tuổi trong nghiên cứu này trung bình là 40, độ lệch chuẩn 10.4, trung vị là 38, người thấp tuổi nhất là 17 tuổi, người cao tuổi nhất là 98 tuổi.

| age | |
|-------------------|-------------------|
| Mean (SD) | 40.0 (10.4) |
| Median [Min, Max] | 38.0 [17.0, 98.0] |

- Các biến phân loại (hay định tính, categorical variables) như nghề nghiệp (job), trình độ học vấn (education), hình thức liên lạc

(contact) thì table1 liệt kê các giá trị ở cột bên trái và báo cáo số lượng từng loại, kèm tỉ lệ % so với tổng số N.

Ví dụ nhìn vào báo cáo của job có thể hình dung sơ bộ tỉ lệ công việc: việc “admin.”, chắc là công việc văn phòng nói chung gồm có 10422 người, chiếm 25.3% trong tổng số dữ liệu quan sát; “unknown” có nghĩa là không biết nghề nghiệp (có thể là khi thực hiện nghiên cứu quên hỏi, hoặc quên nhập liệu hoặc người ta không chịu cung cấp) là 330 người chiếm 0.8% - không đáng kể.

| job | |
|---------------|---------------|
| admin. | 10422 (25.3%) |
| blue-collar | 9254 (22.5%) |
| entrepreneur | 1456 (3.5%) |
| housemaid | 1060 (2.6%) |
| management | 2924 (7.1%) |
| retired | 1720 (4.2%) |
| self-employed | 1421 (3.5%) |
| services | 3969 (9.6%) |
| student | 875 (2.1%) |
| technician | 6743 (16.4%) |
| unemployed | 1014 (2.5%) |
| unknown | 330 (0.8%) |

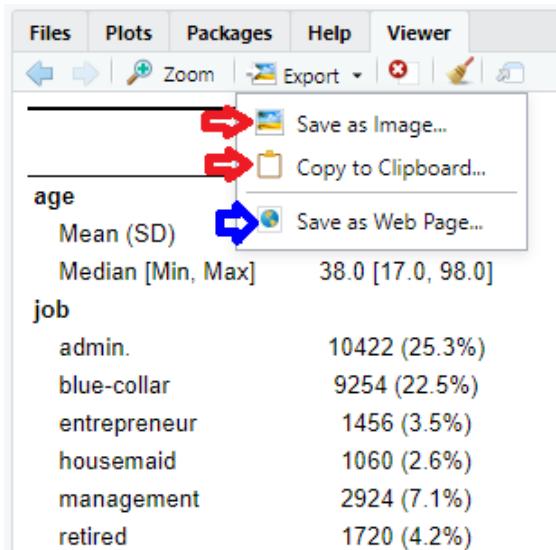
Gợi ý:

- Bạn thử phân tích thêm biến y rồi tự lý giải xem sao?

```
table1(~ age + job + education + contact + emp.var.rate +  
       cons.conf.idx + euribor3m + nr.employed + y, data=df)
```

- Chú ý trong RStudio để lấy được cái hình đưa vào tài liệu theo cách thông thường như dùng menu bên dưới (chỗ mũi tên màu đỏ) thì hình có thể bị cắt mất. Nên dùng chức năng Save as Web Page...để lưu thành trang web rồi copy nội dung vào tài liệu.

Chạm tới AI trong 10 ngày



Thay đổi cú pháp một chút bằng cách dùng dấu | (gọi là dấu more) để phân tích theo một biến phân nhóm. Ví dụ lệnh sau sẽ phân tích các biến age, job, education, emp.var.rate, cons.conf.idx, euribor3m, nr.employed theo các giá trị của hình thức liên lạc (contact): cellular là gọi số di động; telephone là gọi số để bàn.

```
table1(~ age + job + education + emp.var.rate + cons.conf.idx +  
euribor3m + nr.employed | contact, data=df)
```

Kết quả như bên dưới. Cột bên trái là các biến cần phân tích. Nhưng số liệu được phân tích theo các hình thức gọi điện (các giá trị của biến contact)

| | cellular (N=26144) | telephone (N=15044) | Overall (N=41188) |
|-------------------|-----------------------|------------------------|----------------------|
| age | | | |
| Mean (SD) | 40.0 (11.0) | 40.1 (9.43) | 40.0 (10.4) |
| Median [Min, Max] | 38.0 [17.0, 98.0] | 39.0 [18.0, 86.0] | 38.0 [17.0, 98.0] |
| job | | | |
| admin. | 7126 (27.3%) | 3296 (21.9%) | 10422 (25.3%) |
| blue-collar | 5090 (19.5%) | 4164 (27.7%) | 9254 (22.5%) |
| entrepreneur | 855 (3.3%) | 601 (4.0%) | 1456 (3.5%) |
| housemaid | 640 (2.4%) | 420 (2.8%) | 1060 (2.6%) |
| management | 1902 (7.3%) | 1022 (6.8%) | 2924 (7.1%) |
| retired | 1231 (4.7%) | 489 (3.3%) | 1720 (4.2%) |
| self-employed | 893 (3.4%) | 528 (3.5%) | 1421 (3.5%) |
| services | 2311 (8.8%) | 1658 (11.0%) | 3969 (9.6%) |
| student | 671 (2.6%) | 204 (1.4%) | 875 (2.1%) |
| technician | 4637 (17.7%) | 2106 (14.0%) | 6743 (16.4%) |
| unemployed | 620 (2.4%) | 394 (2.6%) | 1014 (2.5%) |
| unknown | 168 (0.6%) | 162 (1.1%) | 330 (0.8%) |

Chạm tới AI trong 10 ngày

| | cellular (N=26144) | telephone (N=15044) | Overall (N=41188) |
|----------------------|-------------------------------|--------------------------------|------------------------------|
| education | | | |
| basic.4y | 2350 (9.0%) | 1826 (12.1%) | 4176 (10.1%) |
| basic.6y | 1247 (4.8%) | 1045 (6.9%) | 2292 (5.6%) |
| basic.9y | 3452 (13.2%) | 2593 (17.2%) | 6045 (14.7%) |
| high.school | 5928 (22.7%) | 3587 (23.8%) | 9515 (23.1%) |
| illiterate | 15 (0.1%) | 3 (0.0%) | 18 (0.0%) |
| professional.course | 3478 (13.3%) | 1765 (11.7%) | 5243 (12.7%) |
| university.degree | 8657 (33.1%) | 3511 (23.3%) | 12168 (29.5%) |
| unknown | 1017 (3.9%) | 714 (4.7%) | 1731 (4.2%) |
| emp.var.rate | | | |
| Mean (SD) | -0.387 (1.66) | 0.897 (0.971) | 0.0819 (1.57) |
| Median [Min, Max] | -0.100 [-3.40, 1.40] | 1.10 [-3.40, 1.40] | 1.10 [-3.40, 1.40] |
| cons.conf.idx | | | |
| Mean (SD) | -41.4 (5.02) | -39.0 (3.35) | -40.5 (4.63) |
| Median [Min, Max] | -42.7 [-50.8, -26.9] | -36.4 [-50.8, -26.9] | -41.8 [-50.8, -26.9] |
| euribor3m | | | |
| Mean (SD) | 3.10 (1.82) | 4.54 (1.09) | 3.62 (1.73) |
| Median [Min, Max] | 4.08 [0.634, 4.97] | 4.86 [0.634, 5.05] | 4.86 [0.634, 5.05] |
| nr.employed | | | |
| Mean (SD) | 5150 (79.2) | 5190 (48.6) | 5170 (72.3) |
| Median [Min, Max] | 5200 [4960, 5230] | 5190 [4960, 5230] | 5190 [4960, 5230] |

Bạn hãy thử phân tích các biến theo trình độ (**education**) rồi diễn giải xem thế nào?

```
table1(~ age + job + contact + emp.var.rate + cons.conf.idx +
euribor3m + nr.employed | education, data=df)
```

Mở rộng một chút, bạn thử phân nhóm theo 2 biến thì như thế nào?

```
table1(~ age + job + emp.var.rate + cons.conf.idx + euribor3m +
nr.employed | education + contact, data=df)
```

Kết quả là table1 trình bày hàng ngang được phân nhóm thành 2 cấp, bên trên là **education**, trong mỗi giá trị của **education** thì gồm các giá trị của **contact** như sau (tôi cắt bớt các cột chỉ chừa lại một cột **university.degree** và vài dòng để minh họa):

Chạm tới AI trong 10 ngày

| | university.degree | | Overall | |
|--------------------|----------------------|-----------------------|-----------------------|------------------------|
| | cellular (N=8657) | telephone (N=3511) | cellular (N=26144) | telephone (N=15044) |
| age | | | | |
| Mean (SD) | 38.7 (9.75) | 39.4 (9.29) | 40.0 (11.0) | 40.1 (9.43) |
| Median [Min, Max] | 36.0 [20.0, 91.0] | 37.0 [22.0, 83.0] | 38.0 [17.0, 98.0] | 39.0 [18.0, 86.0] |
| job | | | | |
| admin. | 4229 (48.9%) | 1524 (43.4%) | 7126 (27.3%) | 3296 (21.9%) |
| blue-collar | 64 (0.7%) | 30 (0.9%) | 5090 (19.5%) | 4164 (27.7%) |
| entrepreneur | 378 (4.4%) | 232 (6.6%) | 855 (3.3%) | 601 (4.0%) |
| housemaid | 87 (1.0%) | 52 (1.5%) | 640 (2.4%) | 420 (2.8%) |
| management | 1408 (16.3%) | 655 (18.7%) | 1902 (7.3%) | 1022 (6.8%) |
| retired | 217 (2.5%) | 68 (1.9%) | 1231 (4.7%) | 489 (3.3%) |
| self-employed | 510 (5.9%) | 255 (7.3%) | 893 (3.4%) | 528 (3.5%) |
| services | 124 (1.4%) | 49 (1.4%) | 2311 (8.8%) | 1658 (11.0%) |
| student | 111 (1.3%) | 59 (1.7%) | 671 (2.6%) | 204 (1.4%) |
| technician | 1319 (15.2%) | 490 (14.0%) | 4637 (17.7%) | 2106 (14.0%) |
| unemployed | 183 (2.1%) | 79 (2.3%) | 620 (2.4%) | 394 (2.6%) |
| unknown | 27 (0.3%) | 18 (0.5%) | 168 (0.6%) | 162 (1.1%) |
| euribor3m | | | | |
| Mean (SD) | 3.19 (1.82) | 4.37 (1.28) | 3.10 (1.82) | 4.54 (1.09) |
| Median [Min, Max] | 4.12 [0.634, 4.97] | 4.86 [0.634, 5.05] | 4.08 [0.634, 4.97] | 4.86 [0.634, 5.05] |
| nr.employed | | | | |
| Mean (SD) | 5150 (82.2) | 5190 (56.9) | 5150 (79.2) | 5190 (48.6) |
| Median [Min, Max] | 5200 [4960, 5230] | 5190 [4960, 5230] | 5200 [4960, 5230] | 5190 [4960, 5230] |

Sử dụng compareGroups

Cài đặt thư viện:

```
install.packages('compareGroups')
```

Chạm tới AI trong 10 ngày

Thứ gọi hàm compareGroups với tham số thứ nhất bắt đầu bằng biến phân nhóm (contact), sau đó là dấu ~, và danh sách các biến cần phân tích cách nhau bởi dấu +. Sau đó gọi tiếp hàm createTable (...) với tham số là kết quả của hàm compareGroups.

```
cg = compareGroups(contact ~ age + education + emp.var.rate +  
cons.conf.idx + euribor3m + nr.employed, data=df)  
createTable(cg)
```

Kết quả như sau:

-----Summary descriptives table by 'contact'-----

| | cellular N=26144 | telephone N=15044 | p.overall |
|---------------------|---------------------|----------------------|-----------|
| age | 40.0 (11.0) | 40.1 (9.43) | 0.138 |
| education: | | | <0.001 |
| basic.4y | 2350 (8.99%) | 1826 (12.1%) | |
| basic.6y | 1247 (4.77%) | 1045 (6.95%) | |
| basic.9y | 3452 (13.2%) | 2593 (17.2%) | |
| high.school | 5928 (22.7%) | 3587 (23.8%) | |
| illiterate | 15 (0.06%) | 3 (0.02%) | |
| professional.course | 3478 (13.3%) | 1765 (11.7%) | |
| university.degree | 8657 (33.1%) | 3511 (23.3%) | |
| unknown | 1017 (3.89%) | 714 (4.75%) | |
| emp.var.rate | -0.39 (1.66) | 0.90 (0.97) | 0.000 |
| cons.conf.idx | -41.39 (5.02) | -38.97 (3.35) | 0.000 |
| euribor3m | 3.10 (1.82) | 4.54 (1.09) | 0.000 |
| nr.employed | 5152 (79.2) | 5193 (48.6) | 0.000 |

Điển giải một chút kết quả:

- Hàm compareGroups và createTable cũng tương tự như hàm table1 với cách phân tích phân nhóm trên. Tức là cột bên trái là các biến cần phân tích, giá trị phân tích theo các giá trị của biến phân nhóm ở cột bên phải.
- Khác với hàm table1, cú pháp phân tích theo nhóm ở đây thì để biến phân nhóm bên trái.
- Điểm khác biệt tiếp theo là có thêm trị số P (cột p.overall).

Thay thay thêm tham số show.p.trend=T trong hàm createTable(...):

```
createTable(cg, show.p.trend = T)
```

Kết quả có thêm cột “p.trend”: trị số P theo trend (tạm thời cách tính như thế nào thì bỏ qua nhé)

-----Summary descriptives table by 'contact'-----

| | cellular N=26144 | telephone N=15044 | p.overall | p.trend |
|--|---------------------|----------------------|-----------|---------|
|--|---------------------|----------------------|-----------|---------|

Chạm tới AI trong 10 ngày

| | | | | |
|---------------------|---------------|---------------|--------|--------|
| age | 40.0 (11.0) | 40.1 (9.43) | 0.138 | 0.138 |
| education: | | | <0.001 | <0.001 |
| basic.4y | 2350 (8.99%) | 1826 (12.1%) | | |
| basic.6y | 1247 (4.77%) | 1045 (6.95%) | | |
| basic.9y | 3452 (13.2%) | 2593 (17.2%) | | |
| high.school | 5928 (22.7%) | 3587 (23.8%) | | |
| illiterate | 15 (0.06%) | 3 (0.02%) | | |
| professional.course | 3478 (13.3%) | 1765 (11.7%) | | |
| university.degree | 8657 (33.1%) | 3511 (23.3%) | | |
| unknown | 1017 (3.89%) | 714 (4.75%) | | |
| emp.var.rate | -0.39 (1.66) | 0.90 (0.97) | 0.000 | 0.000 |
| cons.conf.idx | -41.39 (5.02) | -38.97 (3.35) | 0.000 | 0.000 |
| euribor3m | 3.10 (1.82) | 4.54 (1.09) | 0.000 | 0.000 |
| nr.employed | 5152 (79.2) | 5193 (48.6) | 0.000 | 0.000 |

Quan sát biến emp.var.rate thì thấy độ lệch chuẩn (SD) lớn hơn trị trung bình (mean). Đây là dấu hiệu cho thấy biến này không tuân theo luật phân phối chuẩn. Như vậy dùng số trung bình và độ lệch chuẩn thì không phù hợp. Lúc này bạn thêm tham số method=c (2) cho hàm compareGroups như sau để báo cáo số trung vị (median), bách phân vị 25% (Lower quartile) và bách phân vị 75% (Upper quartile):

```
cg = compareGroups(contact ~ age + education + emp.var.rate +
  cons.conf.idx + euribor3m + nr.employed, method=c(2), data=df)
createTable(cg)
```

-----Summary descriptives table by 'contact'-----

| | cellular N=26144 | telephone N=15044 | p.overall |
|---------------------|------------------------|------------------------|-----------|
| age | 38.0 [32.0;47.0] | 39.0 [33.0;47.0] | <0.001 |
| education: | | | <0.001 |
| basic.4y | 2350 (8.99%) | 1826 (12.1%) | |
| basic.6y | 1247 (4.77%) | 1045 (6.95%) | |
| basic.9y | 3452 (13.2%) | 2593 (17.2%) | |
| high.school | 5928 (22.7%) | 3587 (23.8%) | |
| illiterate | 15 (0.06%) | 3 (0.02%) | |
| professional.course | 3478 (13.3%) | 1765 (11.7%) | |
| university.degree | 8657 (33.1%) | 3511 (23.3%) | |
| unknown | 1017 (3.89%) | 714 (4.75%) | |
| emp.var.rate | -0.10 [-1.80;1.40] | 1.10 [1.10;1.40] | 0.000 |
| cons.conf.idx | -42.70 [-46.20;-36.10] | -36.40 [-41.80;-36.40] | 0.000 |
| euribor3m | 4.08 [1.28;4.96] | 4.86 [4.86;4.96] | <0.001 |
| nr.employed | 5196 [5099;5228] | 5191 [5191;5228] | <0.001 |

Kết quả cho thấy 3 biến age, emp.var.rate, và cons.conf.idx được báo cáo số trung vị và [bách phân vị 25%, bách phân vị 75%]

Gợi ý:

Dùng lệnh ? để đọc thêm tài liệu, ý nghĩa tham số các lệnh này:

```
?compareGroups
?createTable
```

Chạm tới AI trong 10 ngày

Hàm compareGroups có giới hạn là số giá trị của biến phân nhóm lớn hơn 5 thì sẽ báo lỗi như sau (Do biết education có hơn 5 giá trị):

```
cg = compareGroups(education ~ age + job + contact + emp.var.rate +
  cons.conf.idx + euribor3m + nr.employed, data=df)

Error in compareGroups.fit(x = x, y = y, include.label = include.label, : number
  of groups must be less or equal to 5
```

Tham khảo <https://youtu.be/hDQ0T6-i1rk>

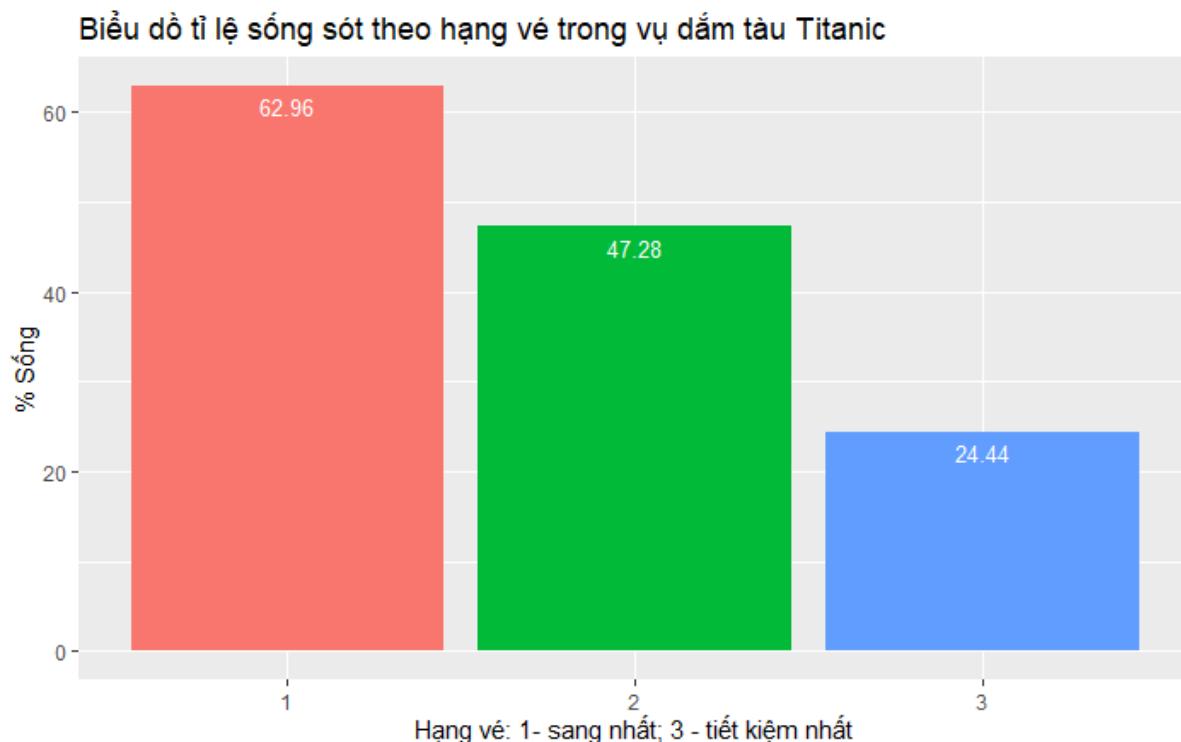
Phân tích Vụ đắm tàu Titanic với R

Đoạn code R sau sẽ đọc dữ liệu trực tiếp từ website của ĐH Standord về vụ đắm tàu Titanic. Sau vụ tai nạn của tàu Titanic thì các nhà nghiên cứu tra lại số liệu để tìm hiểu xem kết quả sống sót của hành khách có liên quan gì đến hạng vé mà họ đã mua không?

```
df =
read.csv('https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv')

library(dplyr)
df_survivedrate = df %>% group_by(Class = Pclass) %>%
  summarise(TotalCount = n(), numSurvived = sum(Survived)) %>%
  mutate(SurvivedRate = numSurvived / TotalCount * 100.0)
df_survivedrate$Class = as.factor(df_survivedrate$Class)

library(ggplot2)
p = ggplot(data = df_survivedrate, aes(x= Class, y = SurvivedRate,
fill=Class))
p = p + geom_bar(stat='identity')
p = p + xlab('Hạng vé: 1- sang nhất; 3 - tiết kiệm nhất') + ylab('%
Sống')
p = p + ggtitle('Biểu đồ tỉ lệ sống sót theo hạng vé trong vụ đắm tàu
Titanic')
p = p + geom_text(aes(label=round(SurvivedRate, 2)), vjust=1.6,
color="white", size=3.5)
p = p + guides(fill=FALSE)
p
```



Biểu đồ cho thấy nếu đi vé hạng sang nhất thì tỉ lệ được cứu sống là gần 63%. Trong khi nếu đi vé rẻ nhất thì cơ may được cứu sống chỉ là 24.44%.

Phân tích Bank Marketing với Python

Tương tự với code R ở phần trước thì 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=';')
```

Quan sát vài dòng dữ liệu bằng hàm head của 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).

Chạm tới AI trong 10 ngày

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 conversi
on
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 |
|--|----------------------|
| df['age'].mean() | 40.02406040594348 |
| df['age'].median() | 38.0 |
| df['age'].mode() | 0 31 dtype: int64 |
| df['age'].quantile() | 38.0 |
| df['age'].min() | 17 |
| df['age'].max() | 98 |
| sd(df\$age) | 10.42125 |
| df['age'].var() | 108.60245116512178 |
| import statistics as st st.stdev(df['age']) | 10.421249980934048 |

Vài 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>.

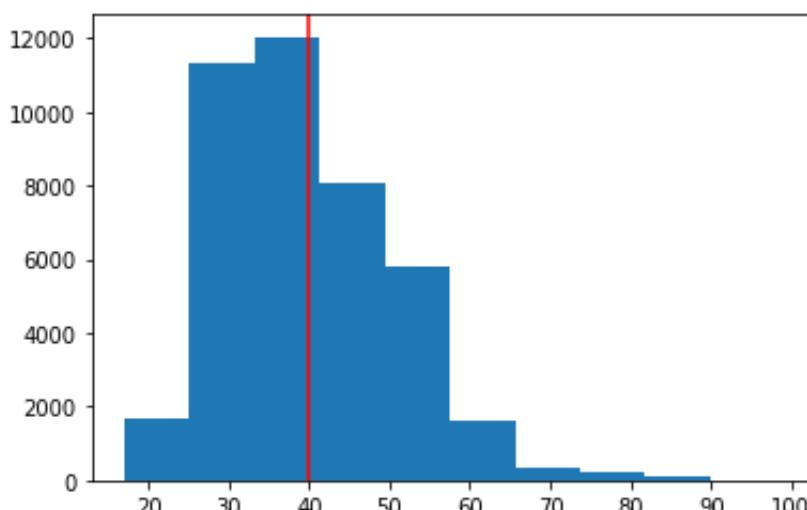
Chạm tới AI trong 10 ngày

- Hàm quantile() trong thư viện pandas của Python trả lại ít thông tin hơn trong R.
- Trong thư viện pandas có sẵn hàm mode. Trong khi R thì chưa có sẵn.
- Trong R thì có sẵn hàm range(df). Trong pandas thì chưa có. Phải dùng hàm min() và max().
- Hàm tính độ lệch chuẩn cũng không có sẵn trong thư viện pandas.

Xem histogram tuổi và đường trung bình:

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

plt.show()
```

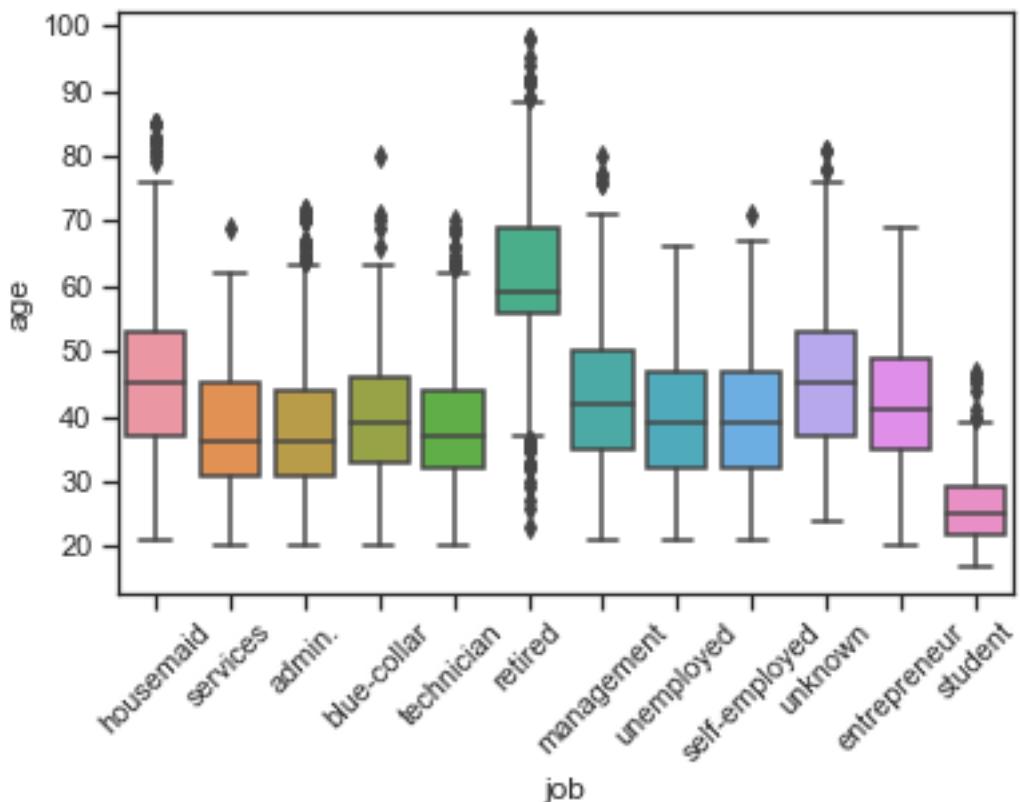


Xem phân bố tuổi theo công việc

```
import pandas as pd
import seaborn as sns

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

df.head()
ax = sns.boxplot(x = 'job', y="age", data = df)
ax.set_xticklabels(ax.get_xticklabels(), rotation=45)
```



So với thư viện ggplots trong R thì thư viện seaborn trong Python vẽ biểu đồ boxplot không đẹp bằng.

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**.

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 mảng 2 giá trị gồm số dòng và số cột của data frame. Phần tử của mảng được đánh số từ 0. Để lấy ra số dòng thì lấy phần tử thứ nhất của mảng thì 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:

Chạm tới AI trong 10 ngày

```
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:

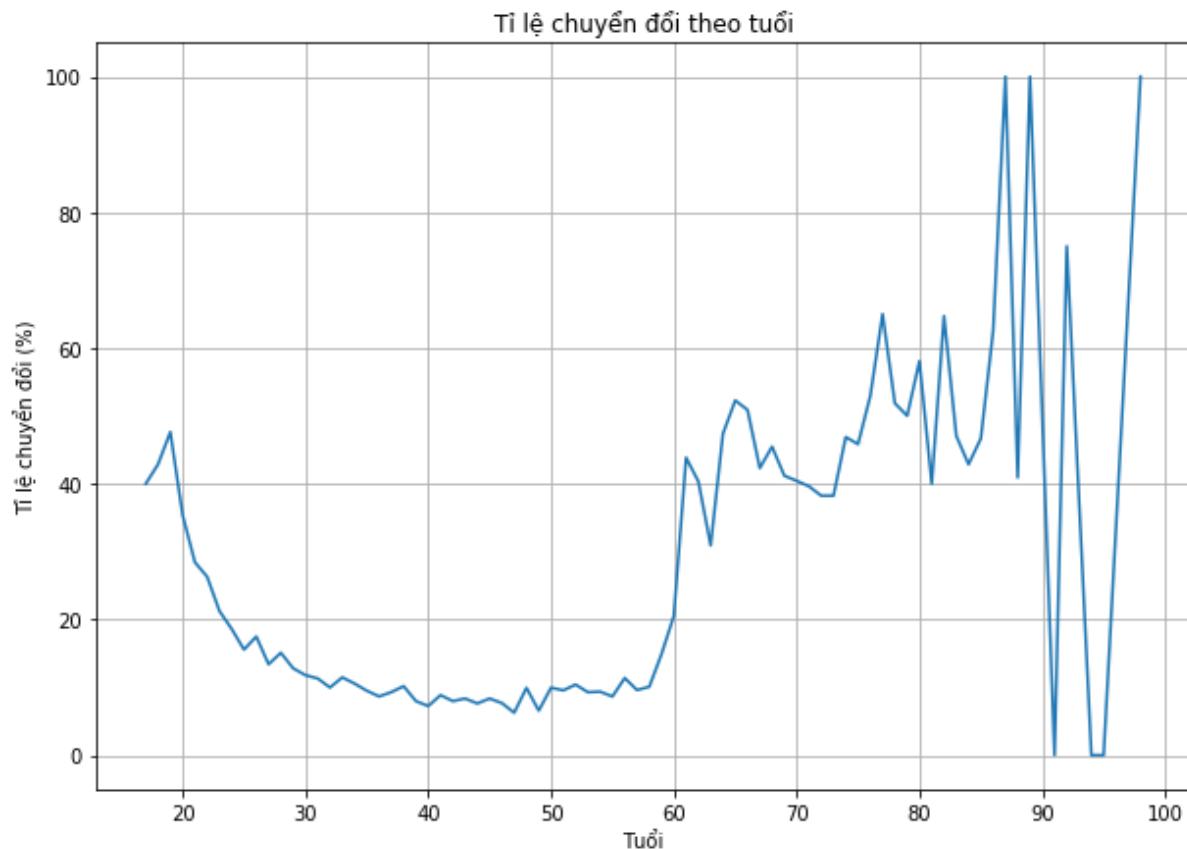
```
conversions_by_age = df.groupby(by='age')['conversion'].sum() / df.groupby(by='age')['conversion'].count() * 100.0

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()
```

Chạm tới AI trong 10 ngày



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:

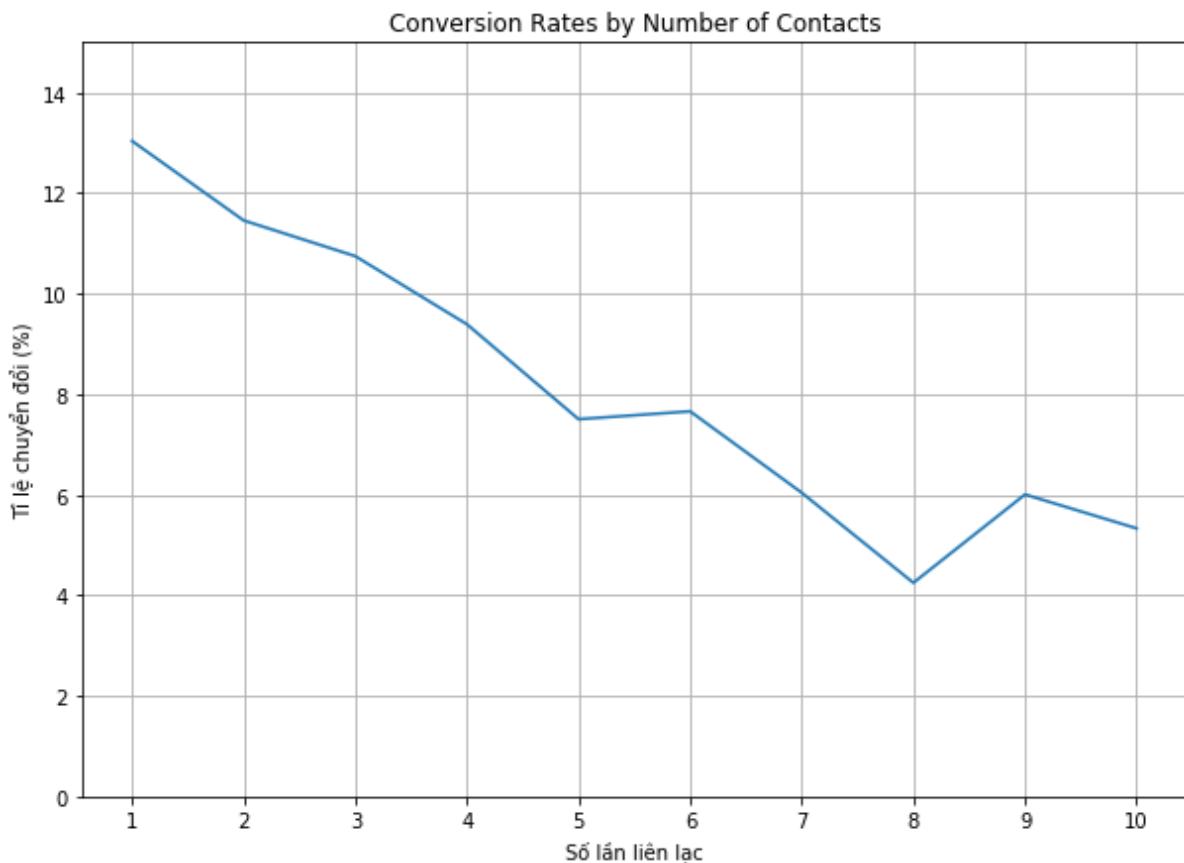
```
conversions_by_contacts = sum_conversion_by_campaign /  
count_conversion_by_campaign * 100.0
```

Xem biểu đồ:

```
import matplotlib.pyplot as plt  
ax = conversions_by_contacts[:10].plot(
```

Chạm tới AI trong 10 ngày

```
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:

```
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+'  
)
```

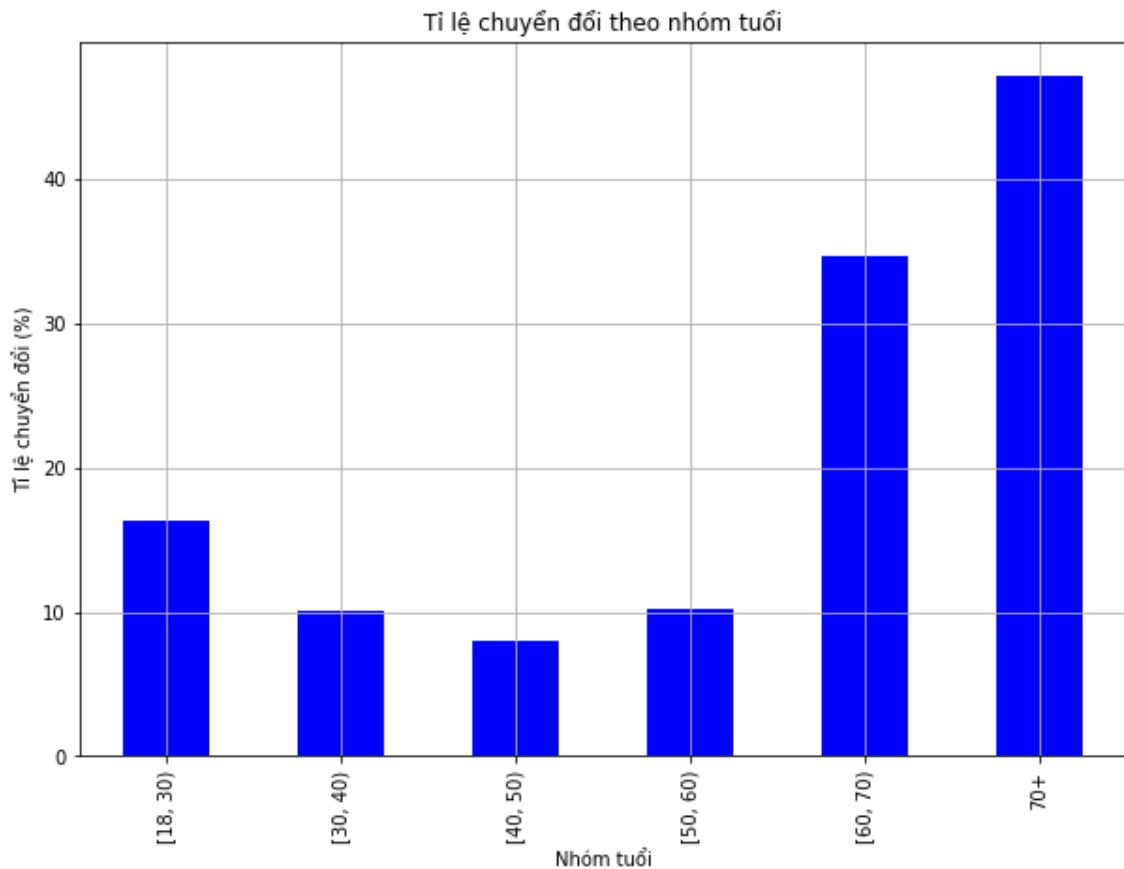
Chú ý dấu \ ở cuối mỗi dòng ý nói chưa kết thúc lệnh.

Chạm tới AI trong 10 ngày

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='Ti lệ chuyển đổi theo nhóm tuổi'  
)  
  
ax.set_xlabel('Nhóm tuổi')  
ax.set_ylabel('Ti lệ chuyển đổi (%)')  
  
plt.show()
```

Chạm tới AI trong 10 ngày



Phân tích Vụ đắm tàu Titanic với Python

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

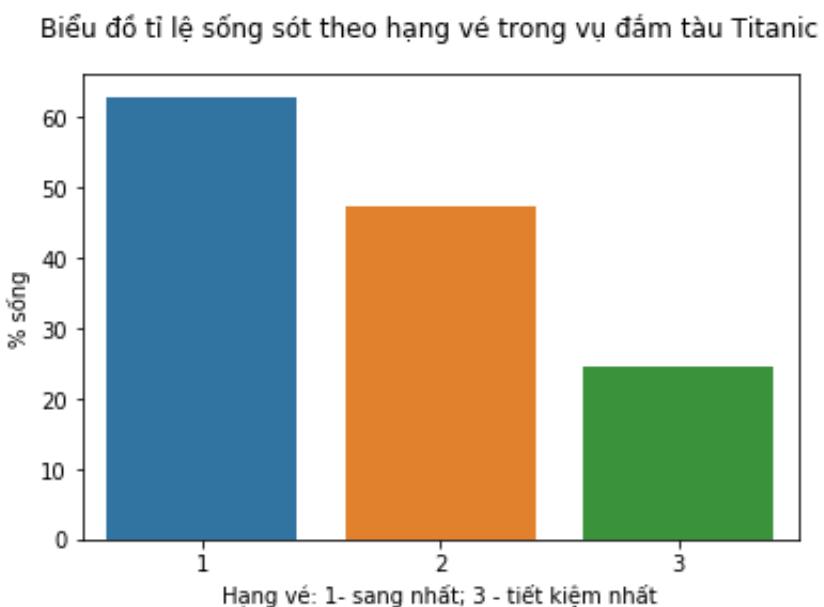
df =
pd.read_csv('https://web.stanford.edu/class/archive/cs/cs109/cs109.116
6/stuff/titanic.csv')
df.head()

rateSurvived_by_Pclass = df.groupby(by='Pclass')['Survived'].sum() /
df.groupby(
    by='Pclass'
) ['Survived'].count() * 100.0

rateSurvived_by_Pclass
```

Chạm tới AI trong 10 ngày

```
plt.figure().suptitle('Biểu đồ tần số sống sót theo hạng vé trong vụ  
đắm tàu Titanic')  
sns.barplot([1, 2, 3], rateSurvived_by_Pclass)  
plt.xlabel('Hạng vé: 1- sang nhất; 3 - tiết kiệm nhất')  
plt.ylabel('% sống')
```



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 tôi vào 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 PMC⁷ về một loại thuốc Zoledronate và gãy xương.

⁷ <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2324066/>

Chạm tới AI trong 10 ngày

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 lá⁸. 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 là sánh hai tỉ lệ. Phần tiếp theo sẽ cùng nhau khám phá ý tưởng đằng sau các phương pháp và cách sử dụng R và Python.

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.

⁸ <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 thời mới hay?

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 cõi, 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:

$$s = \sqrt{\frac{p(1-p)}{N}}$$

Chạm tới AI trong 10 ngày

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 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 lương⁹ 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 hẹp 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 2000 đô 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

⁹ 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.

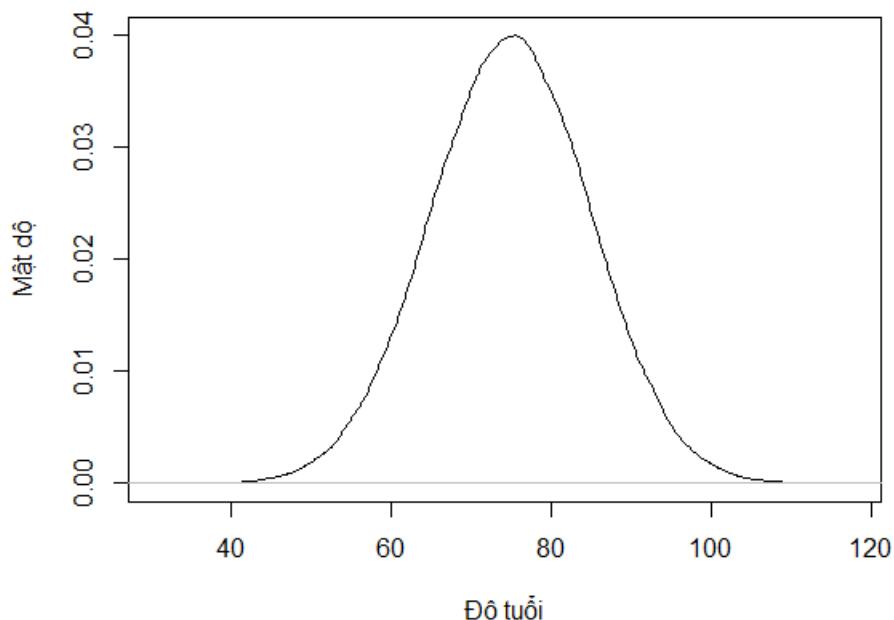
Đối với một biến X , gọi x (nhỏ) là tất cả các giá trị của X thì cộng hết tất cả $f(x)$ là thì bạn đoán là bao nhiêu? Nhớ là $f(x)$ là xác suất của X khi $X = x$. Xác suất là khả năng xảy ra của biến cố. Khả năng thì xảy ra từ 0% đến 100%. Tức là $f(x)$ có giá trị từ 0.0 đến 1.0. Vì thế hình dung Tổng($f(x)$) 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, tạo 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:

Minh họa Phân bố xác suất tuổi thọ của người Việt Nam



Từ thực tế như vậy nên hàm normal distribution $f(x)$ được nhà toán học Gauss phát triển 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 được 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 sử dụng.

Một câu hỏi tiếp theo là làm cách nào để biết dữ liệu có phải là phân bố chuẩn hay không?

Trong R 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?

Chạm tới AI trong 10 ngày

```
df = read.csv('https://thachln.github.io/datasets/bank/bank-additional-full.csv', sep=';')
shapiro.test(df$age[0:5000])
```

Kết quả:

```
Shapiro-Wilk normality test
data: df$age[0:5000]
W = 0.97337, p-value < 2.2e-16
```

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 5000 dòng rồi chạy lại Shapiro test xem sao:

```
shapiro.test(sample(df$age, 5000))
```

Bạn có thể thử nhiều lần, kết quả trị số p vẫn nhỏ hơn 0.05.

Một phương pháp khác là Anderson-Darling, không giới hạn dữ liệu:

```
packages <- c('nortest')
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

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

#Anderson-Darling normality test
library(nortest)
ad.test(df$age)
```

Kết quả
Anderson-Darling normality test
data: df\$age
A = 444.73, p-value < 2.2e-16

Trị số p vẫn nhỏ hơn 0.05. Vì thế có thể kết luận dữ liệu age trong dự án Bank Marketing không tuân theo luật phân bố chuẩn.

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 |
| <i>Độ lệch chuẩn tính bằng:</i> | $\sqrt{\frac{0.131(1 - 0.131)}{1062}}$ | $\sqrt{\frac{0.086(1 - 0.086)}{1065}}$ |
| Hiệu số ảnh hưởng d = | $0.131 - 0.086 = \mathbf{0.045}$ | |
| Độ lệch chuẩn của d. s = | $\sqrt{0.01035^2 + 0.00861^2} = \mathbf{0.0135}$ | |
| KTC 95% | $0.045 \pm 1.96 \times s = \mathbf{0.0186} \sim \mathbf{0.714}$ | |
| 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 R:

```
prop.test(x=c(139, 92), n=c(1062, 1065))

 2-sample test for equality of proportions with continuity correction

data: c(139, 92) out of c(1062, 1065)
X-squared = 10.422, df = 1, p-value = 0.001245
alternative hypothesis: two.sided
95 percent confidence interval:
 0.01717529 0.07182500
sample estimates:
 prop 1    prop 2 
0.13088512 0.08638498
```

Một số số liệu trong kết quả R và cách tính tay ở trên có sai số là do làm tròn các số lẻ.

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. Hàm **prop.test** trong R không cho biết tỉ số z test.
- **pval**: trị số P. Theo cách tính tay, R và cả 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 toàn 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 kíp 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 trong đó hay không?

Ví dụ trong một công ty làm phần mềm A sau khi nhân viên qua đào tạo và đưa vào dự án làm chính thức. Đơn vị 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 cỡ nhỏ, tức là các biến cỡ 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 hẹn 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 R và 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

Bài tôi trình bày một ứng dụng tưởng tượng 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 Coordinator¹⁰ (DC). DC triển khai câu hỏi này rõ hơn một chút:

- Giả định toàn bộ kết quả của nhân viên được nộp đầy đủ lên máy chủ GIT¹¹ của công ty mỗi ngày, mỗi tuần. Theo quy định của công ty thì ngày

¹⁰ 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 Analytics, Data Scientist.

¹¹ 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>.

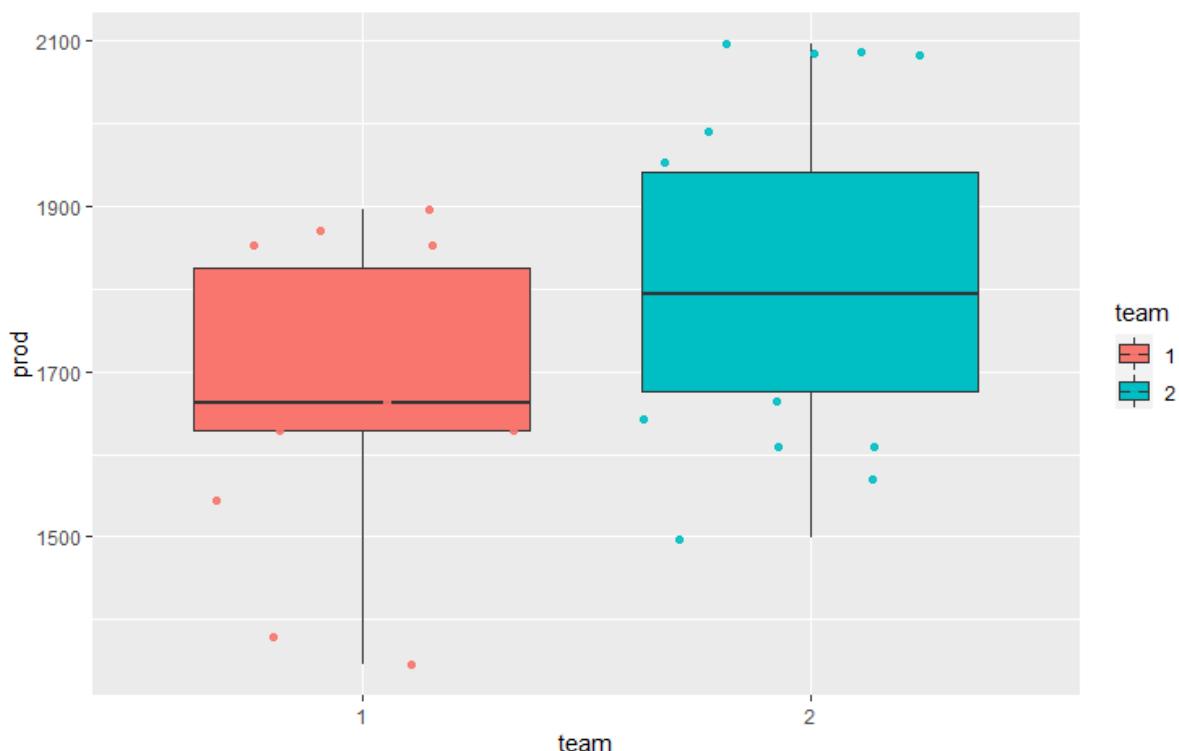
cuối tuần toàn bộ sản phẩm (mã nguồn, tài liệu) phải được nhân viên nộp (Push) lên máy chủ theo đúng quy trình để các bộ phận khác có thể học hỏi và góp ý giúp mình. Đặc biệt là bộ phận Đánh Giá Tài Sản có thể ước lượng tài sản mềm của công ty.

- 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 R sau đọc file dữ liệu và vẽ biểu đồ boxplot về năng suất lập trình (biến prod) giữa hai nhóm (biết team).

```
df =  
read.csv('https://thachln.github.io/datasets/sample_dev_prod.  
csv', header = T)  
  
df$team = as.factor(df$team)  
  
  
library(ggplot2)  
p = ggplot(df, aes(x=team, y=prod, fill=team))  
p = p + geom_boxplot()  
p = p + geom_jitter(aes(color=team), size=1.5, alpha=0.9)  
p
```



Phản 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**

Chạm tới AI trong 10 ngày

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 aov trong R để phân tích phương sai như sau:

```
av = aov(prod ~ team, data=df)
summary(av)
```

Kết quả:

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) | | | | | | |
|----------------|-----|---------|---------|---------|----------|-----|------|------|-----|-------|---|
| team | 1 | 154356 | 154356 | 4.923 | 0.0331 * | | | | | | |
| Residuals | 35 | 1097359 | 31353 | | | | | | | | |
| | --- | | | | | | | | | | |
| Signif. codes: | 0 | '***' | 0.001 | '**' | 0.01 | '*' | 0.05 | '. ' | 0.1 | ' . ' | 1 |

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 Residuals). Với điểm định F = 4.923 và trị số P = 0.0331, chúng ta kết luận rằng có sự khác biệt giữa hai nhóm về năng suất lập trình.

Ghi chú: bạn thấy hàm aov được sử dụng với tham số prod ~ 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.

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 mô cách đơn giản hơn là dùng kiểm định t (t-test).

```
t.test(df$prod ~ df$team)

Welch Two Sample t-test

data: df$prod by df$team
t = -2.2506, df = 31.668, p-value = 0.03149
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-250.67064 -12.43845
sample estimates:
mean in group 1 mean in group 2
1685.400      1816.955
```

Năng suất trung bình của team 1 là 1685 LOCs/man.month và của team 2 là 1817 LOCs/man.month. Do đó mức độ khác biệt giữa hai team là 132 (Team 1 có năng suất trung bình thấp hơn team 2 là 132 LOCs/man.month). Khoảng tin cậy của khác biệt là -250.67 đến -12.44. 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 12 đến 251 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).

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 hay 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 = subset(df, df$team==1)
t.test(df1$prod, mu=2000)
```

```
df2 = subset(df, df$team==2)
t.test(df2$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

Giả 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:

```
df1 = subset(df, df$team==1)
t.test(df1$prod, mu=2000)
```

One Sample t-test

```
data: df1$prod
t = -7.1991, df = 14, p-value = 4.571e-06
```

Chạm tới AI trong 10 ngày

```
alternative hypothesis: true mean is not equal to 2000
95 percent confidence interval:
 1591.673 1779.127
sample estimates:
mean of x
 1685.4
```

Kết quả cho thấy trị số $P = 4.571e-06$, thấp hơn mức α (0.05) đã xác định.

Và với Năng suất trung bình của team 1 là 1685.4 LOCs/month dương hiên là thấp hơn mức tiêu chuẩn 2000 LOCs/man.month. Đồng thời KTC95% giao động từ 1592 đến 1779 (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 R như sau:

```
df2 = subset(df, df$team==2)
t.test(df2$prod, mu=2000)

One Sample t-test

data: df2$prod
t = -4.7149, df = 21, p-value = 0.0001178
alternative hypothesis: true mean is not equal to 2000
95 percent confidence interval:
1736.219 1897.691
sample estimates:
mean of x
1816.955
```

Phản 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?

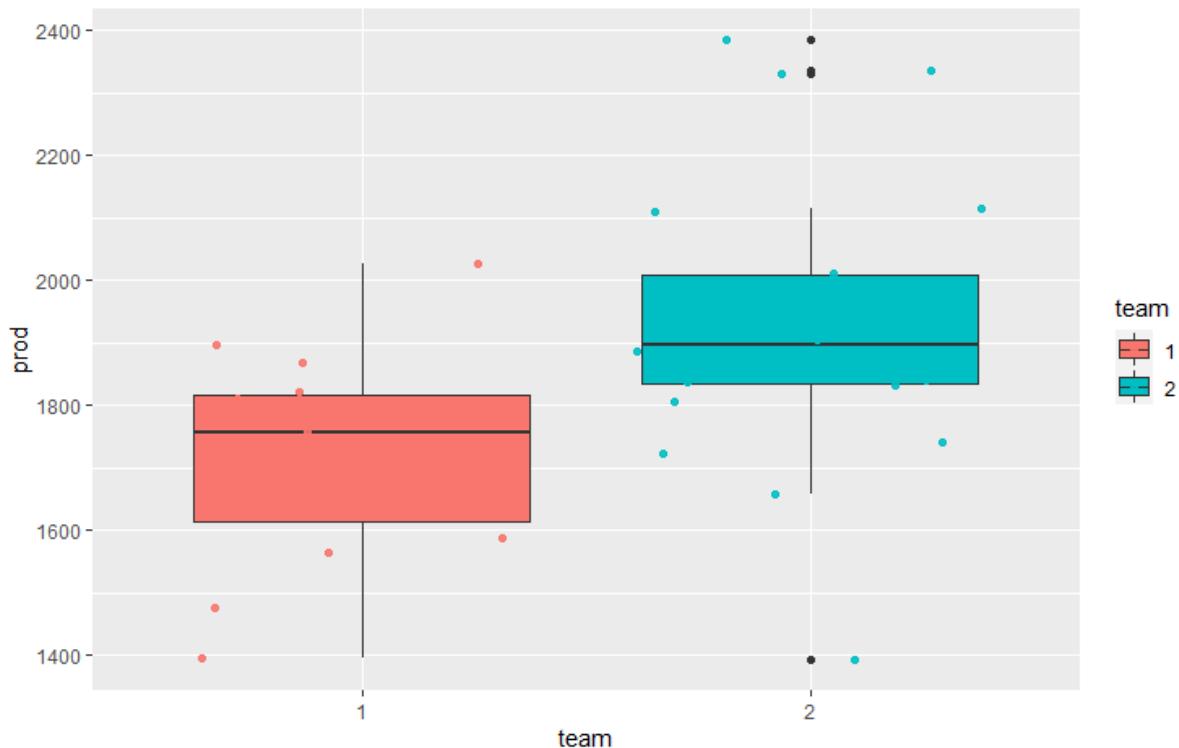
Cụ thể là hãy phân tích dữ liệu thu thập được ở đây:

https://thachln.github.io/datasets/sample_dev_prod_2.csv.

```
df =
read.csv('https://thachln.github.io/datasets/sample_dev_prod_
2.csv', header = T)
df$team = as.factor(df$team)

library(ggplot2)
p = ggplot(df, aes(x=team, y=prod, fill=team))
p = p + geom_boxplot()
p = p + geom_jitter(aes(color=team), size=1.5, alpha=0.9)
p
```

Chạm tới AI trong 10 ngày



Sử dụng t.test trong R để so sánh năng suất của team 2 với năng suất tiêu chuẩn 2000 LOCs/man.month:

```
df2 = subset(df, df$team==2)
t.test(df2$prod, mu=2000)
```

Kết quả như sau:

```
One Sample t-test
data: df2$prod
t = -1.4189, df = 21, p-value = 0.1706
alternative hypothesis: true mean is not equal to 2000
95 percent confidence interval:
1828.410 2032.408
sample estimates:
mean of x
1930.409
```

Năng suất trung bình của team 2 là 1930 LOCs/man.month rõ ràng là nhỏ hơn tiêu chuẩn 2000 LOCs/man.month. Tuy nhiên KTC95% dao động từ 1828 đến 2032, không phải đều nhỏ hơn 2000. Đồng thời P value 0.1705 lớn hơn trị số α (0.05) đã thiết lập.

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).

Ngày 4 – Chủ đề: Dữ liệu lớn

Ngày hôm nay sẽ bàn tinh 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 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 17: Giới thiệu cách lưu trữ dữ liệu trong các phần mềm quản lý dữ liệu chuyên nghiệp, minh họa bằng phần mềm mã nguồn mở¹² 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 18: Hướng dẫn sử dụng Ubuntu cơ bản cho người chưa biết.

Bài 19: Hướng dẫn cài đặt Hadoop để có thể làm quen, trải nghiệm.

Bài 20: 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.

¹² 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 17: 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.

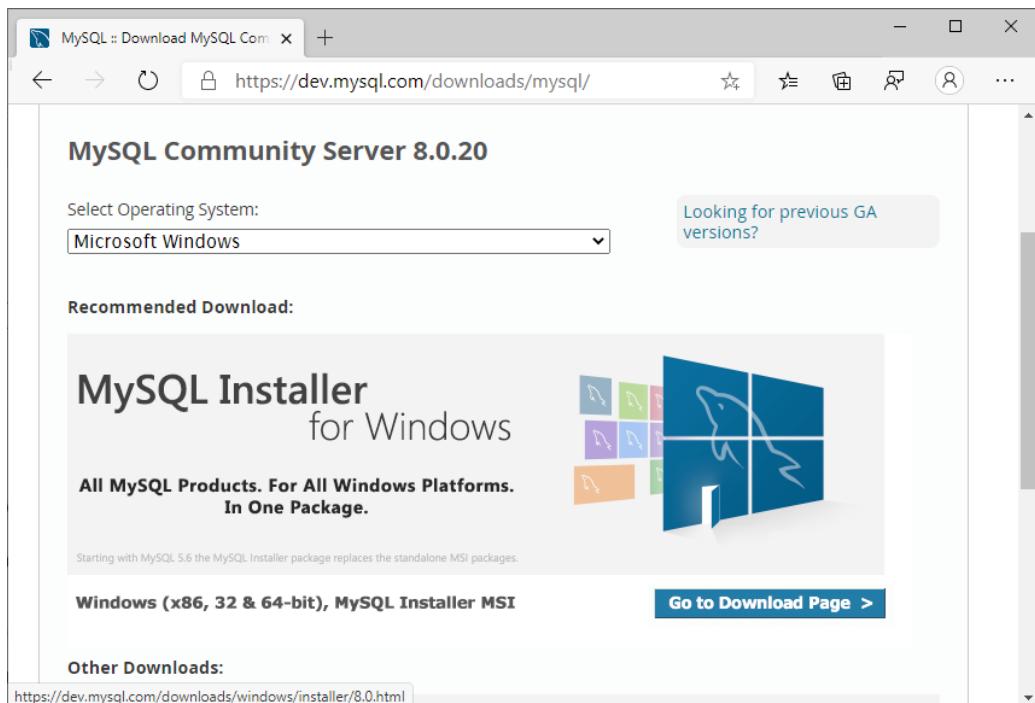
Trong các hệ thống xử lý thông tin thì sẽ có các phần mềm chuyên dụng để lưu trữ, quản lý dữ liệu chặt chẽ hơn. Thông thường dữ liệu mà các bạn xử lý là dữ liệu dạng bảng (Table), trong đó gồm các cột (Column, hoặc Field – trường dữ liệu). Loại dữ liệu này rất thích hợp để lưu trữ trong các phần mềm 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.

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)

Chạm tới AI trong 10 ngày



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:

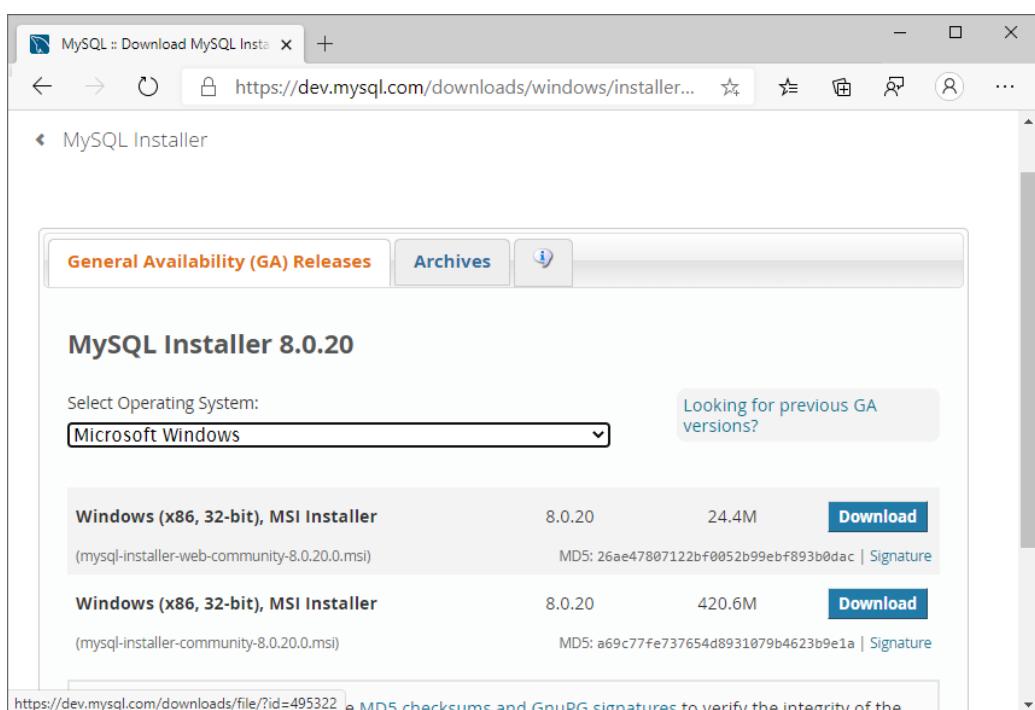
MySQL Installer for Windows

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

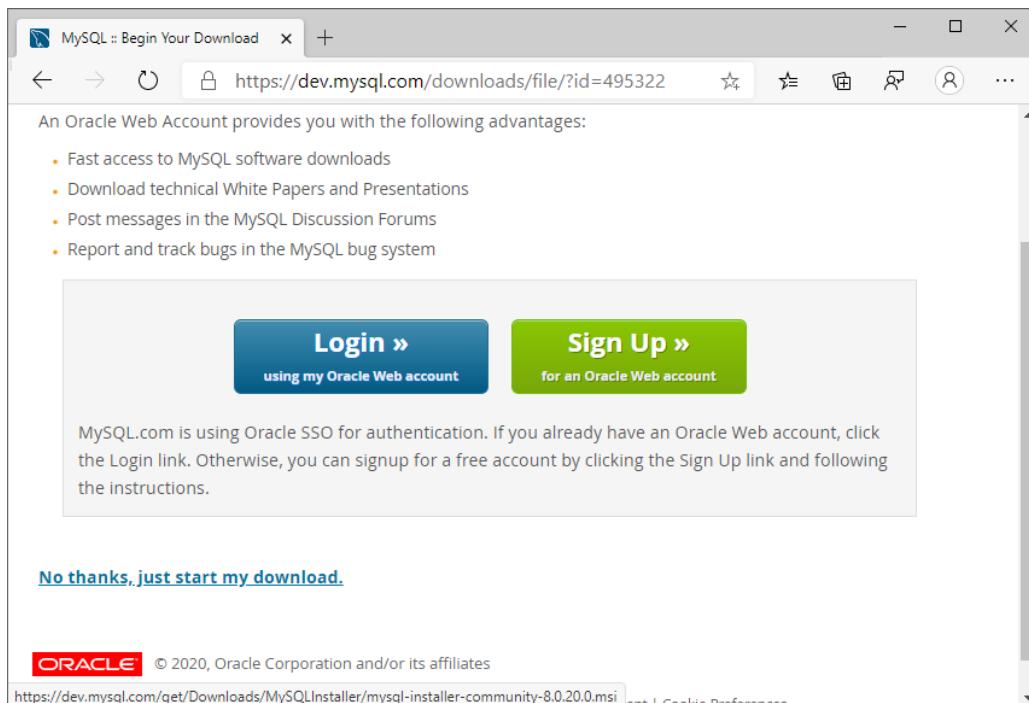
Select Operating System: Microsoft Windows Looking for previous GA versions?

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

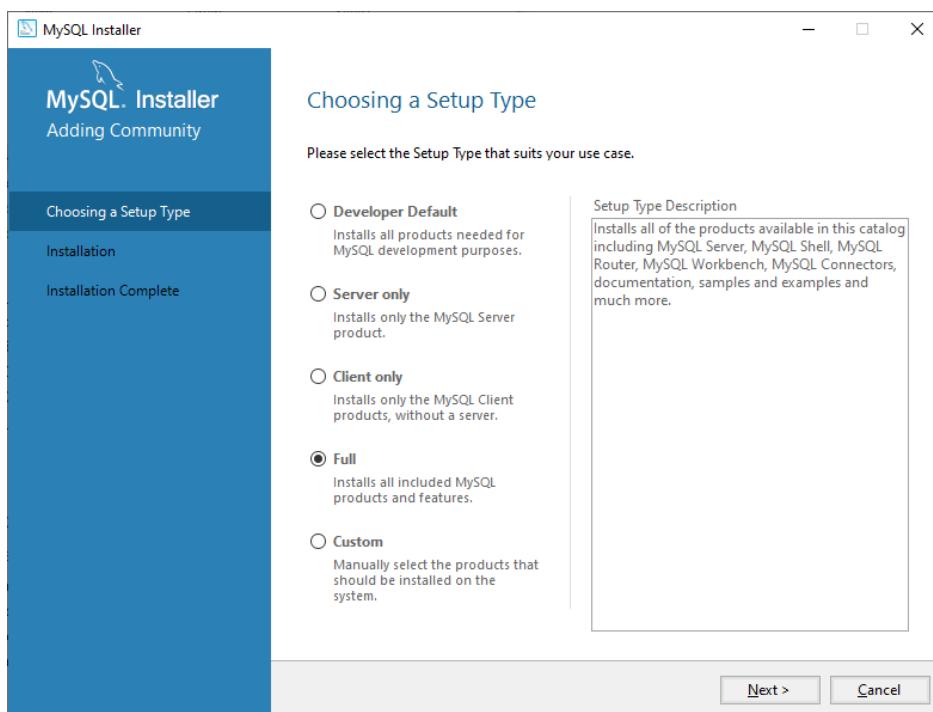
https://dev.mysql.com/downloads/file/?id=495322 e MD5 checksums and GnuPG signatures to verify the integrity of the

Chạm tới AI trong 10 ngày



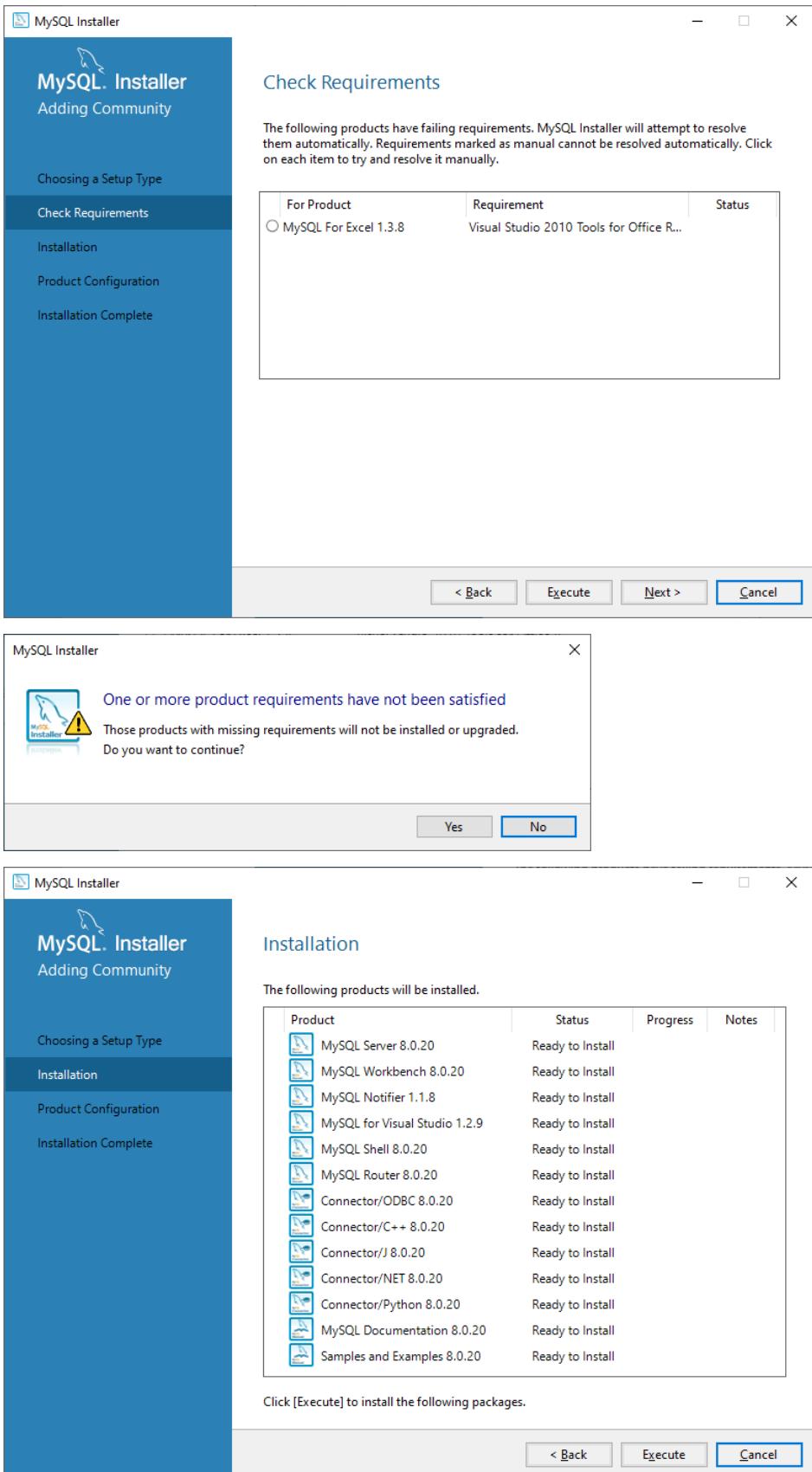
Sau khi tải file “mysql-installer-community-8.0.20.0.msi” về máy, bạn double-click vào nó để cài đặt.

Trong màn hình đầu tiên, bạn cho chế độ cài Full (đầy đủ) để làm quen.

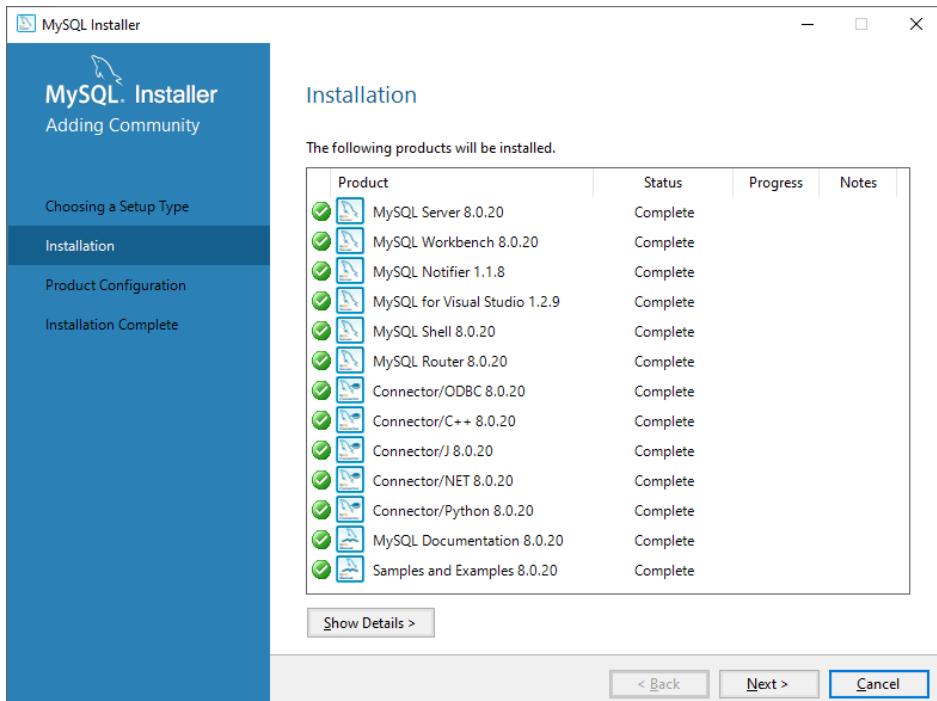


Các màn hình tiếp theo bạn nhấn Next và Yes.

Chạm tới AI trong 10 ngày



Chạm tới AI trong 10 ngày

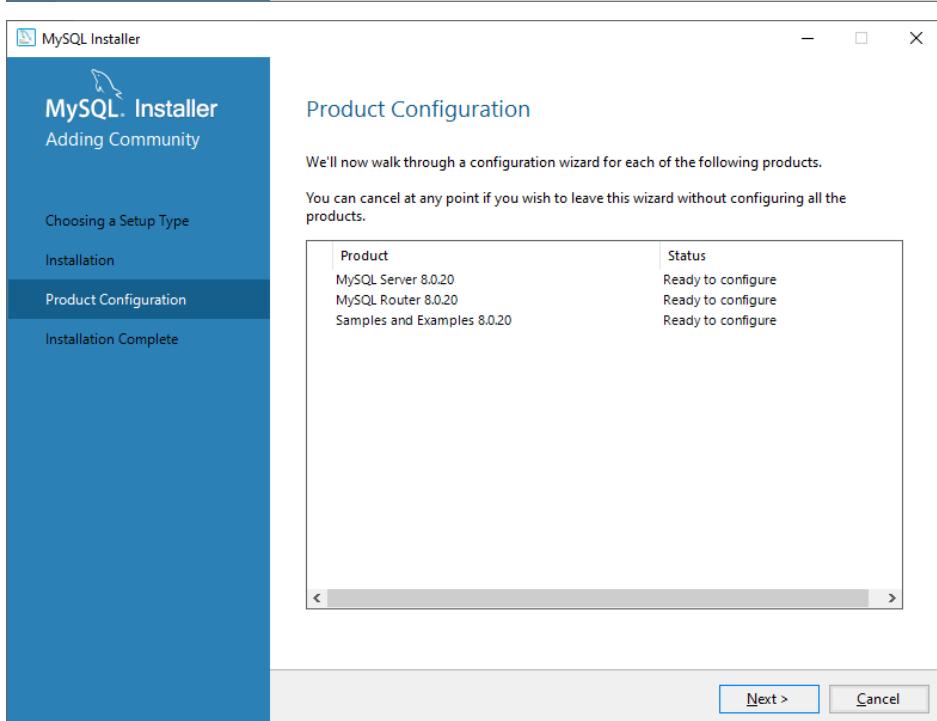


The screenshot shows the MySQL Installer window titled "Installation". On the left, a sidebar lists "Choosing a Setup Type", "Installation" (which is selected), "Product Configuration", and "Installation Complete". The main area displays a table of installed products:

| 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 "Product Configuration". On the left, the sidebar shows "Choosing a Setup Type", "Installation" (selected), "Product Configuration", and "Installation Complete". The main area displays a table of products ready to configure:

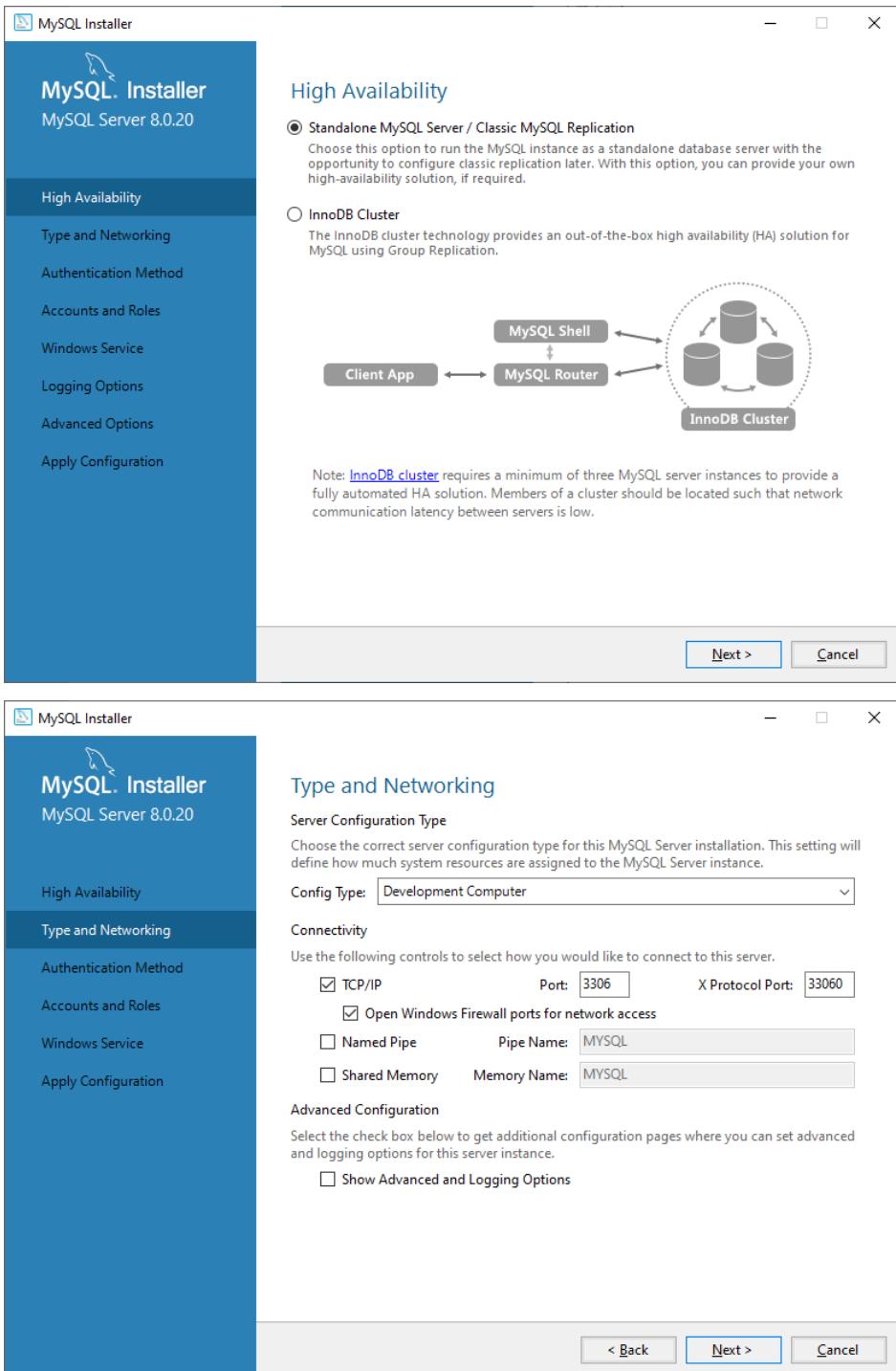
| Product | Status |
|-----------------------------|--------------------|
| MySQL Server 8.0.20 | Ready to configure |
| MySQL Router 8.0.20 | Ready to configure |
| Samples and Examples 8.0.20 | Ready to configure |

We'll now walk through a configuration wizard for each of the following products.

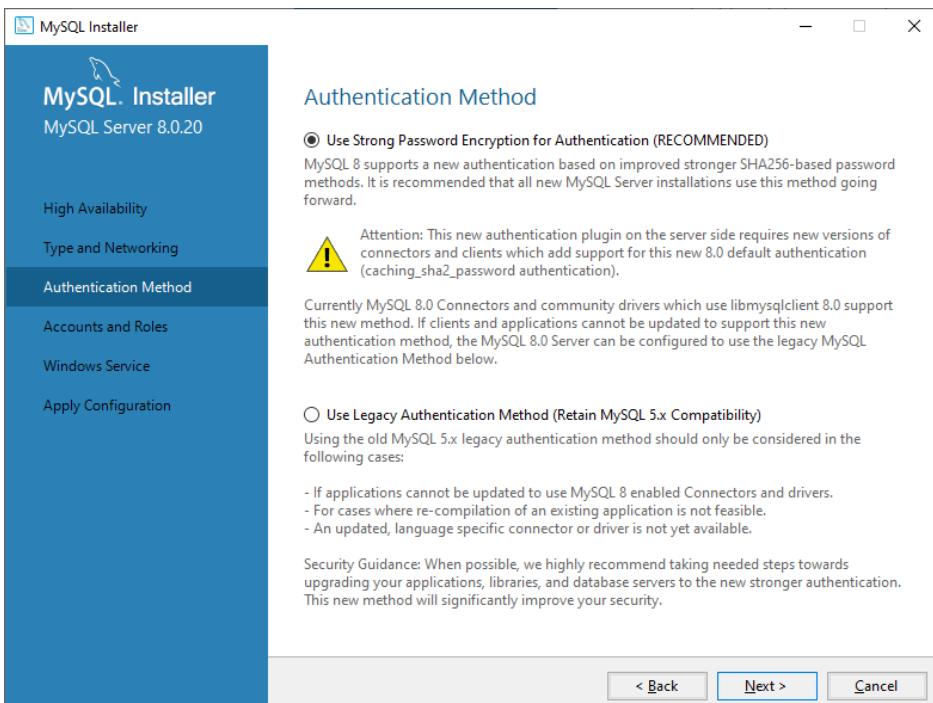
You can cancel at any point if you wish to leave this wizard without configuring all the products.

[Next >](#) [Cancel](#)

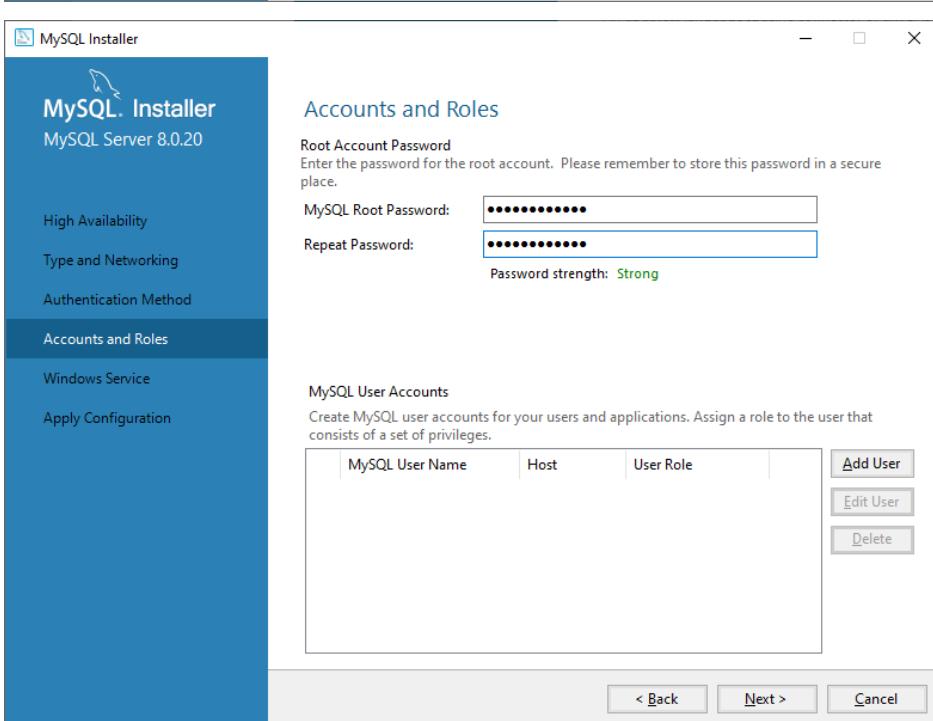
Chạm tới AI trong 10 ngày



Chạm tới AI trong 10 ngày

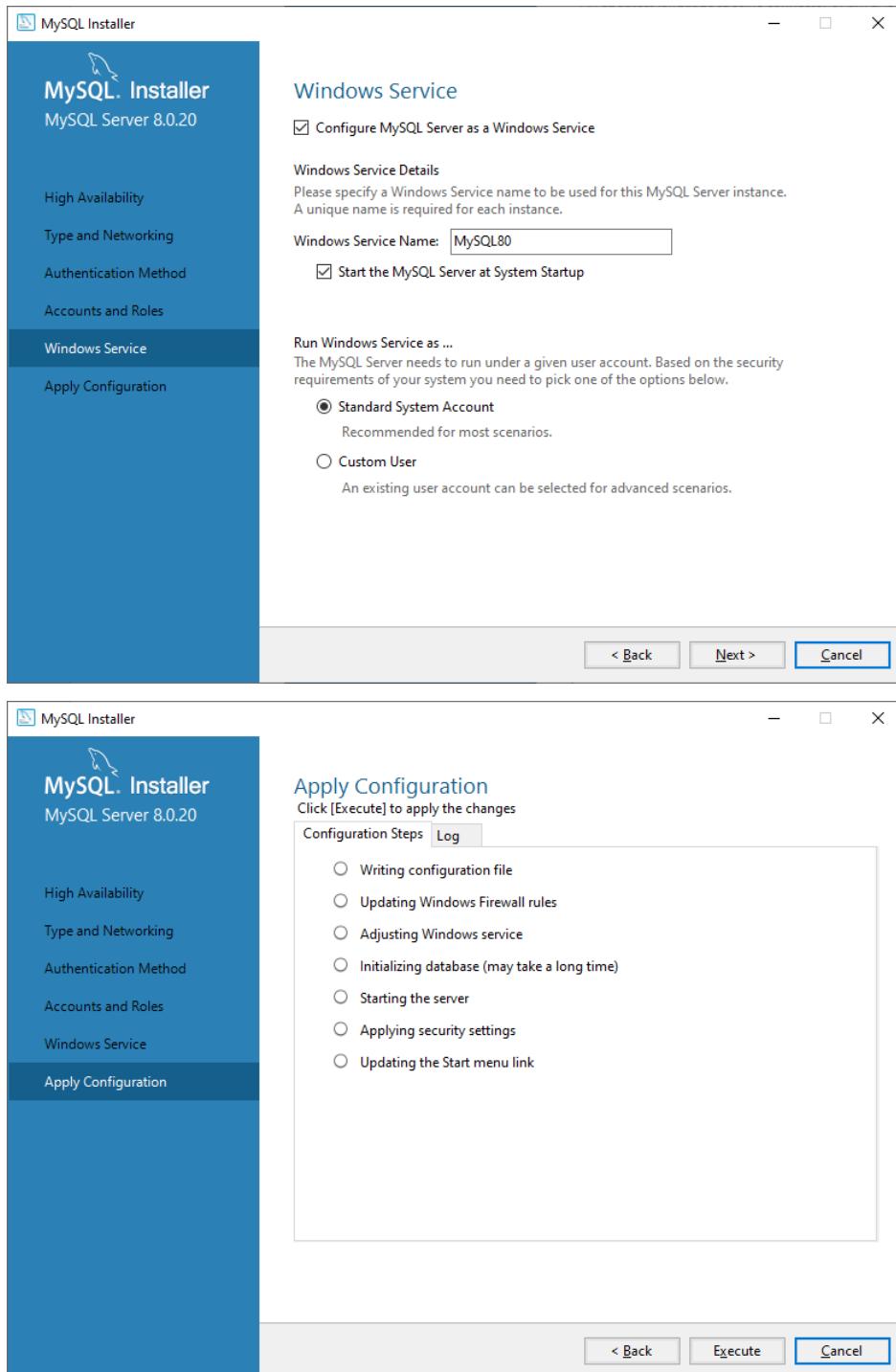


The screenshot shows the MySQL Installer for MySQL Server 8.0.20. The left sidebar has tabs for High Availability, Type and Networking, Authentication Method (which is selected), Accounts and Roles, Windows Service, and Apply Configuration. The main panel title is "Authentication Method". It contains 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 that MySQL 8 supports a new authentication method based on SHA256, and it is recommended for new installations. A warning icon indicates that new connectors and clients need support for the new authentication plugin. Another note says that if clients and applications cannot be updated, the legacy MySQL Authentication Method can be used. A "Security Guidance" section encourages upgrading to the new authentication method for better security. At the bottom are "Back", "Next >", and "Cancel" buttons.

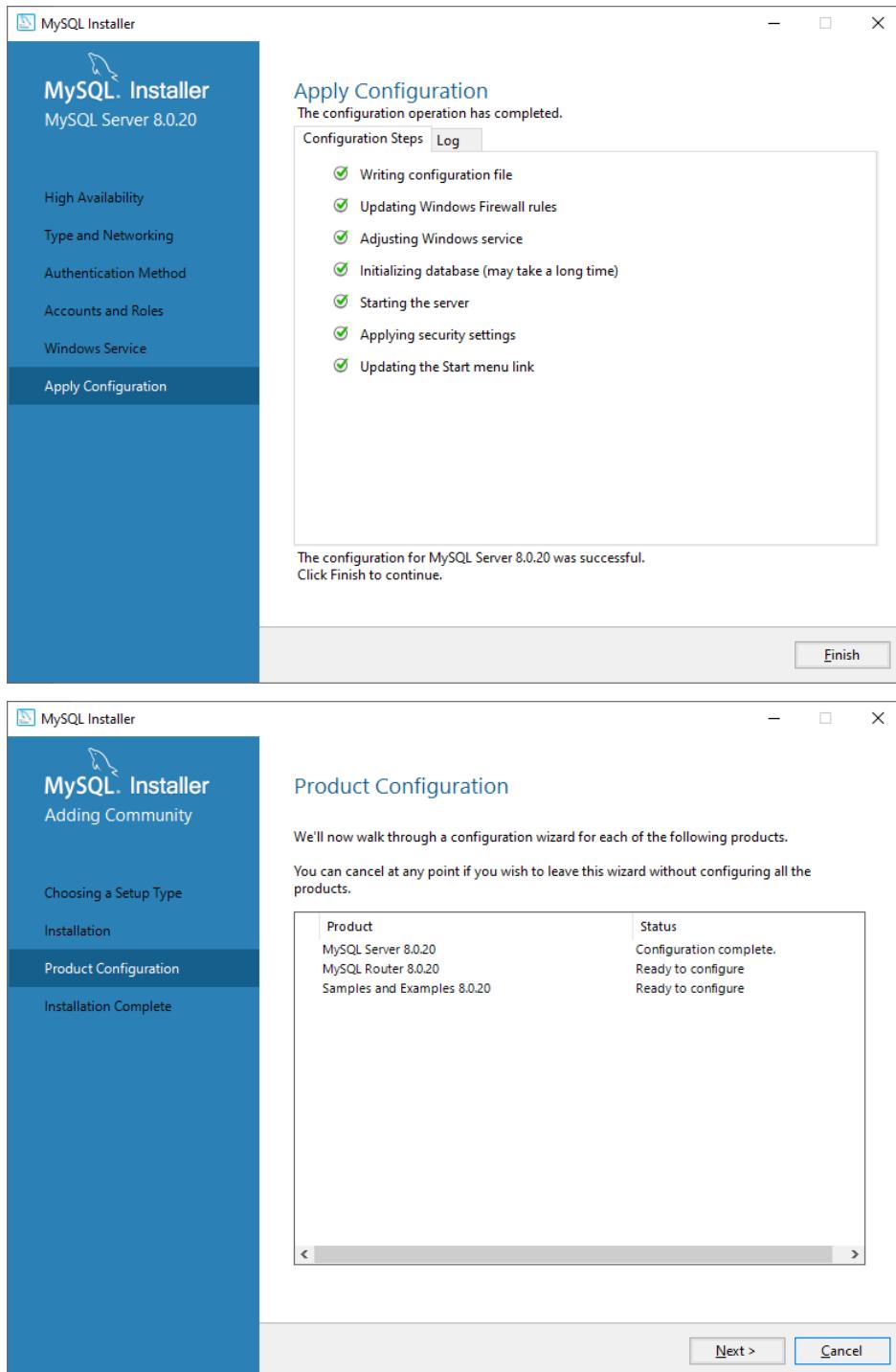


The screenshot shows the MySQL Installer for MySQL Server 8.0.20. The left sidebar has tabs for High Availability, Type and Networking, Authentication Method (selected), Accounts and Roles, Windows Service, and Apply Configuration. The main panel title is "Accounts and Roles". It includes sections for "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 "Back", "Next >", and "Cancel" buttons.

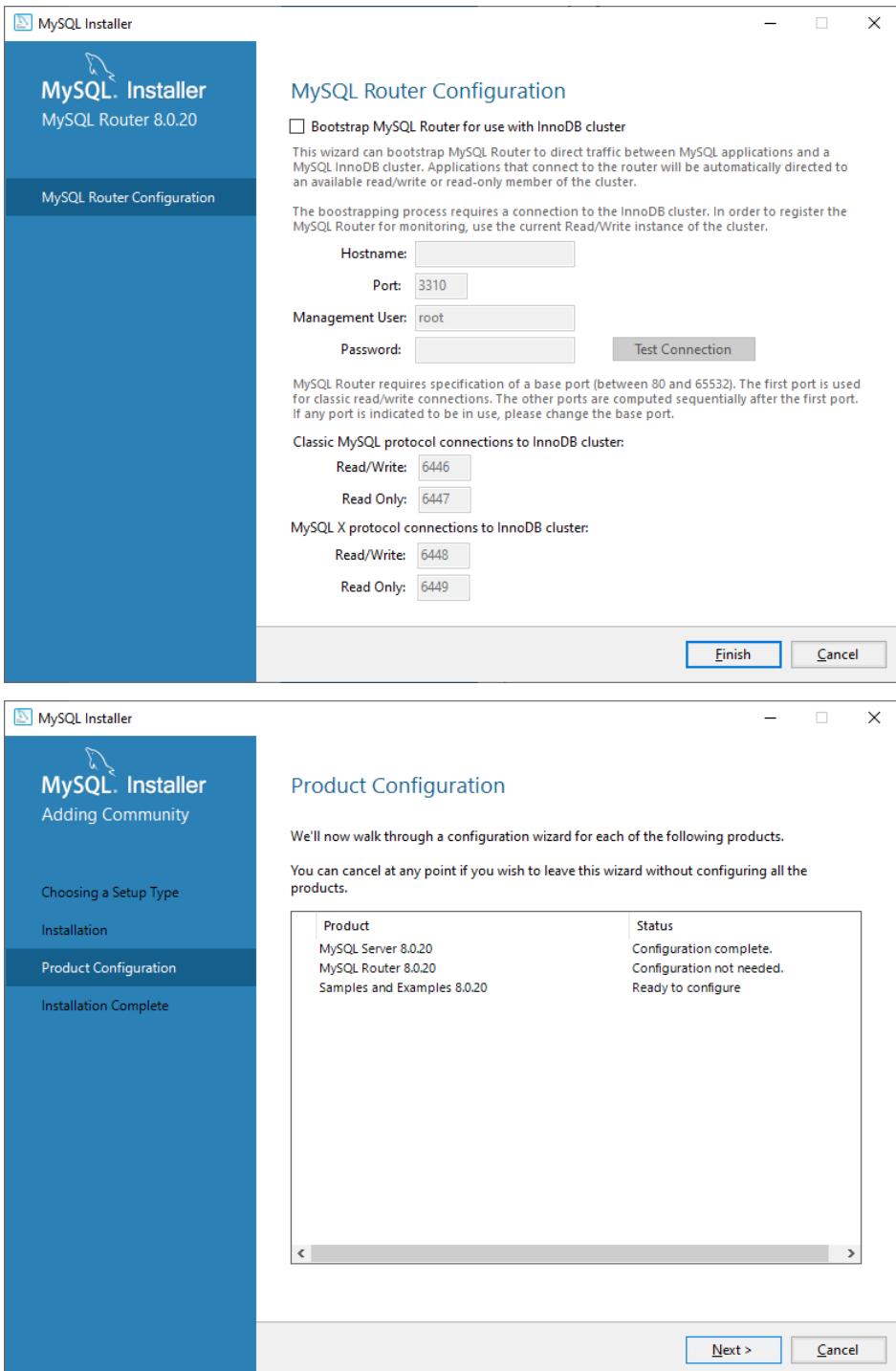
Chạm tới AI trong 10 ngày



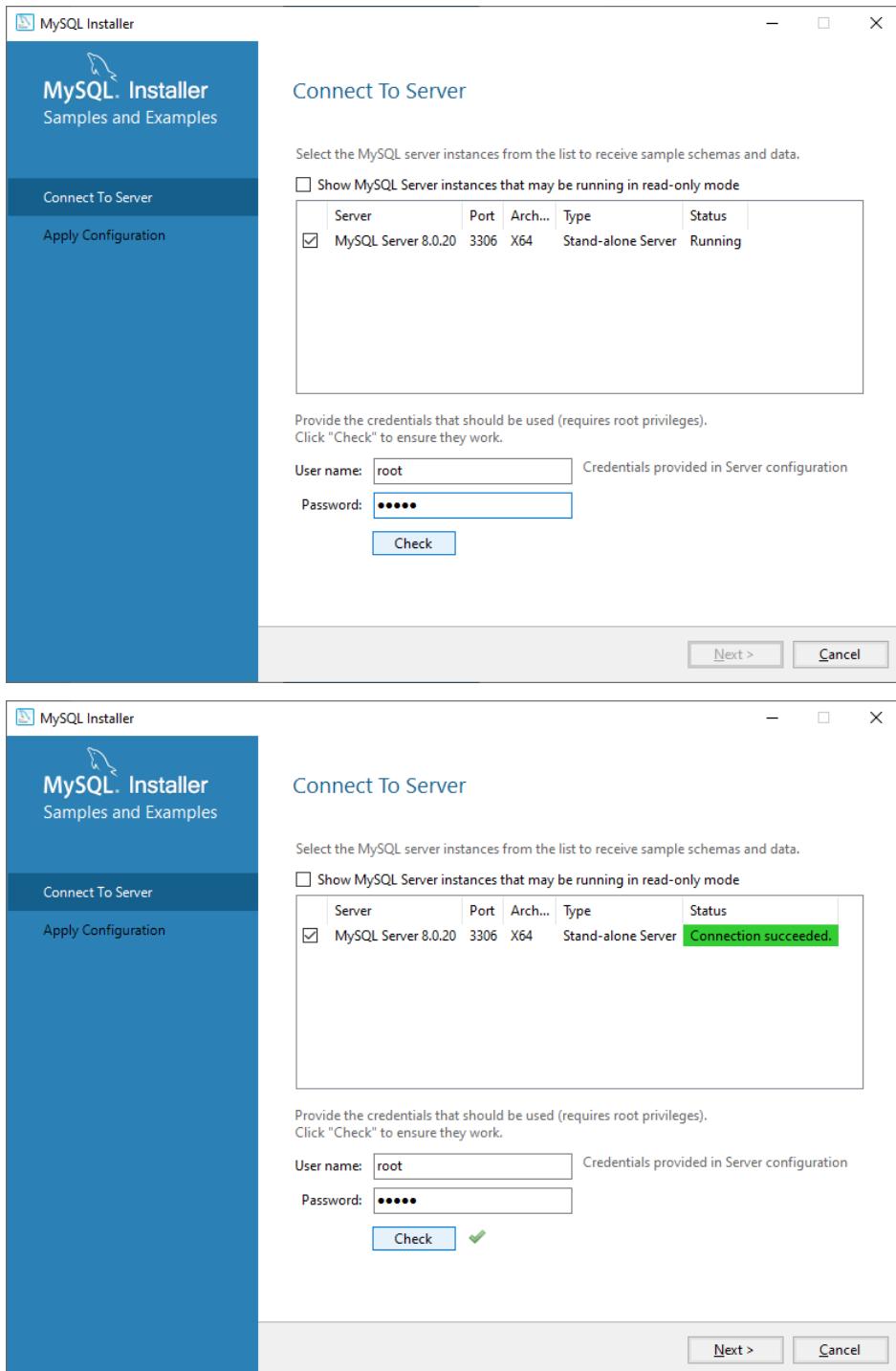
Chạm tới AI trong 10 ngày



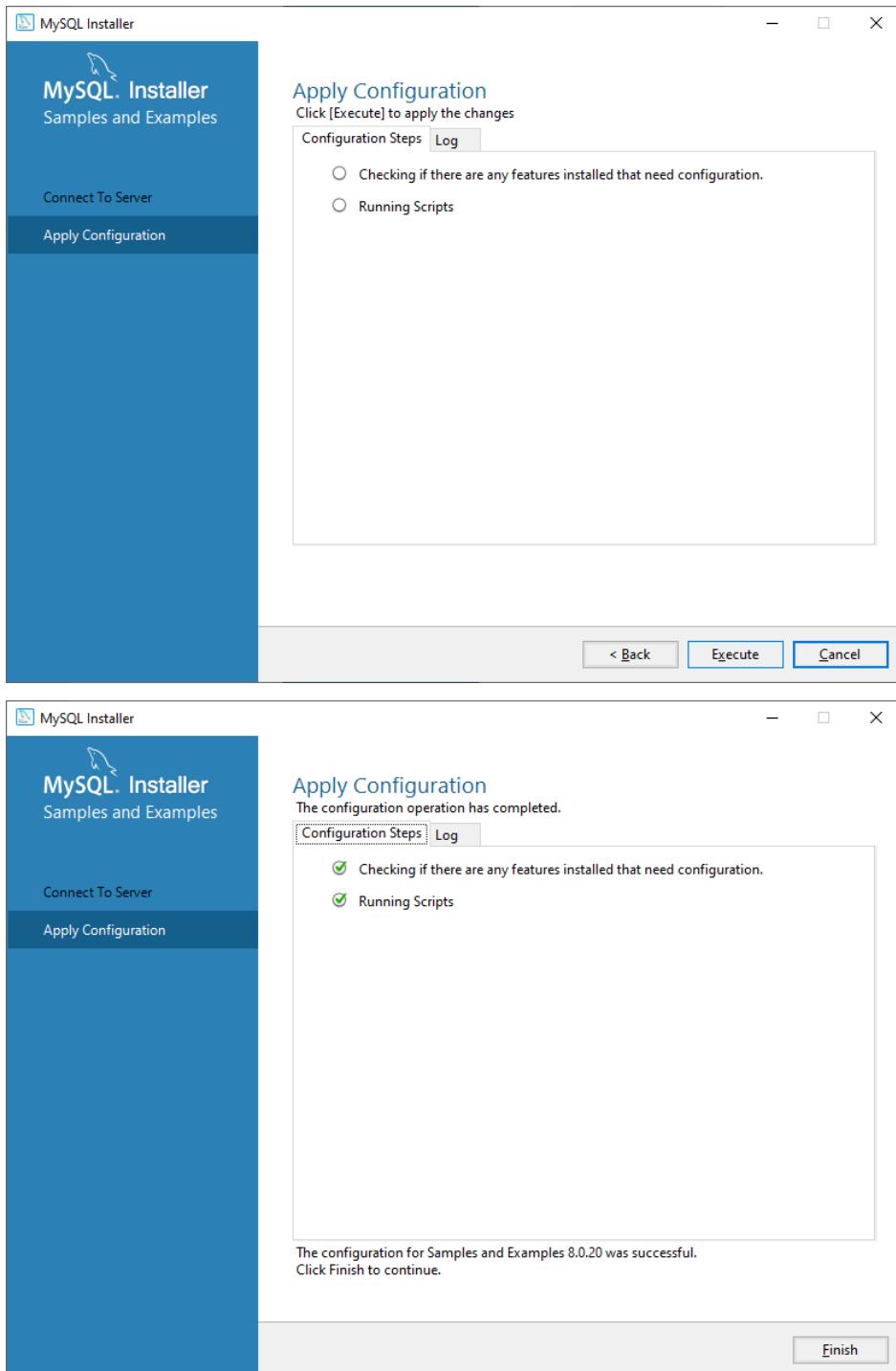
Chạm tới AI trong 10 ngày



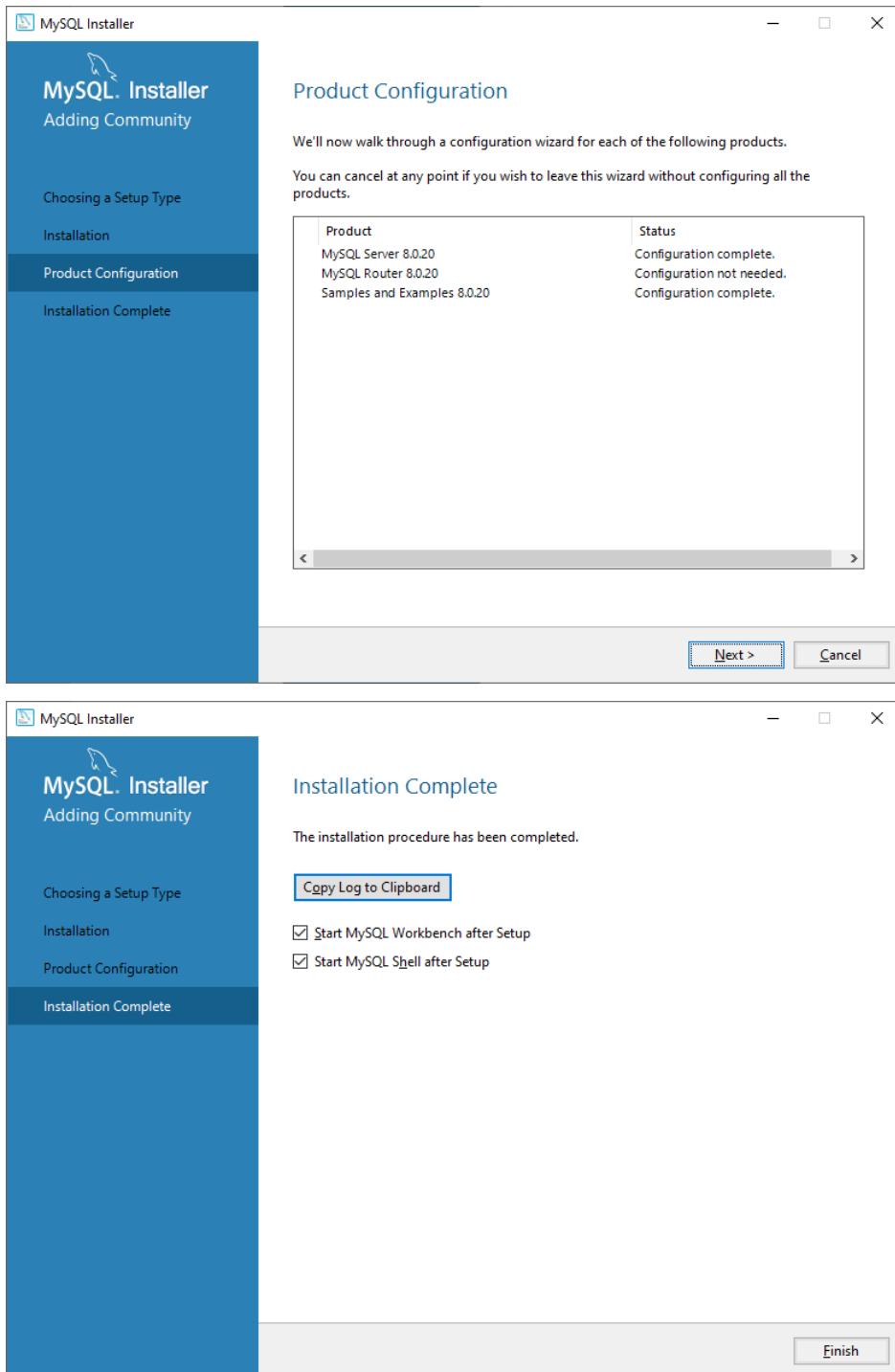
Chạm tới AI trong 10 ngày



Chạm tới AI trong 10 ngày

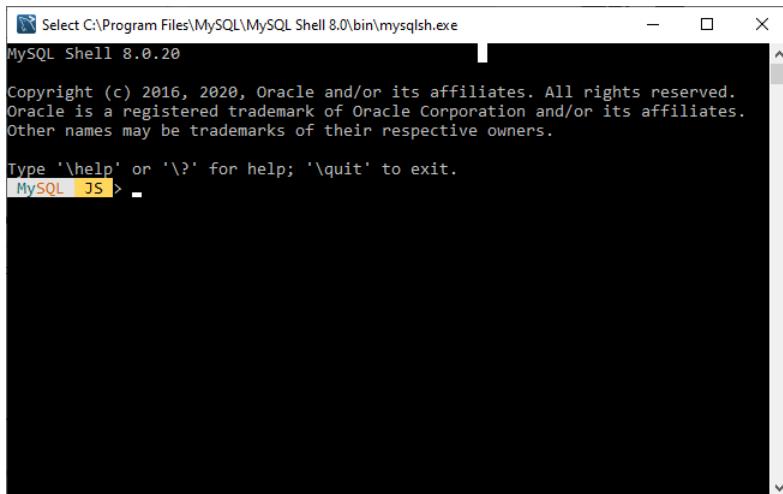


Chạm tới AI trong 10 ngày

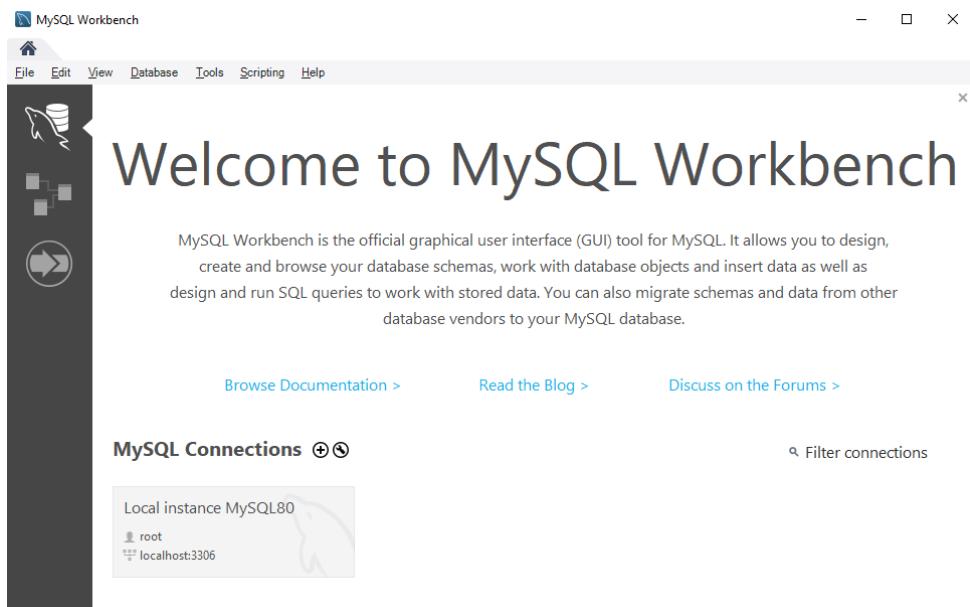


Sau khi bấm Finished thì 2 cùa sổ tương ứng của 2 chức năng trong MySQL sẽ hiển thị ra như sau:

Chạm tới AI trong 10 ngày



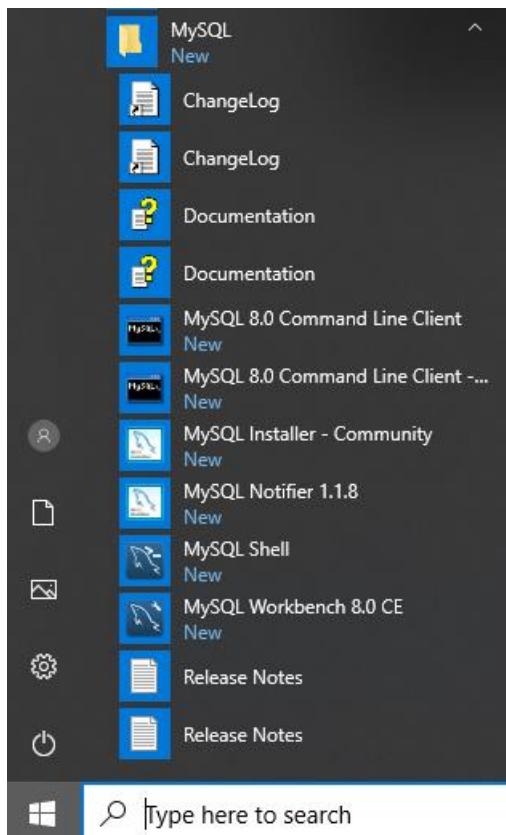
Gõ \quit và nhấn Enter để thoát khỏi cửa sổ màu đen này. Chúng ta sẽ sử dụng sau.



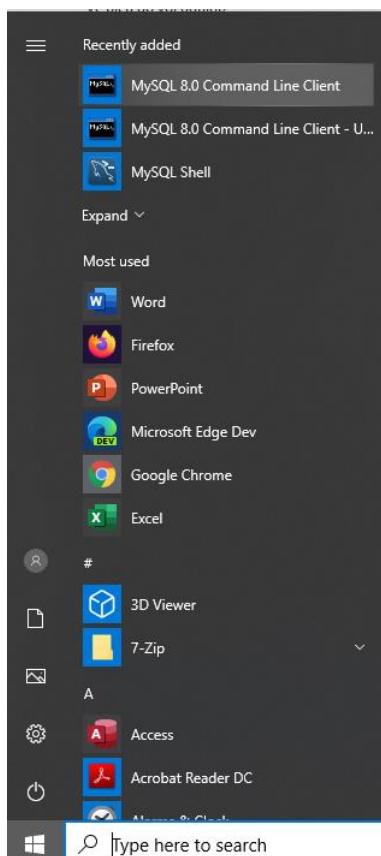
Bạn bấm nút x ở góc phải trên để thoát khỏi công cụ MySQL Workbench này.

Sau khi cài đặt xong thì trong nút Start của Windows, bạn sẽ thấy các menu của phần mềm MySQL như sau.

Chạm tới AI trong 10 ngày



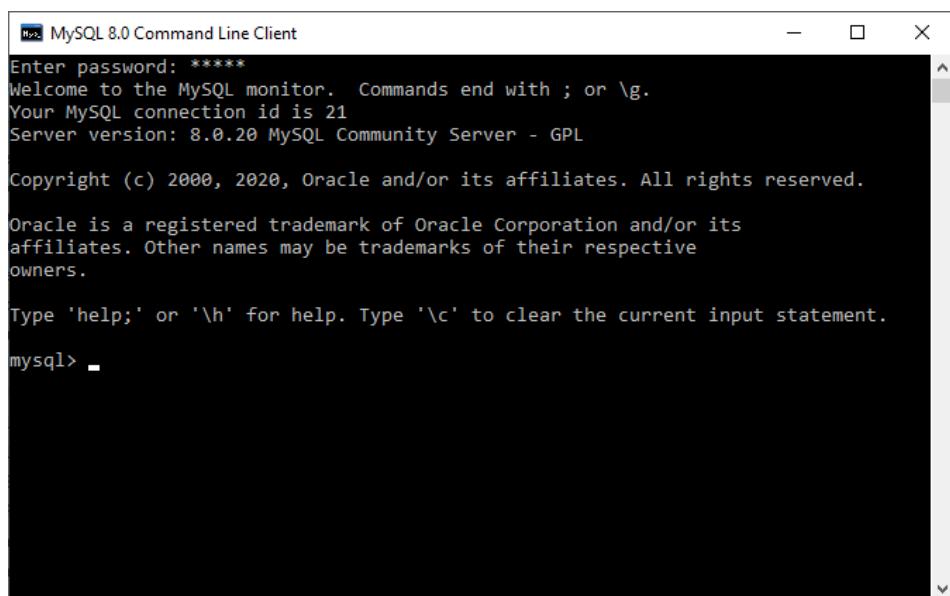
Hoặc trong nút Start có mục Recently added như bên dưới.



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. It displays the following text:
MySQL 8.0 Command Line Client
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

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.

Chạm tới AI trong 10 ngày

```
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, 1 warning (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, 1 warning (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;
```

Khởi tạo tài khoản cho cơ sở dữ liệu:

```
CREATE USER user_mysql@'localhost' IDENTIFIED WITH  
mysql_native_password BY 'mysqlpass';
```

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;
```

Lệnh R sau đây cài đặt thư viện, đọc dữ liệu từ Internet và nạp vào cơ sở dữ liệu MySQL.

```
# Bước 1: Cài đặt thư viện  
packages <- c('RMySQL', 'dbWriteTable')  
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {  
  install.packages(setdiff(packages, rownames(installed.packages())))  
}  
# Bước 2: Sử dụng thư viện để kết nối với MySQL  
library(RMySQL)  
# Bước 3: Chuẩn bị các thông số để kết nối tới MySQL  
server = 'localhost'  
dbname = 'mydata'  
username = 'user_mysql'  
password = 'mysqlpass'  
  
# Bước 4: Kết nối tới MySQL  
mysqlconn = dbConnect(MySQL(), user = username, password = password,  
  dbname = dbname, host = server)  
  
# Đọc dữ liệu từ Internet  
df = read.csv('https://thachln.github.io/datasets/bank/bank-  
additional-full.csv', sep=';')  
  
# Bước 5: Lưu data frame vào MySQL  
dbWriteTable(mysqlconn, value = df, name = "bank_additional_full",  
  append = TRUE )
```

Chạm tới AI trong 10 ngày

Trong cửa sổ lệnh của MySQL, bạn thực hiện lệnh sau để kiểm tra dữ liệu có được lưu trữ hay không?

Thực hiện lệnh “use mydata” để chuyển sang làm việc với cơ sở dữ liệu có tên là mydata:

```
mysql> use mydata;
```

Xem các bảng dữ liệu:

```
mysql> show tables;
```

| |
|----------------------|
| Tables_in_mydata |
| bank_additional_full |

1 row in set (0.01 sec)

Đếm số dòng dữ liệu:

```
select count(*) from bank_additional_full;
```

| |
|----------|
| count(*) |
| 41188 |

1 row in set (9.21 sec)

Xem cấu trúc bảng dữ liệu

```
mysql> describe bank_additional_full;
```

| Field | Type | Null | Key | Default | Extra |
|----------------|--------|------|-----|---------|-------|
| row_names | text | YES | | NULL | |
| age | bigint | YES | | NULL | |
| job | text | YES | | NULL | |
| marital | text | YES | | NULL | |
| education | text | YES | | NULL | |
| default | text | YES | | NULL | |
| housing | text | YES | | NULL | |
| loan | text | YES | | NULL | |
| contact | text | YES | | NULL | |
| month | text | YES | | NULL | |
| day_of_week | text | YES | | NULL | |
| duration | bigint | YES | | NULL | |
| campaign | bigint | YES | | NULL | |
| pdays | bigint | YES | | NULL | |
| previous | bigint | YES | | NULL | |
| poutcome | text | YES | | NULL | |
| emp.var.rate | double | YES | | NULL | |
| cons.price.idx | double | YES | | NULL | |
| cons.conf.idx | double | YES | | NULL | |
| euribor3m | double | YES | | NULL | |
| nr.employed | double | YES | | NULL | |
| y | text | YES | | NULL | |

22 rows in set (0.01 sec)

Như vậy bạn thấy là có thể lưu trữ nội dung một file CSV vào một bảng (table) của cơ sở dữ liệu. Column trong file CSV tương ứng Field (trường dữ liệu) trong CSDL.

Đọc dữ liệu từ MySQL

R cho phép bạn thực hiện câu truy vấn (query) và chuyển dữ liệu thành data frame.

```
sql = "SELECT * from bank_additional_full"
rs = dbSendQuery(mysqlconn, sql)
df = dbFetch(rs, n = -1)
names(df)
```

Đóng kết nối

```
dbHasCompleted(rs)
dbDisconnect(mysqlconn)
```

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 có được không? Câu trả lời là được. Phần mềm Hadoop sẽ giúp chúng ta làm được việc

Chạm tới AI trong 10 ngày

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 20.0. Để làm quen thôi thì bạn dùng phiên bản Desktop là được.

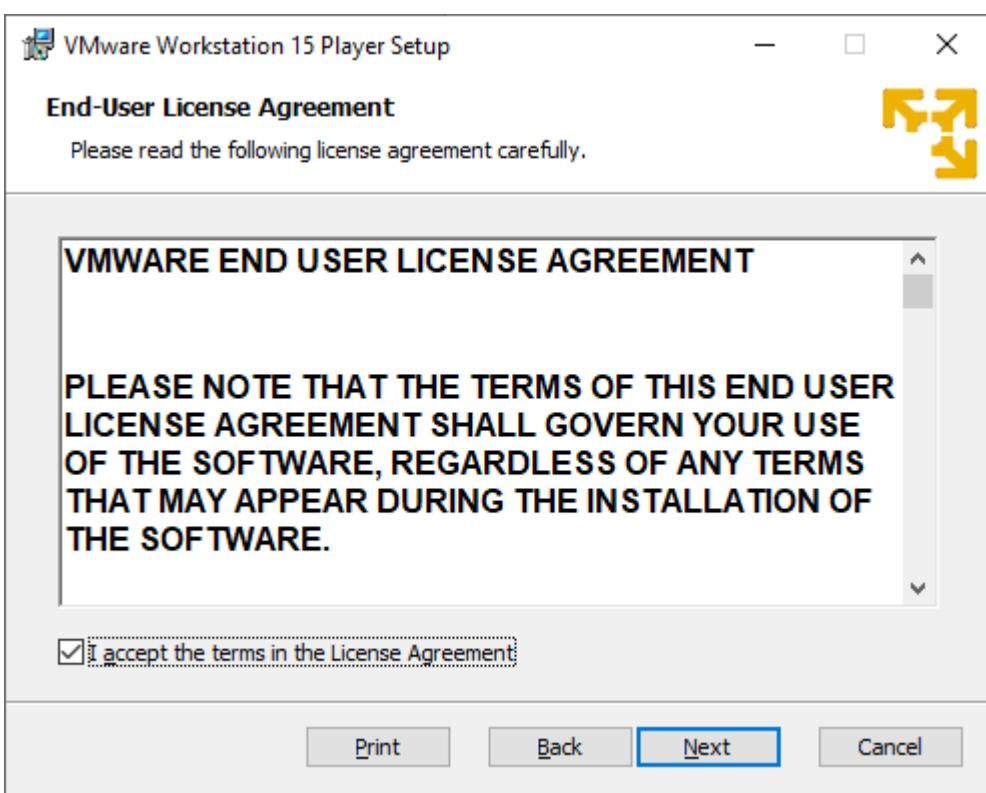
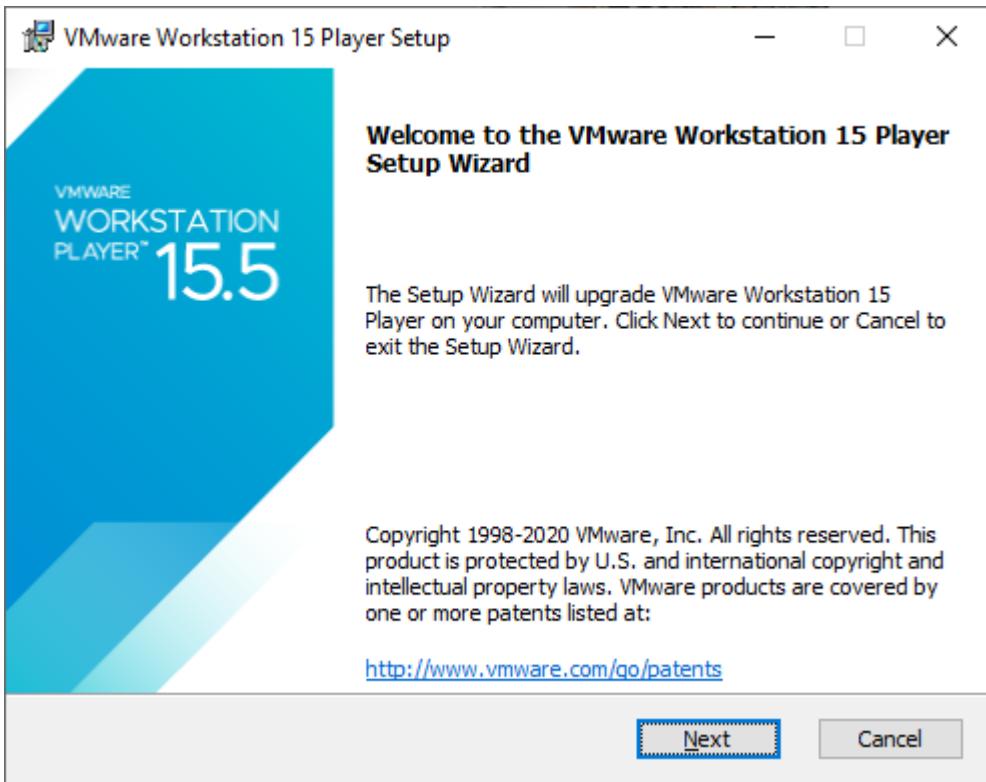
Sử dụng VMware

Vào trang web <https://www.vmware.com/go/downloadworkstationplayer> để tải và cài đặt VMware Workstation Player (gọi tắt là VMware).

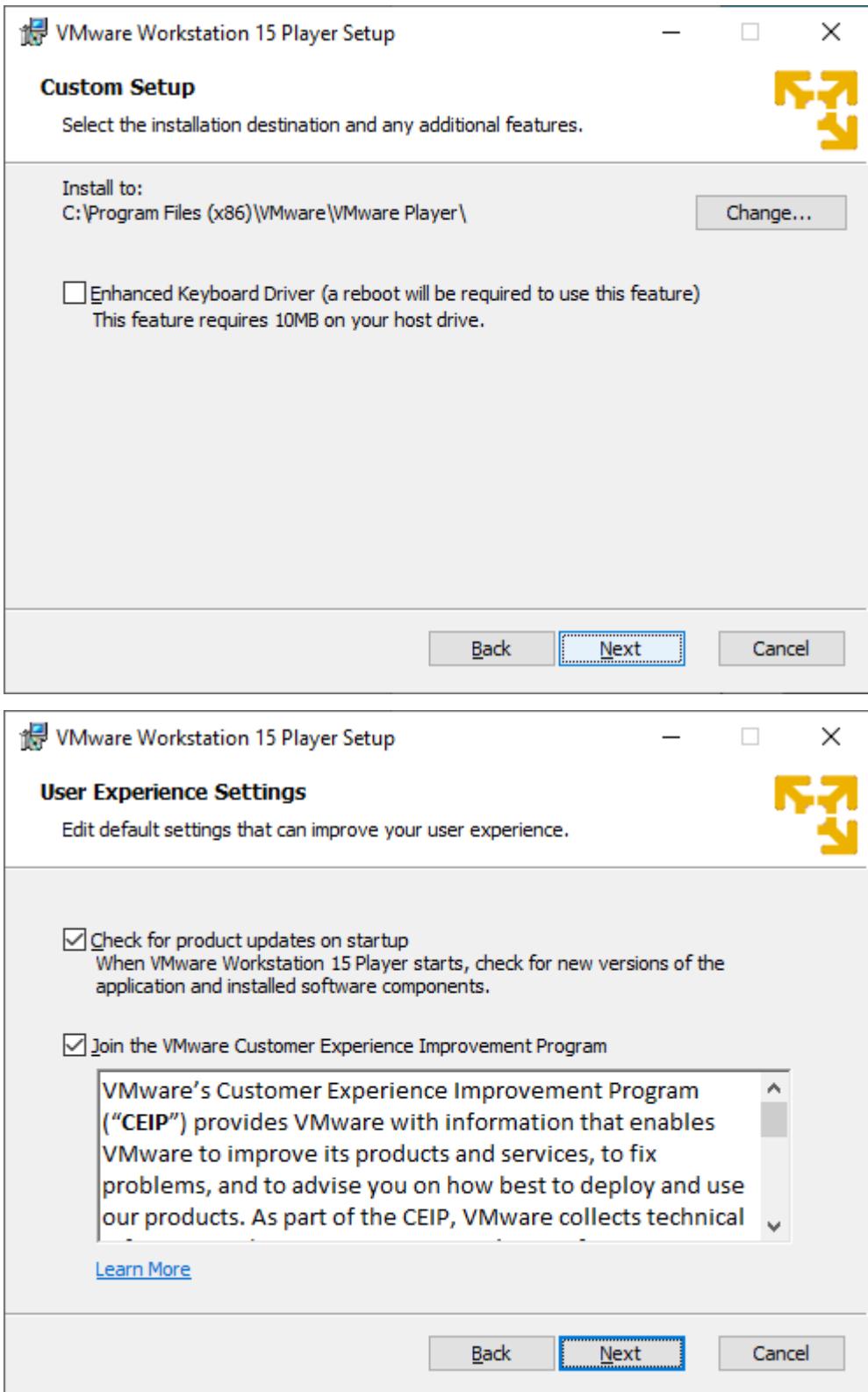
The screenshot shows the VMware website at https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_w... The top navigation bar includes links for US, Login, Training, Community, and Store. Below the navigation is a breadcrumb trail: Home > All Downloads > VMware Workstation Player. A dropdown menu shows Major Version: 15.0 (latest) and Minor Version: 15.5.2 (latest). There are two download options: "Product Downloads" and "Open Source". The "Product Downloads" section lists "VMware Workstation 15.5.2 Player for Windows 64-bit Operating Systems" (exe | 138.46 MB) with a "Download" button. The "Open Source" section lists "VMware Workstation 15.5.2 Player for Linux 64-bit" (bundle | 157.80 MB) with a "Download" button.

Quá trình cài đặt tương đối dễ dàng, bạn cứ làm theo hướng dẫn, cơ bản là chọn Yes và Next:

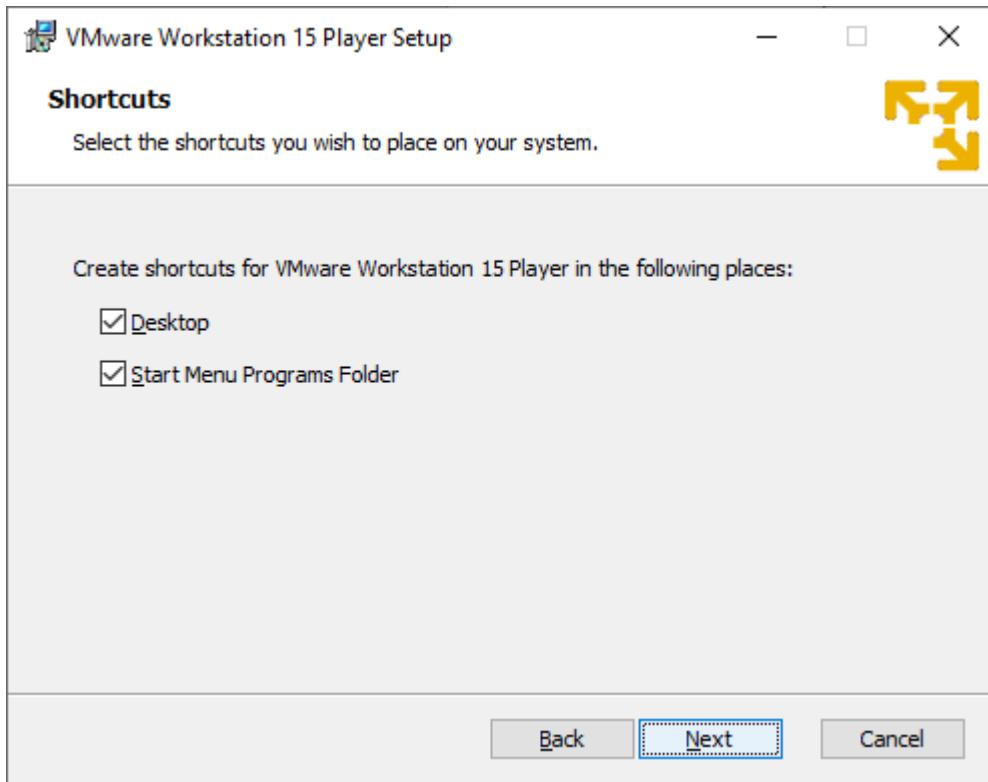
Chạm tới AI trong 10 ngày



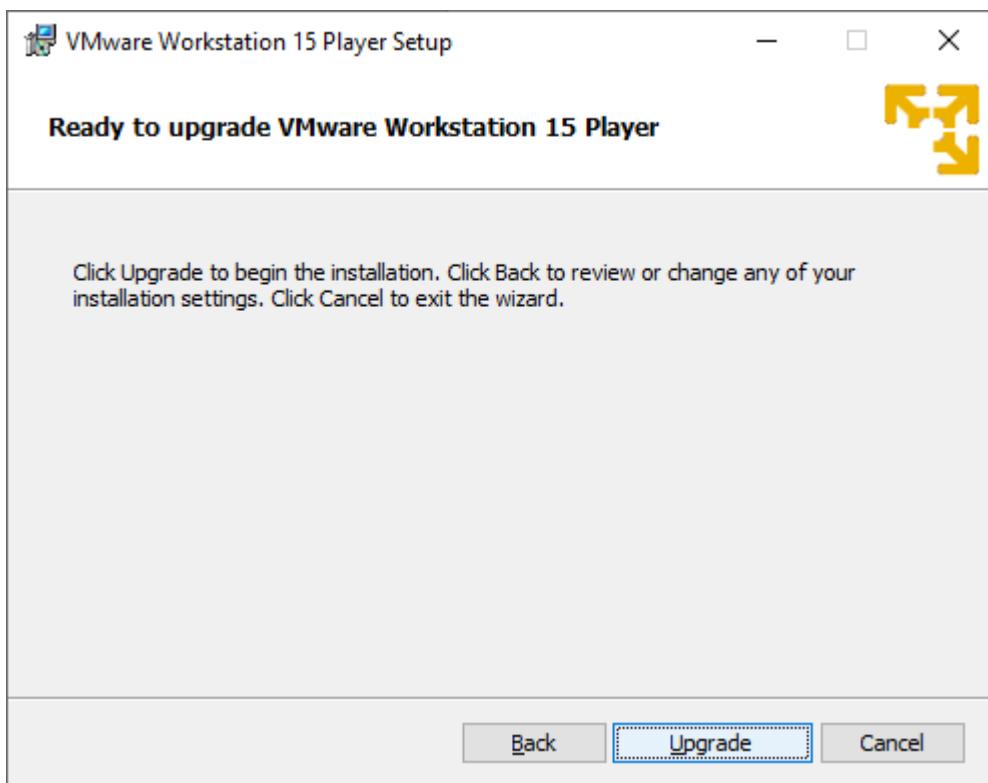
Chạm tới AI trong 10 ngày



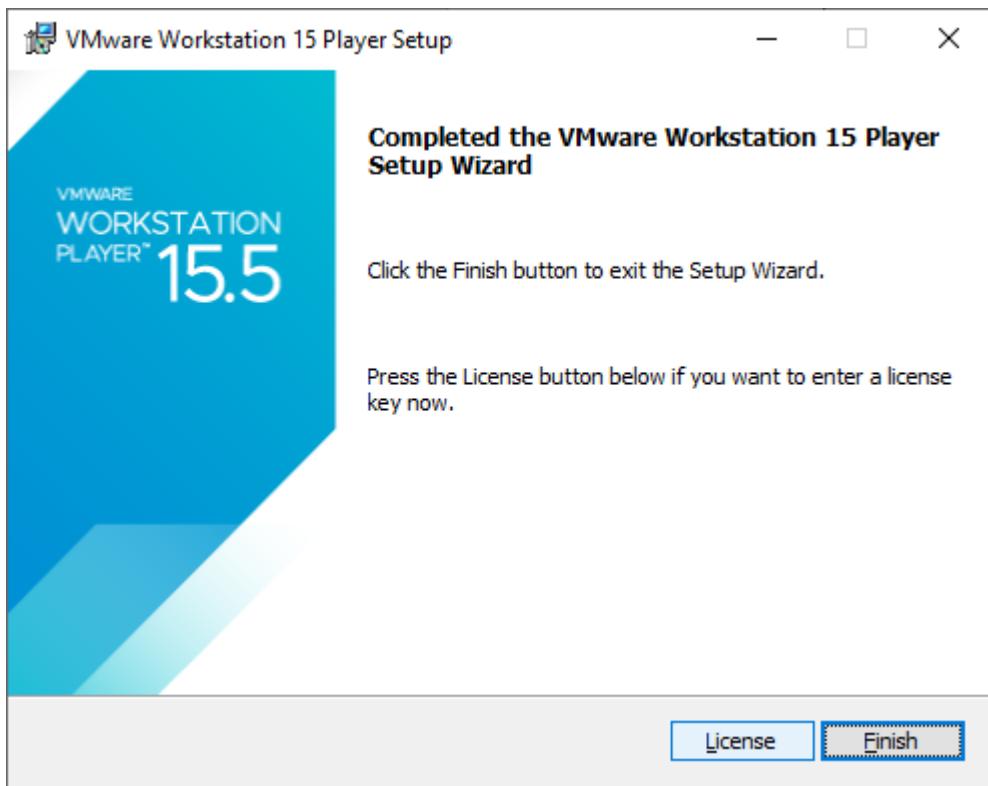
Chạm tới AI trong 10 ngày



Nếu máy tính của bạn đã có VMware thì có thể nâng cấp bằng cách bấm nút Upgrade.

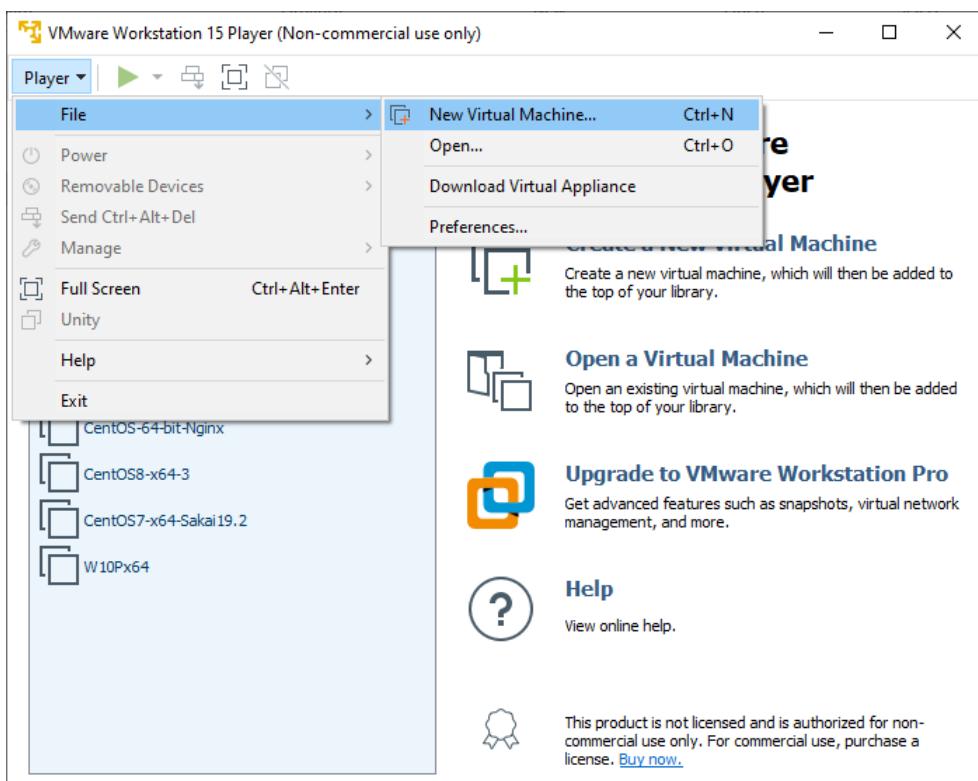


Chạm tới AI trong 10 ngày

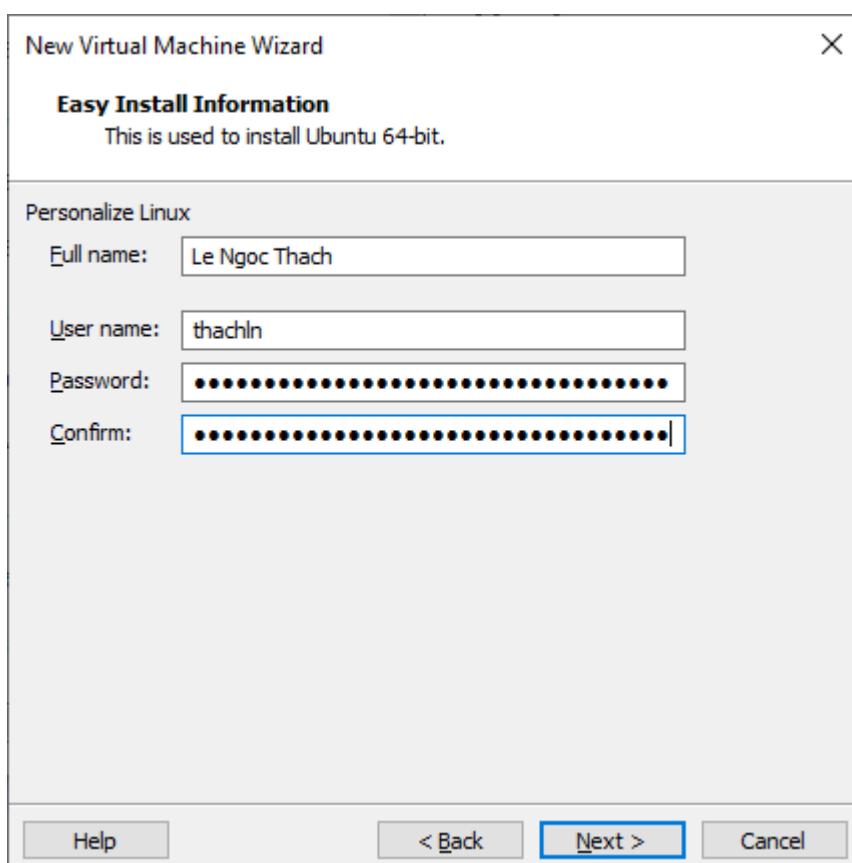
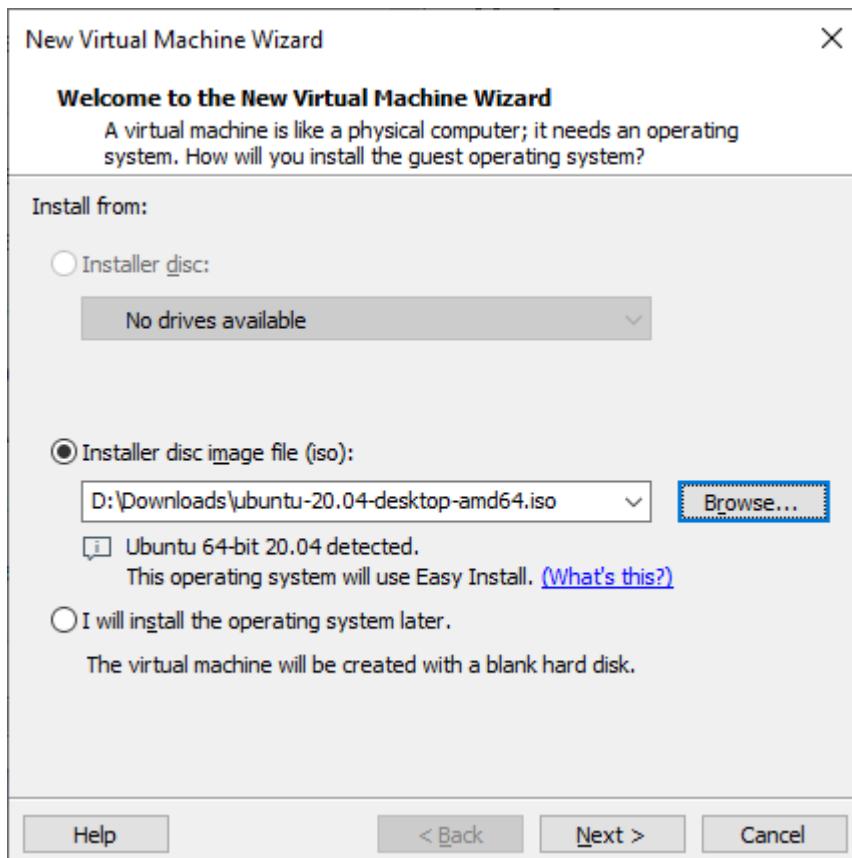


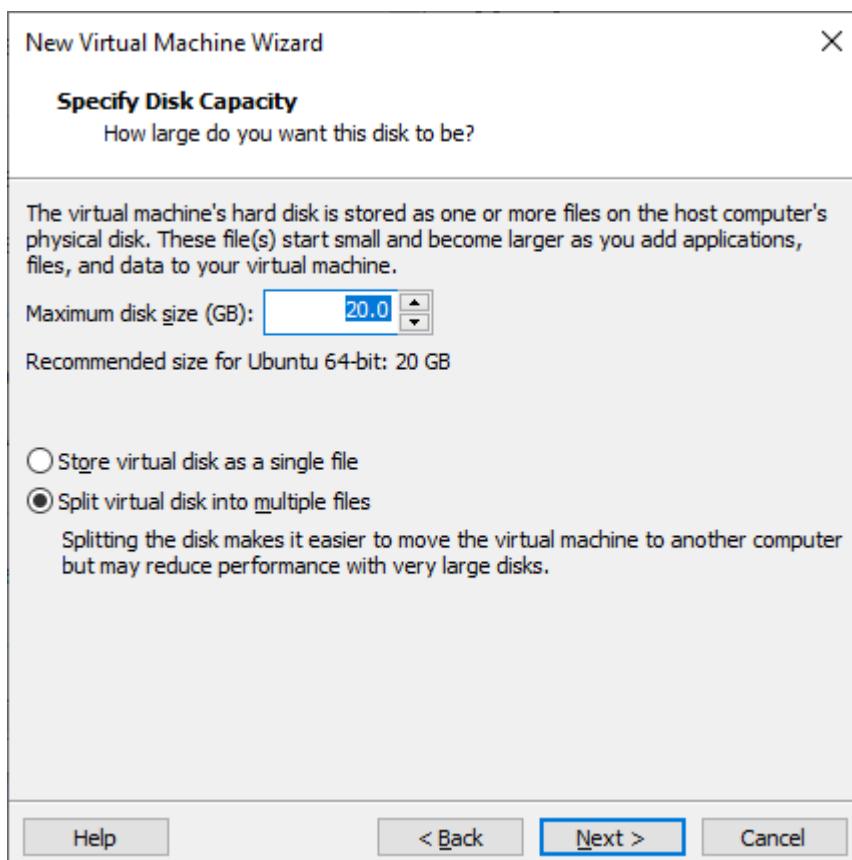
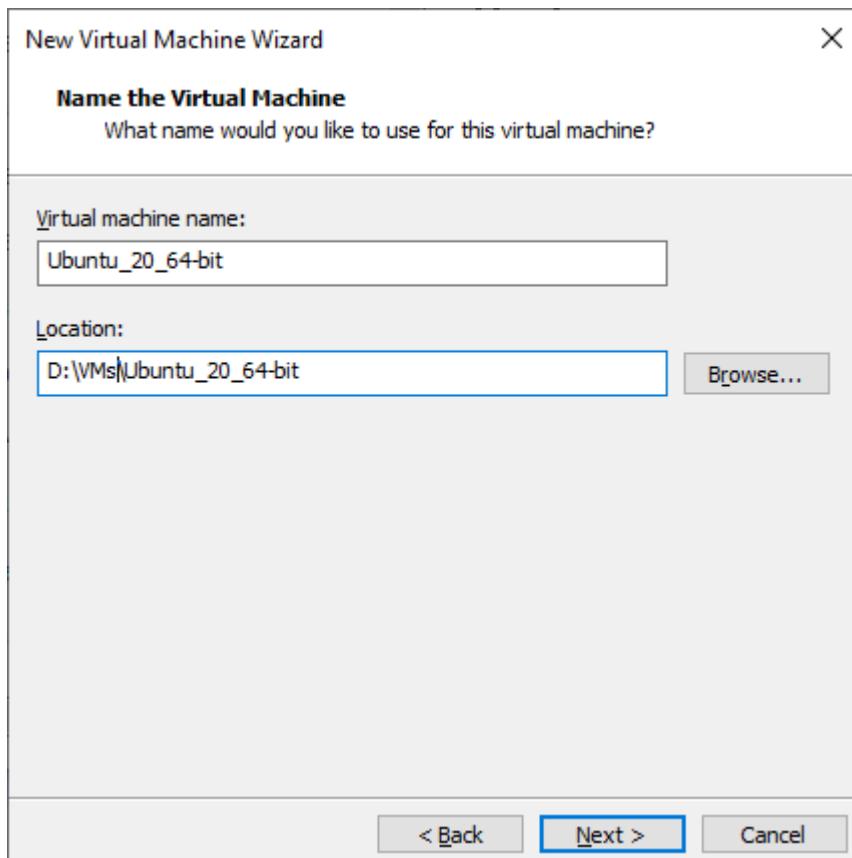
Tạo máy ảo Ubuntu

Sau đó khởi động VMware và khởi tạo máy ảo theo hướng dẫn sau đây

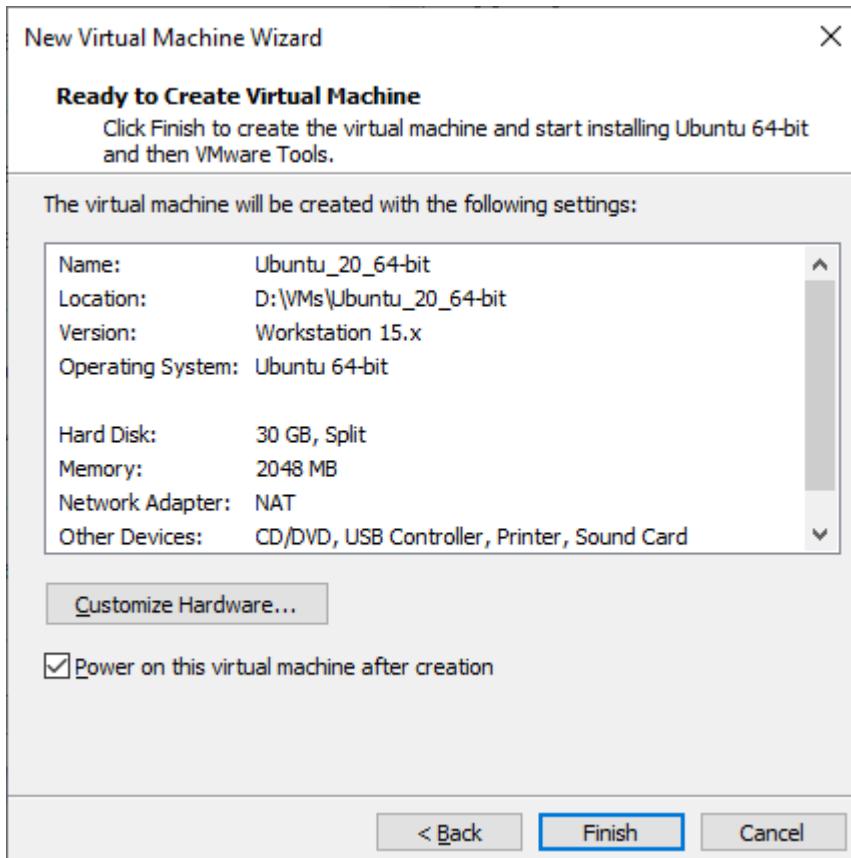


Chạm tới AI trong 10 ngày



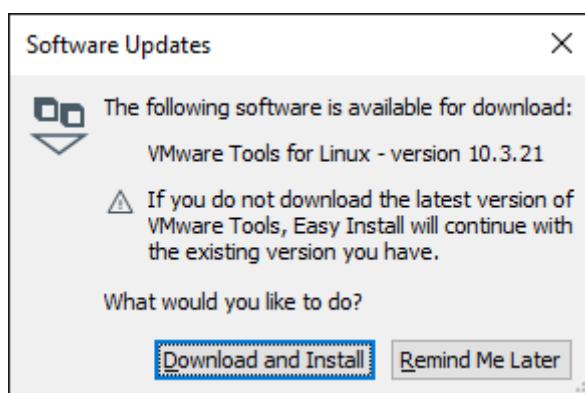


Chạm tới AI trong 10 ngày



Bấm nút Finish để hoàn thành việc tạo máy tính ảo bên trong phần mềm VMWare.

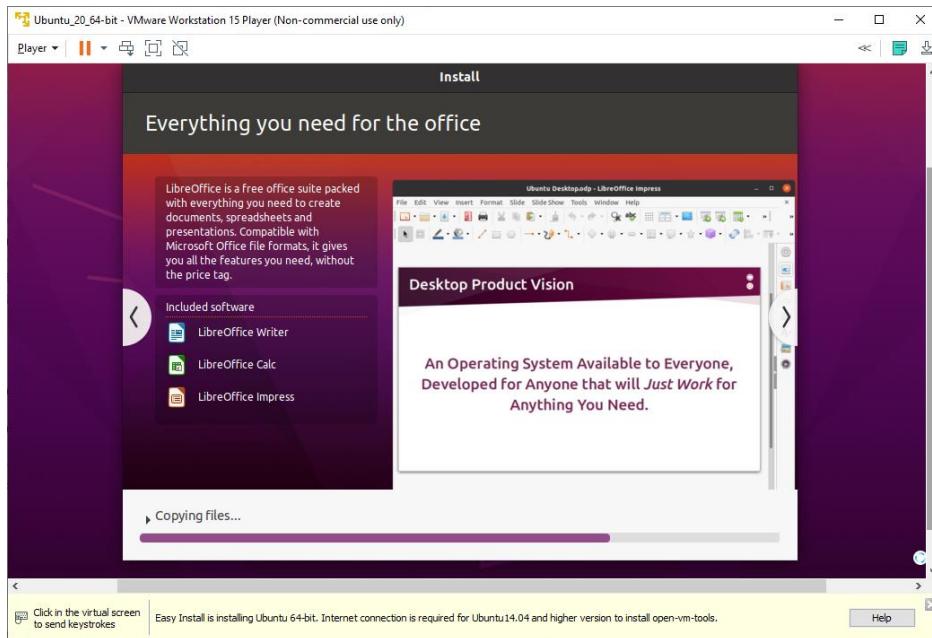
VMware sẽ tự động bắt đầu quá trình cài đặt Ubuntu cho bạn. Trong lúc cài đặt thì có thể hiển thị hộp thoại bên dưới. Bạn cứ chọn Download and Install.



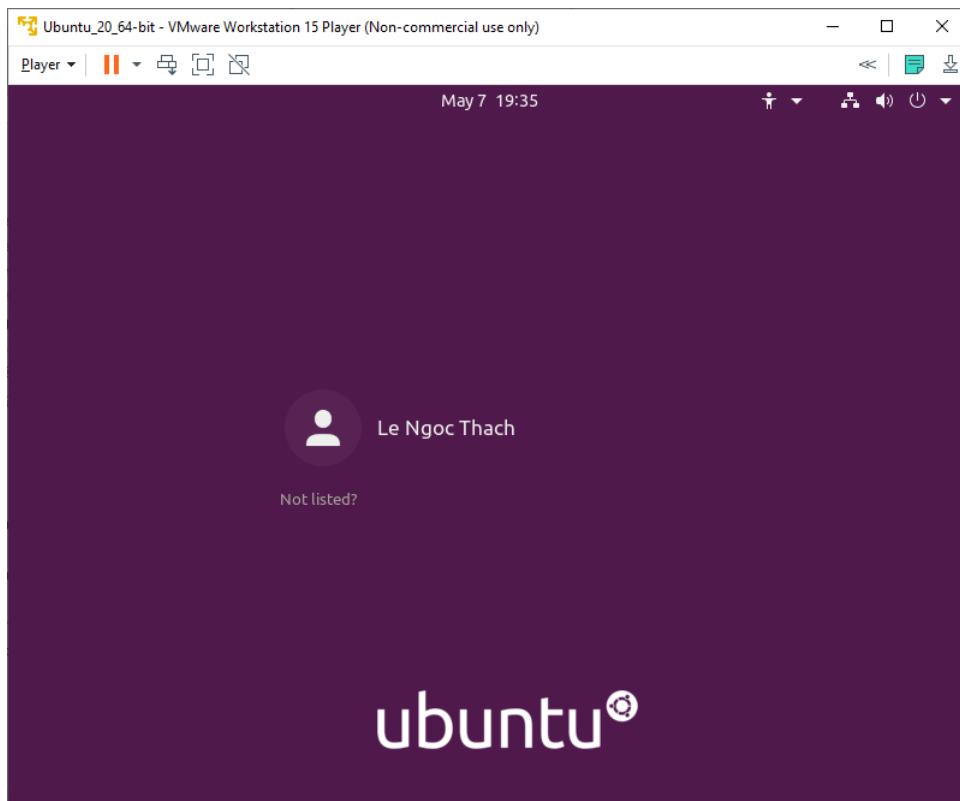
Như vậy lúc này trên máy tính Windows 10 của bạn có phần mềm VMware. Bên trong phần mềm VMware lại có một cái máy tính Ubuntu 20.0. Thật là tuyệt vời phải không?

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.

Chạm tới AI trong 10 ngày

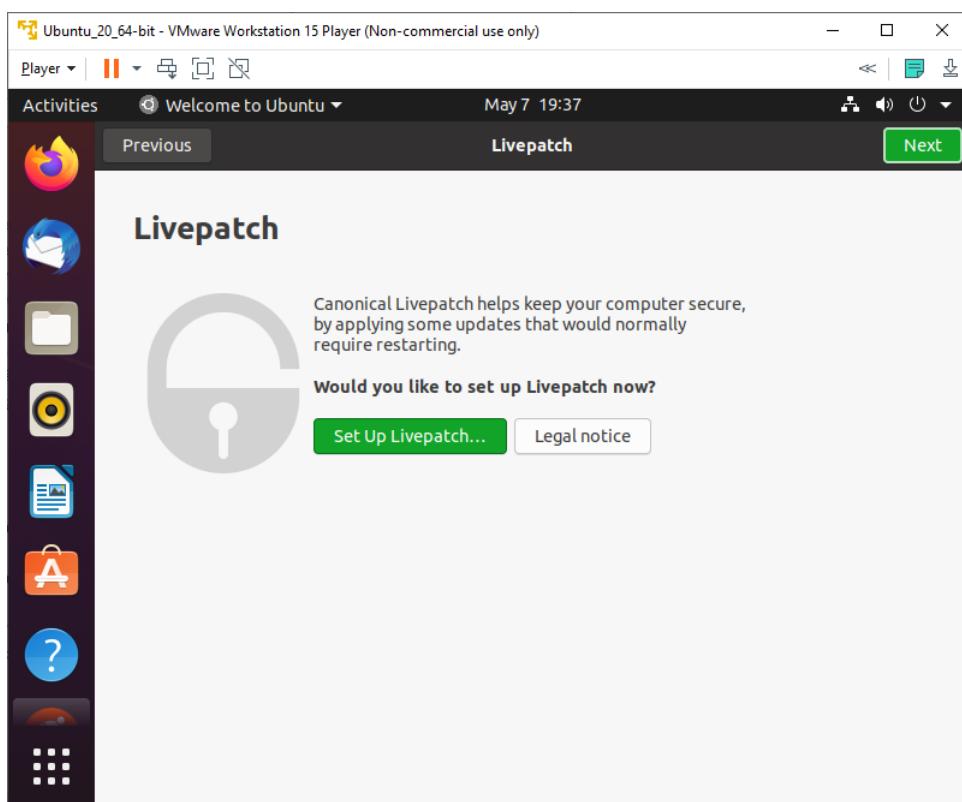
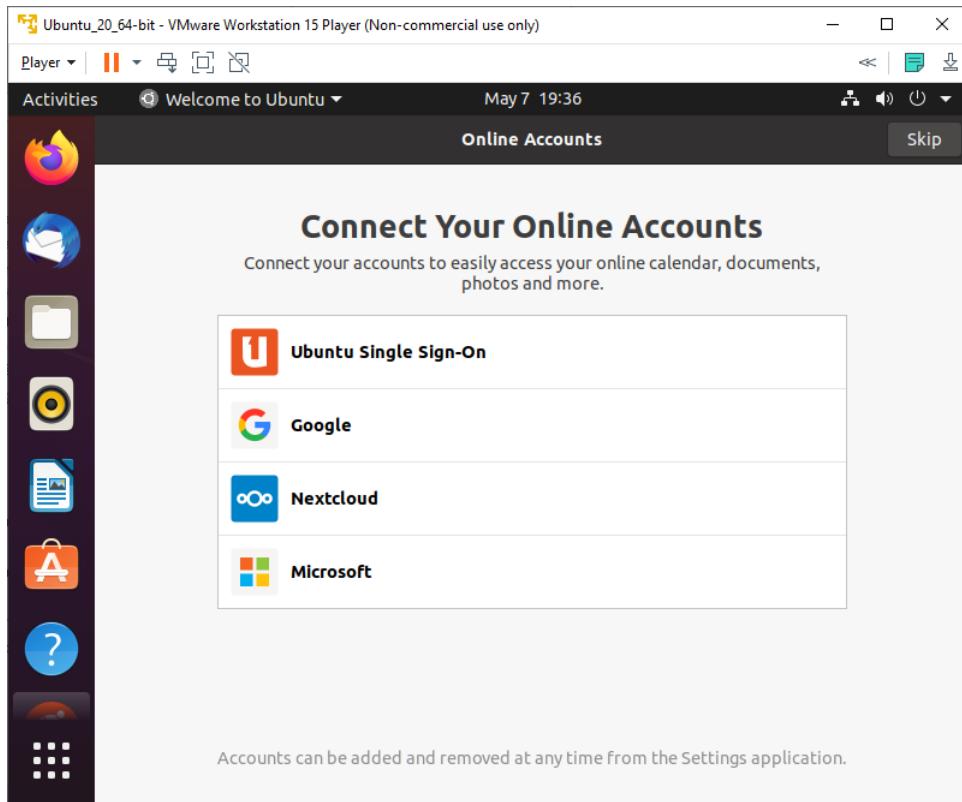


Sau khi cài đặt xong thì bấm vào tên mà bạn đã khai báo lúc nãy. Sau đó gõ password để đăng nhập

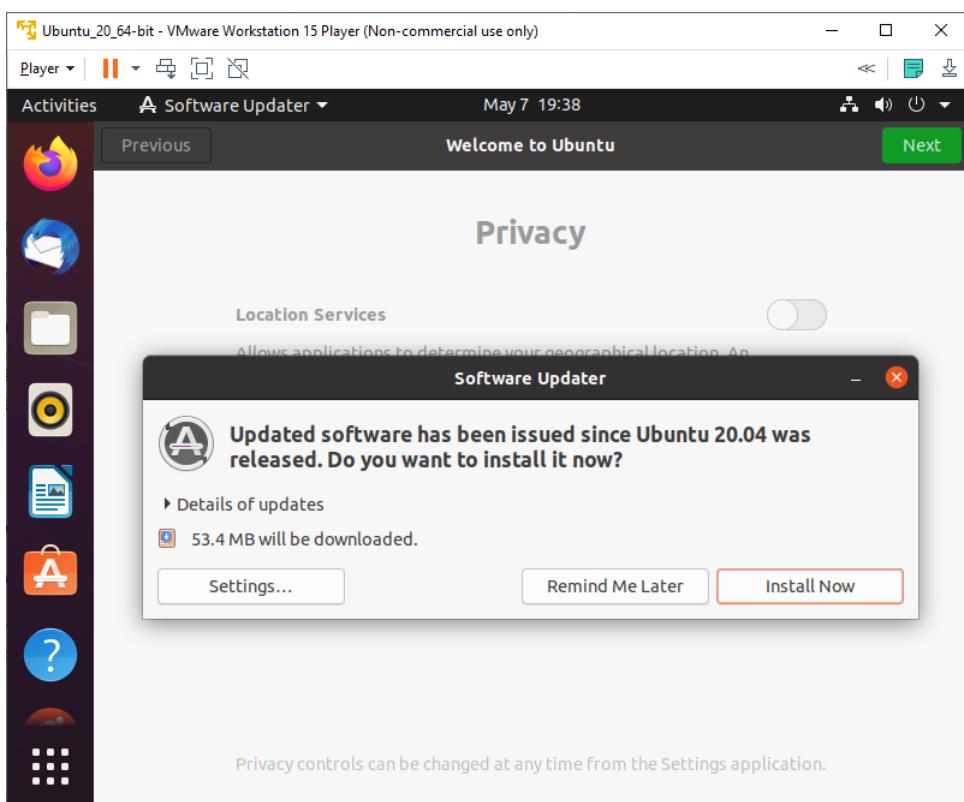
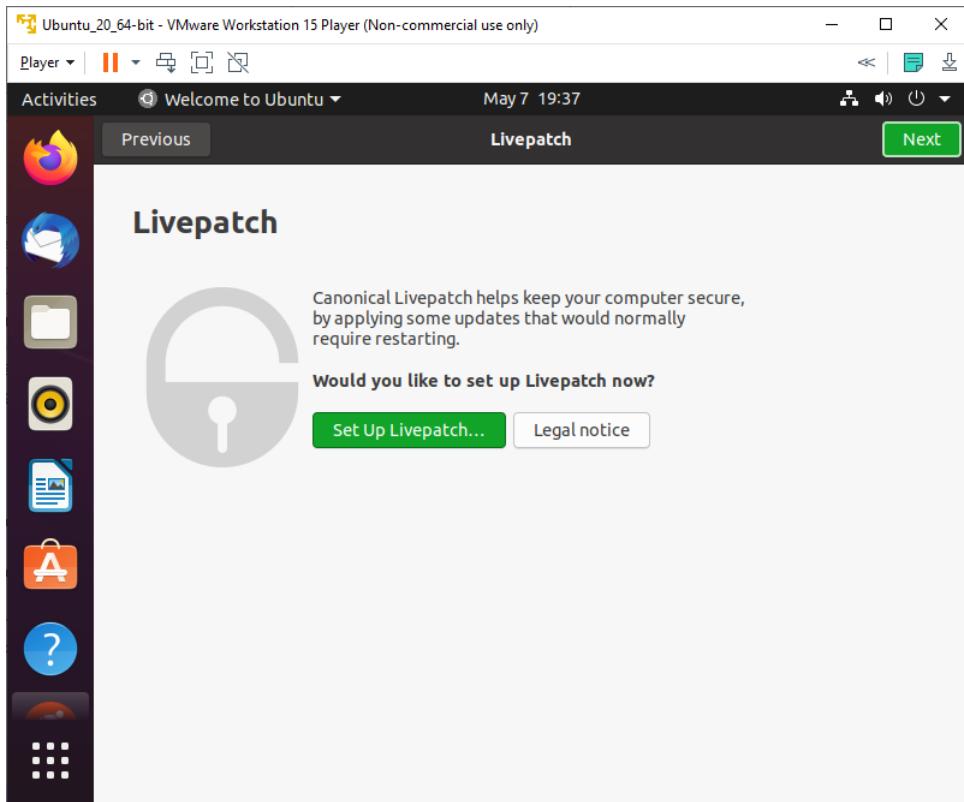


Sau khi đăng nhập, bạn theo hướng dẫn bấm nút ở góc phải trên Skip, Next, và Done.

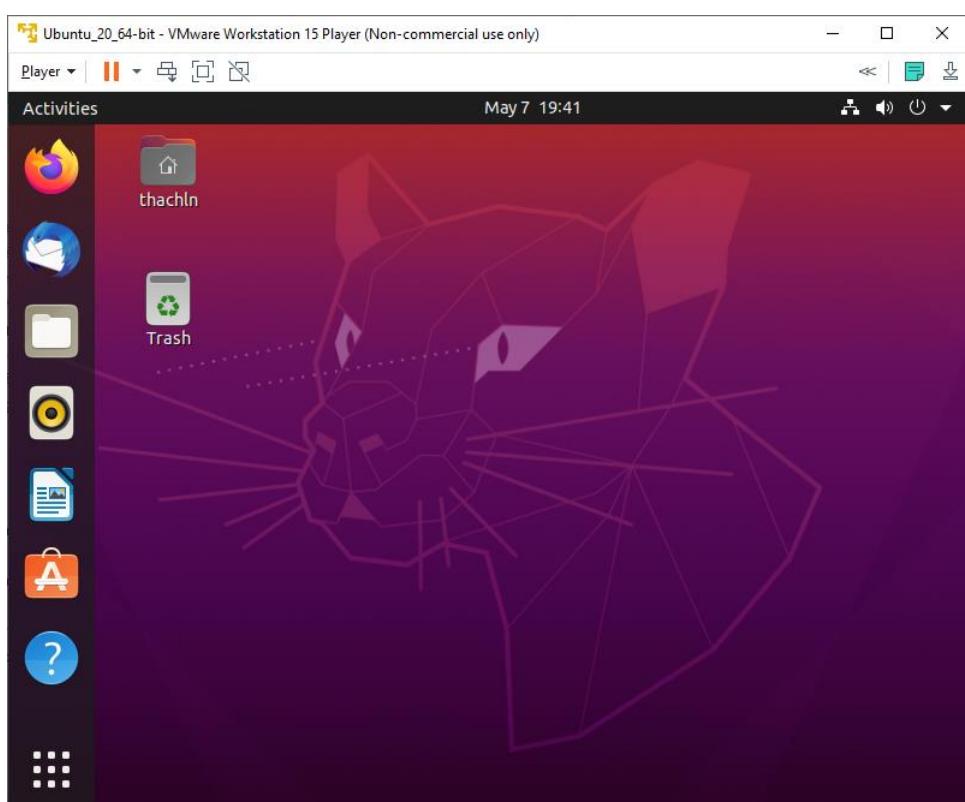
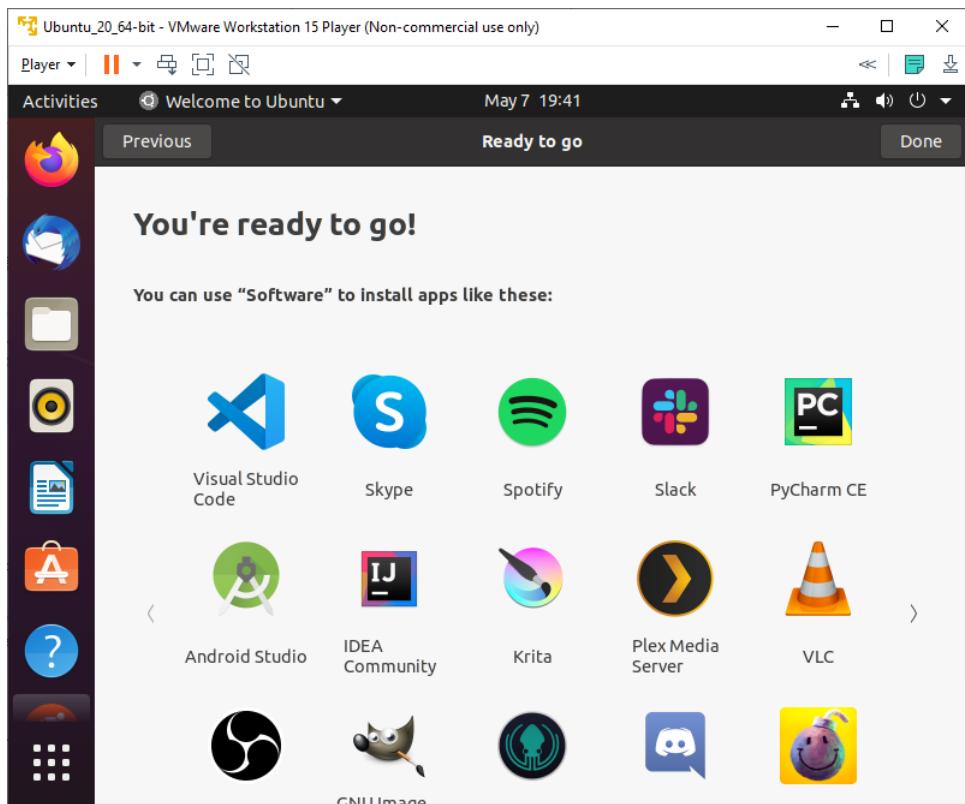
Chạm tới AI trong 10 ngày



Chạm tới AI trong 10 ngày



Chạm tới AI trong 10 ngày



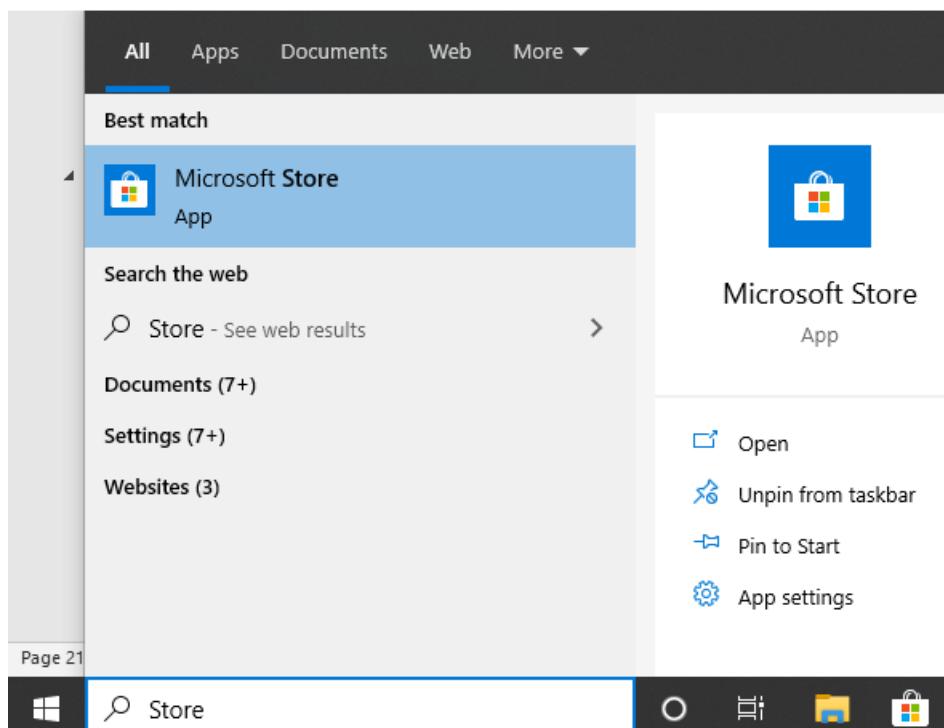
Như vậy bạn đã có máy ảo Ubuntu 20.04,

Để phóng to máy ảo đầy màn hình để làm việc thì bấm vào biểu tượng trong thanh công cụ của VMware.

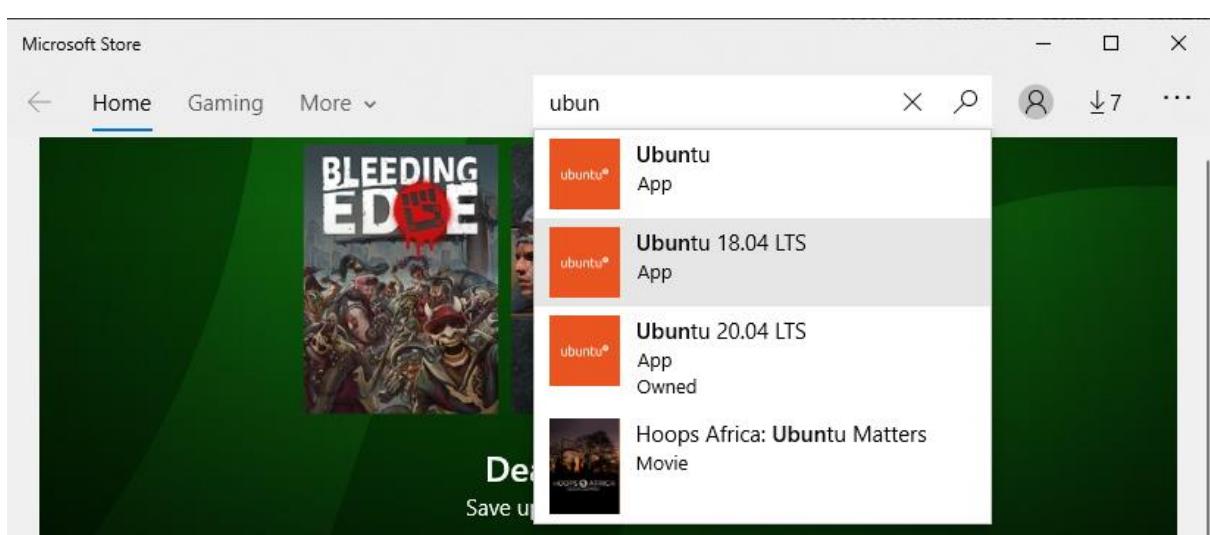
Cài đặt Ubuntu nhúng trong Windows

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:



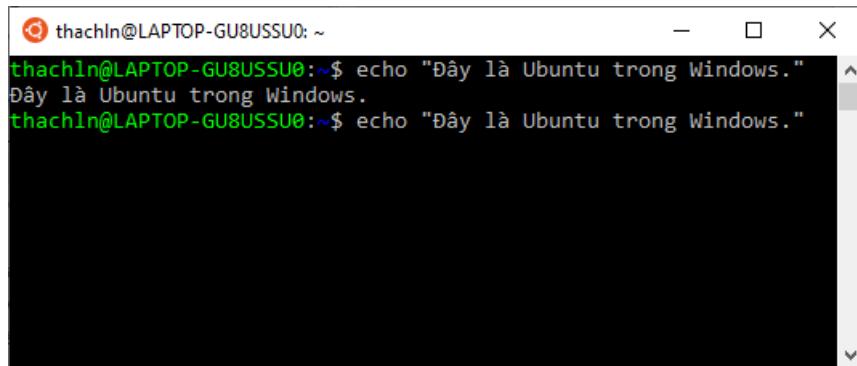
Bấm vào biểu tượng hoặc mục Microsoft Store. Sau đó gõ Ubuntu trên ô Search. Tôi cài Ubuntu 18.04 LTS để thực hiện một số phần ví dụ trong eBook này.



Chạm tới AI trong 10 ngày

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!

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
```

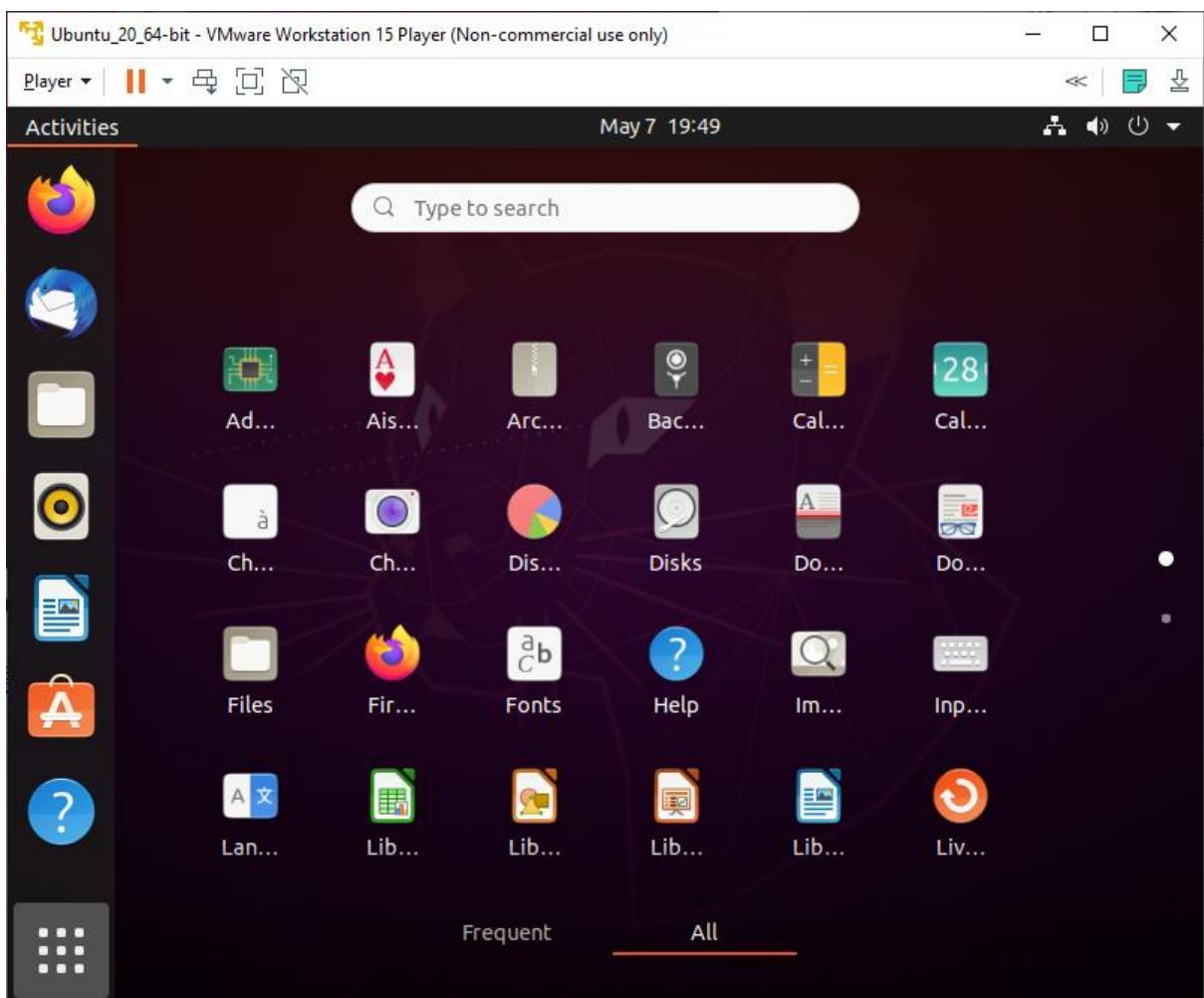
Hy vọng bạn có trải nghiệm sử dụng Ubuntu ngay trên Windows 10 để hỗ trợ học hành, cho công việc tốt!

Cám ơn Microsoft! Cám ơn Ubuntu!

Bài 18: 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.

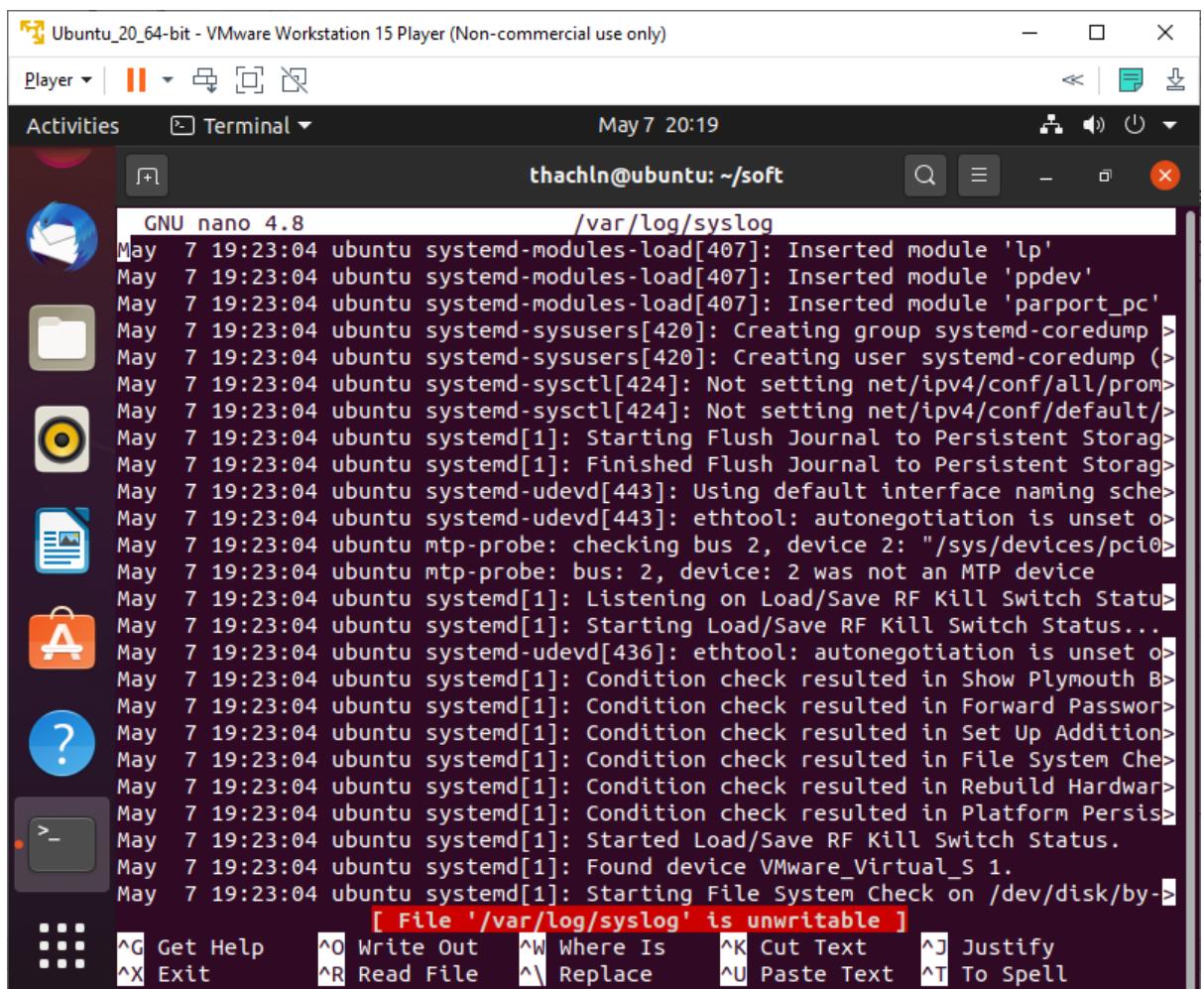
Chạm tới AI trong 10 ngày

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 menu: Get Help, Write Out, Where Is, Cut Text, Justify, Exit, Read File, Replace, Paste Text, 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

Chạm tới AI trong 10 ngày

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
```

Chạm tới AI trong 10 ngày

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 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 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 xem sao:

Chạm tới AI trong 10 ngày

```
● 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 nhấp 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
```

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)
```

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:

Chạm tới AI trong 10 ngày

```
sudo ln -nsf /opt/jdk /usr/lib/jvm/default-java
```

Vì đường dẫn Java khá dài nên hơi 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.

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:

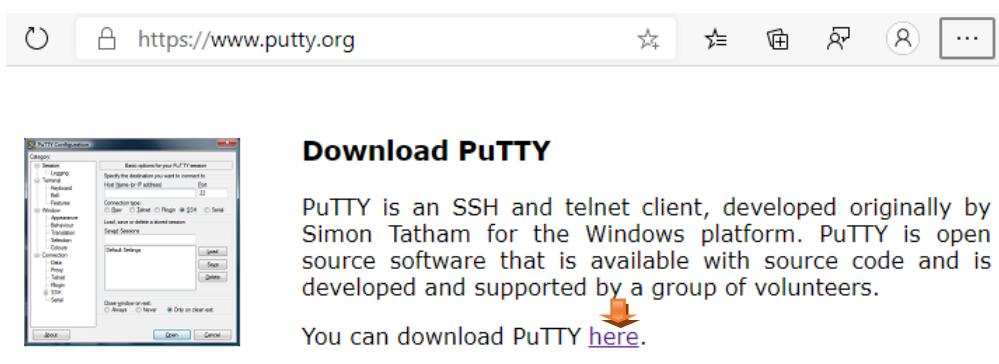
```
sudo service ssh restart
```

Cài tường lửa cho Ubuntu

```
apt install firewalld
```

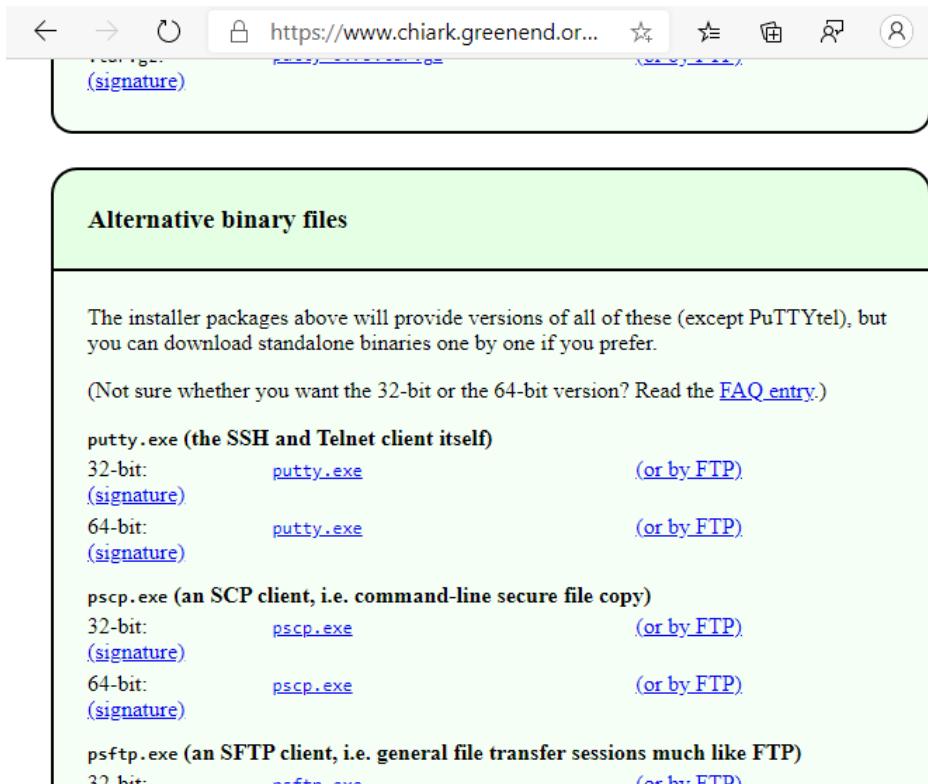
Sử dụng công cụ truy cập từ xa

Để làm việc từ xa thì trên máy chủ chạy dịch vụ SSH Server, trên 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/>



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.

Chạm tới AI trong 10 ngày



The screenshot shows a web browser window with the URL <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. The page title is "Alternative binary files". It contains text about the installer packages providing versions of all the binaries except PuTTYtel, and links to download standalone binaries for 32-bit and 64-bit architectures for Putty, PSCP, and PSFTP.

Alternative binary files

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)

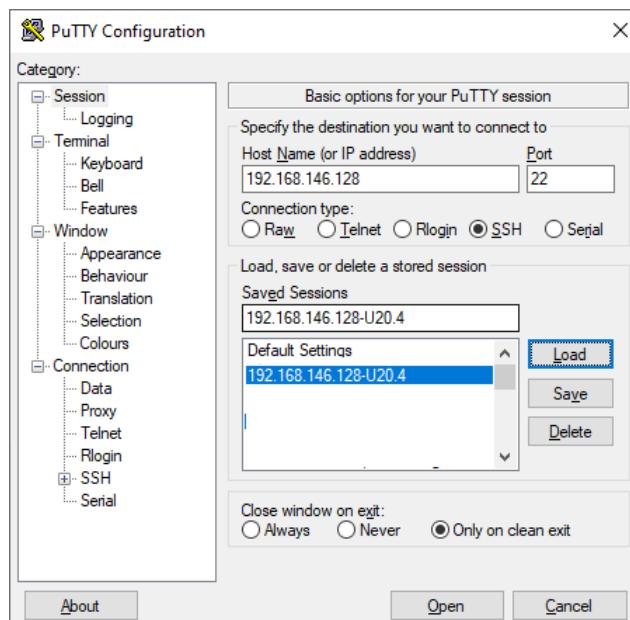
32-bit: [pscp.exe](#) ([\(or by FTP\)](#))
[\(signature\)](#)

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](#) ([\(or by FTP\)](#))

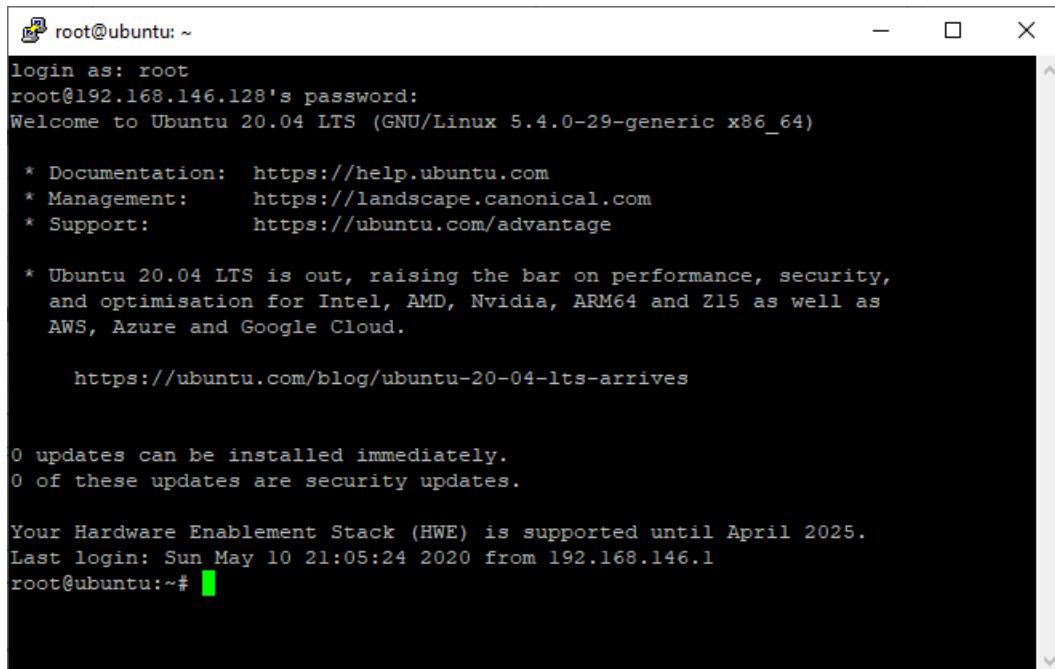
Khởi động putty.exe và điền thông số: địa chỉ IP và port (mặc định SSH sẽ là port 22)



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.

Chạm tới AI trong 10 ngày



root@ubuntu: ~
login as: root
root@192.168.146.128's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-29-generic x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

* Ubuntu 20.04 LTS is out, raising the bar on performance, security,
and optimisation for Intel, AMD, Nvidia, ARM64 and Z15 as well as
AWS, Azure and Google Cloud.

https://ubuntu.com/blog/ubuntu-20-04-lts-arrives

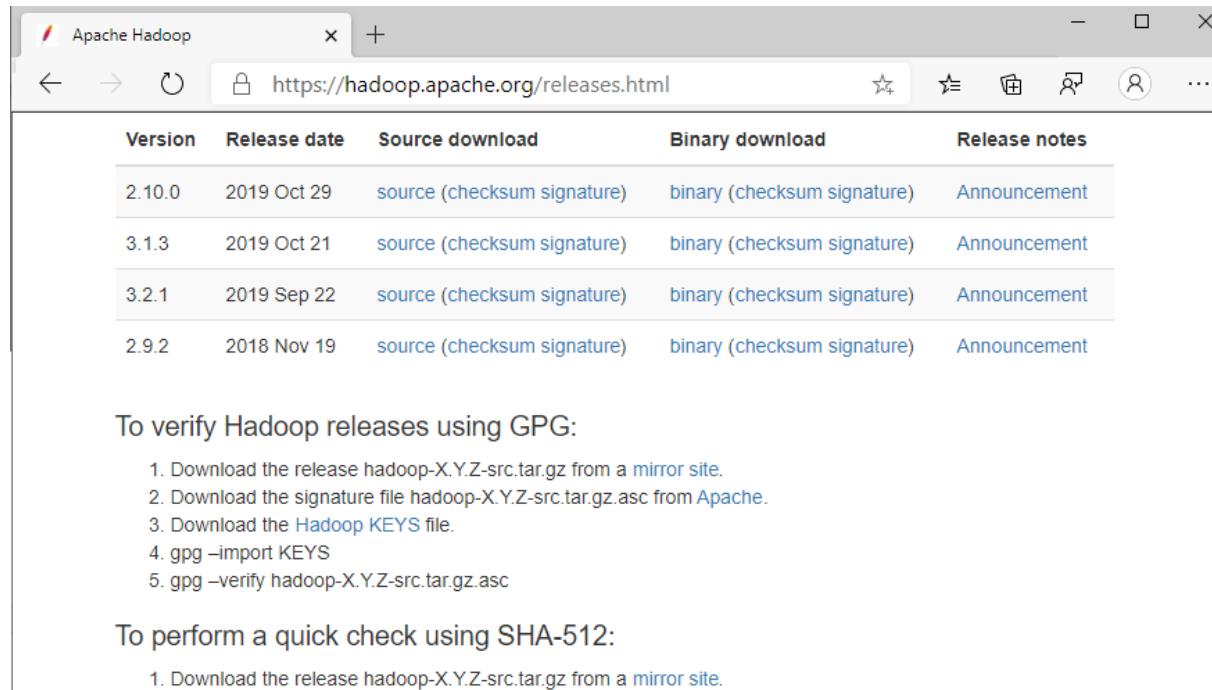
0 updates can be installed immediately.
0 of these updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun May 10 21:05:24 2020 from 192.168.146.1
root@ubuntu:~#

Bài 19: Cài đặt Hadoop

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 2.10.0 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 2.10.0 về thư mục hiện hành bằng lệnh wget <url>:

```
wget http://mirrors.viethosting.com/apache/hadoop/common/hadoop-2.10.0/hadoop-2.10.0.tar.gz
```

Giải nén

Giải nén ra thư mục /opt

```
sudo tar -xvzf ./hadoop-2.10.0.tar.gz -C /opt
```

Chạm tới AI trong 10 ngày

Tạo ảnh xạ thư mục /opt/hadoop-2.10.0 vào /opt/hadoop. Bước này giống như tạo shortcut trên Windows, thay vì truy cập vào đường dẫn dài /opt/hadoop-2.10.0 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-2.10.0 /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 các 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:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
```

Chạm tới AI trong 10 ngày

```
</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_C
```

Chạm tới AI trong 10 ngày

```
ONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_M  
APRED_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
```

Chuẩn bị dữ liệu cho Hadoop

```
/opt/hadoop/bin/hdfs namenode -format
```

Khởi động hadoop

```
/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”:

```
nano /opt/hadoop/logs/hadoop-root-datanode-ubuntu.log
```

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

Chạm tới AI trong 10 ngày

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 50075 như sau:

```
INFO org.mortbay.log: jetty-6.1.26
INFO org.mortbay.log: Started HttpServer2$SelectChannelConnectorWithSafeStartup@localhost:36541
INFO org.apache.hadoop.hdfs.server.datanode.web.DatanodeHttpServer: Listening HTTP traffic on /0.0.0.0:50075
INFO org.apache.hadoop.util.JvmPauseMonitor: Starting JVM pause monitor
INFO org.apache.hadoop.server.datanode.DataNode: dnUserName = root
```

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:

```
firewall-cmd --permanent --add-port=50070/tcp
firewall-cmd --permanent --add-port=9000/tcp
firewall-cmd --permanent --add-port=50075/tcp
firewall-cmd --permanent --add-port=8088/tcp
firewall-cmd --reload
```

Truy cập Hadoop từ trình duyệt

Từ máy tính chạy Windows, bạn mở trình duyệt truy cập vào địa chỉ <http://192.168.146.128:50070/>

Chạm tới AI trong 10 ngày

The screenshot shows a web browser window titled "Namenode information". The address bar indicates the URL is "Not secure | 192.168.146.128:50070/dfshealth.html#tab-...". The main content area is titled "Hadoop" and has a green header bar with tabs: "Overview" (which is selected), "Datanodes", "Datanode Volume Failures", "Snapshot", "Startup Progress", and "Utilities". Below the header, the page title is "Overview '192.168.146.128:9000' (active)". A table provides cluster details:

| | |
|-----------------------|--|
| Started: | Fri May 08 13:55:09 +0700 2020 |
| Version: | 2.10.0, re2f1f118e465e787d8567dfa6e2f3b72a0eb9194 |
| Compiled: | Wed Oct 23 02:10:00 +0700 2019 by jhung from branch-2.10.0 |
| Cluster ID: | CID-4f81ded2-501b-4576-97a8-0b78ae81d0c9 |
| Block Pool ID: | BP-1233815817-127.0.1.1-1588920843901 |

Below the table, under the "Summary" section, it says "Security is off." and "Safemode is off.".

Truy cập <http://192.168.146.128:50075/>:

Chạm tới AI trong 10 ngày

The screenshot shows a web browser window titled "DataNode Information". The address bar indicates the URL is "Not secure | 192.168.146.128:50075/datanode.html". The main content area displays "DataNode on ubuntu:50010". Below this, there are two tables. The first table shows cluster details:

| | |
|-------------|--|
| Cluster ID: | CID-4f81ded2-501b-4576-97a8-0b78ae81d0c9 |
| Version: | 2.10.0 |

The second table lists Block Pools:

| Namenode Address | Block Pool ID | Actor State | Last Heartbeat | Last Block Report | Last Block Report Size (Max Size) |
|----------------------|---------------------------------------|-------------|----------------|-------------------|-----------------------------------|
| 192.168.146.128:9000 | BP-1233815817-127.0.1.1-1588920843901 | RUNNING | 2s | 40 minutes | 0 B (64 MB) |

Below the tables, the section "Volume Information" is visible, featuring a table with columns: Directory, StorageType, Capacity Used, Capacity Left, Capacity Reserved, Reserved Space for Replicas, and Blocks.

Truy cập <http://192.168.146.128:8088/>

Chạm tới AI trong 10 ngày

The screenshot shows the Hadoop Web UI interface. On the left, there's a sidebar with a navigation tree under 'Cluster' containing links like 'About', 'Nodes', 'Node Labels', 'Applications' (with sub-options: NEW, NEW_SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED), 'Scheduler', and 'Tools'. The main content area has three sections: 'Cluster Metrics' (with tables for Apps Submitted, Apps Pending, Apps Running, Apps Completed, and Containers Running, all showing 0), 'Cluster Nodes Metrics' (with tables for Active Nodes, Decommissioning Nodes, and Decommissioned Nodes, all showing 0), and 'Scheduler Metrics' (with a table for Scheduler Type showing Capacity Scheduler). Below these is a table titled 'Show 20 entries' with columns for ID, User, Name, Application Type, Queue, Application Priority, Start Time, Launch Time, Finish Time, and State. A message at the bottom says 'Showing 0 to 0 of 0 entries'.

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 đó!

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
```

Bài 20: Trải nghiệm Hadoop với R và 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:

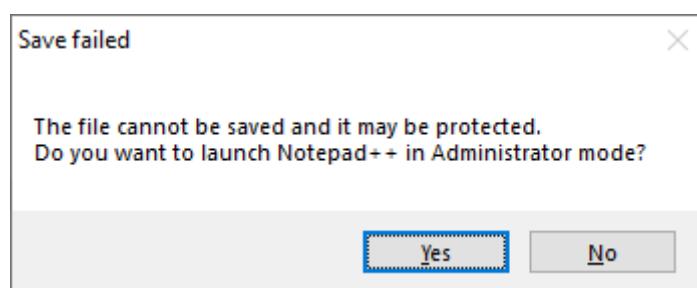
Dùng Notepad++ mở file:

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/

Chạm tới AI trong 10 ngày

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.

```
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
```

Chạm tới AI trong 10 ngày

```
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”!

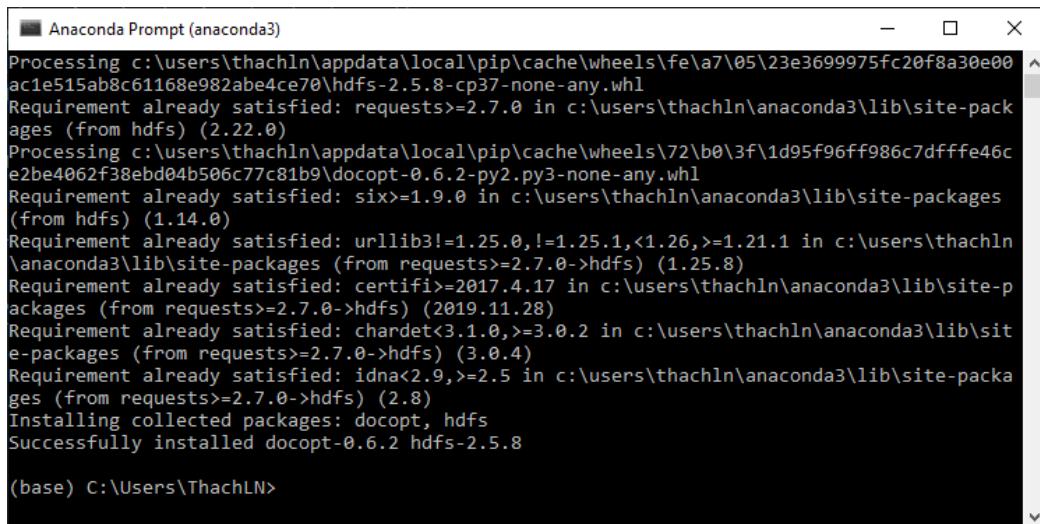
Lưu trữ dữ liệu lên Hadoop

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:

Chạm tới AI trong 10 ngày

```
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:

Chạm tới AI trong 10 ngày

Browsing HDFS

Not secure | ubuntu:50070/explorer.html#/user/root

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

Not secure | ubuntu:50070/explorer.html#/user/root

Hadoop

Overview

Browse Directory

/user/root

Show 25 entries

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()
```

Giới thiệu RHipe

RHipe viết tắt của R and Hadoop Integrated Programming Environment: R và Môi trường lập trình tích hợp Hadoop. Đây là dự án mã nguồn mở để phân tích dữ liệu lớn trong đó sử dụng R trên nền tảng hệ thống Hadoop.

Để trải nghiệm việc phân tích dữ liệu dùng R và Hadoop thì chúng ta cần phải thực hành trên Ubuntu. Hãy nhớ là vừa rồi chúng ta đã có một hệ thống Big Data Hadoop trên Ubuntu. Bây giờ chúng ta sẽ cài thêm Rstudio lên Ubuntu luôn.

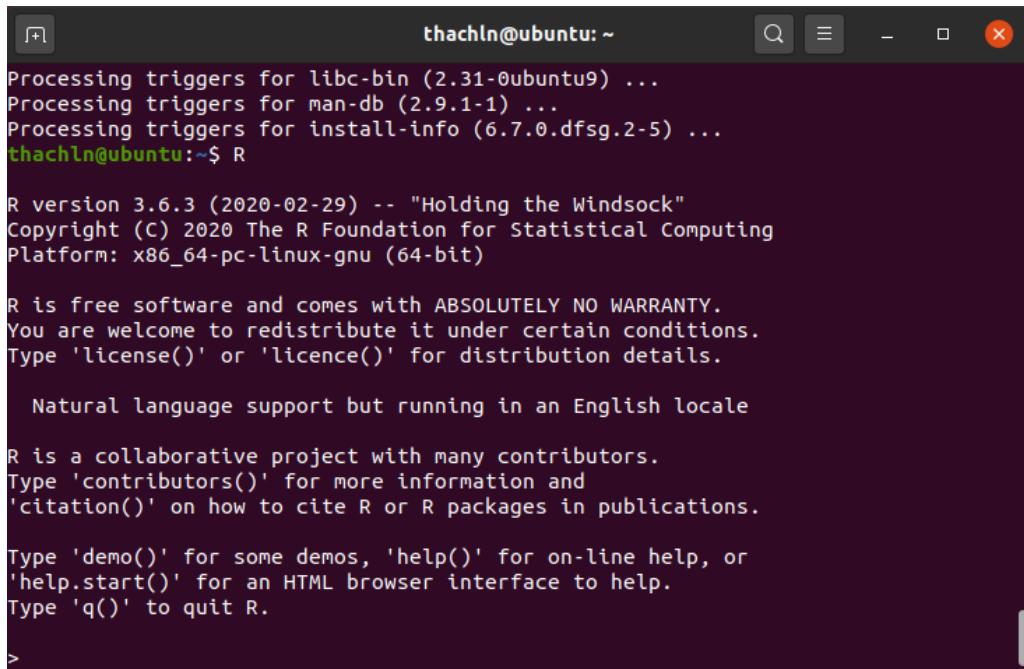
Cài đặt R cho Ubuntu

Đầu tiên cài R bằng dòng lệnh sau:

```
sudo apt -y install r-base
```

Sau khi cài xong, hãy thử R thì sẽ thấy ra dấu nhắc của R:

Chạm tới AI trong 10 ngày



```
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
thachln@ubuntu:~$ R

R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

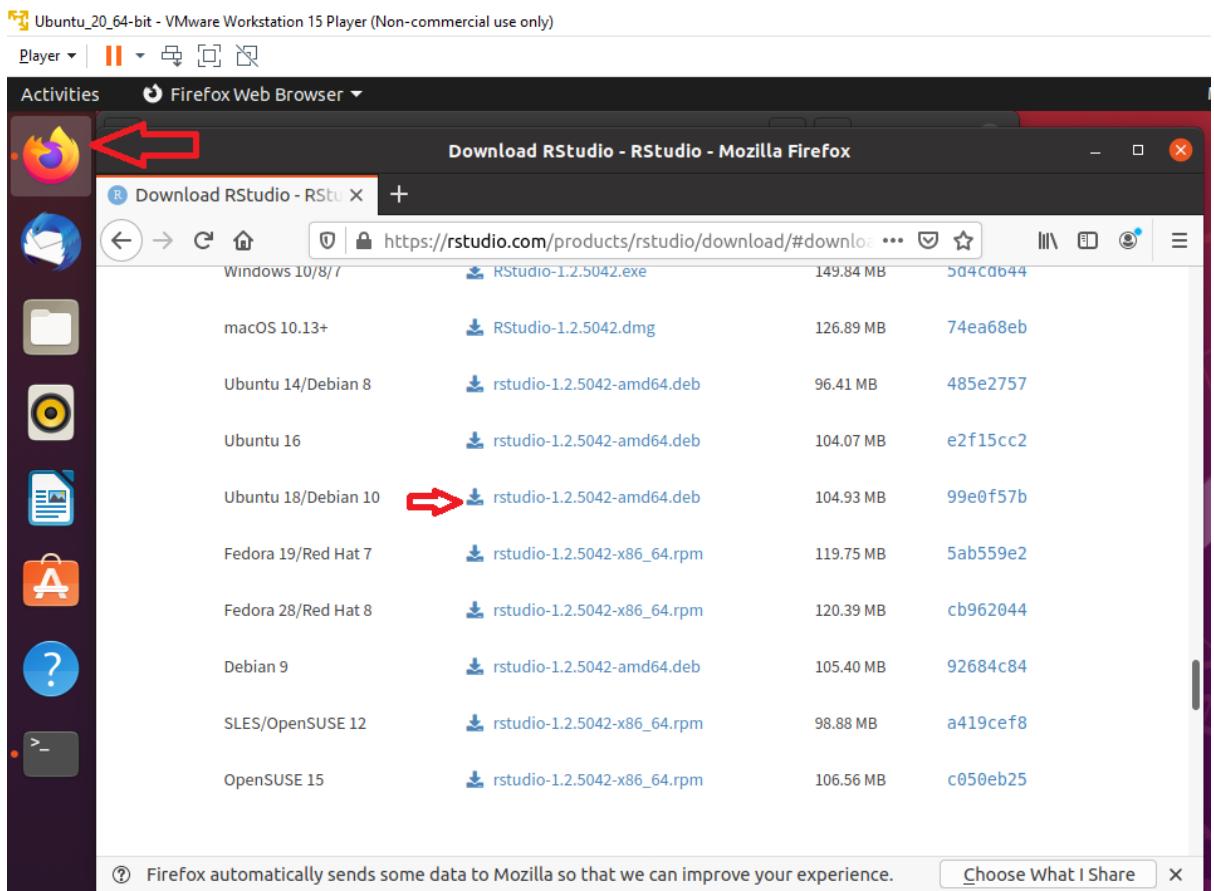
Cài RStudio cho Ubuntu

Tiếp theo chúng ta cần cài RStudio cho Ubuntu. Ngay tại thời điểm này thì chưa có bản RStudio cho Ubuntu 20.xx. Vì thế chúng ta sẽ dùng bản cho Ubuntu 18.

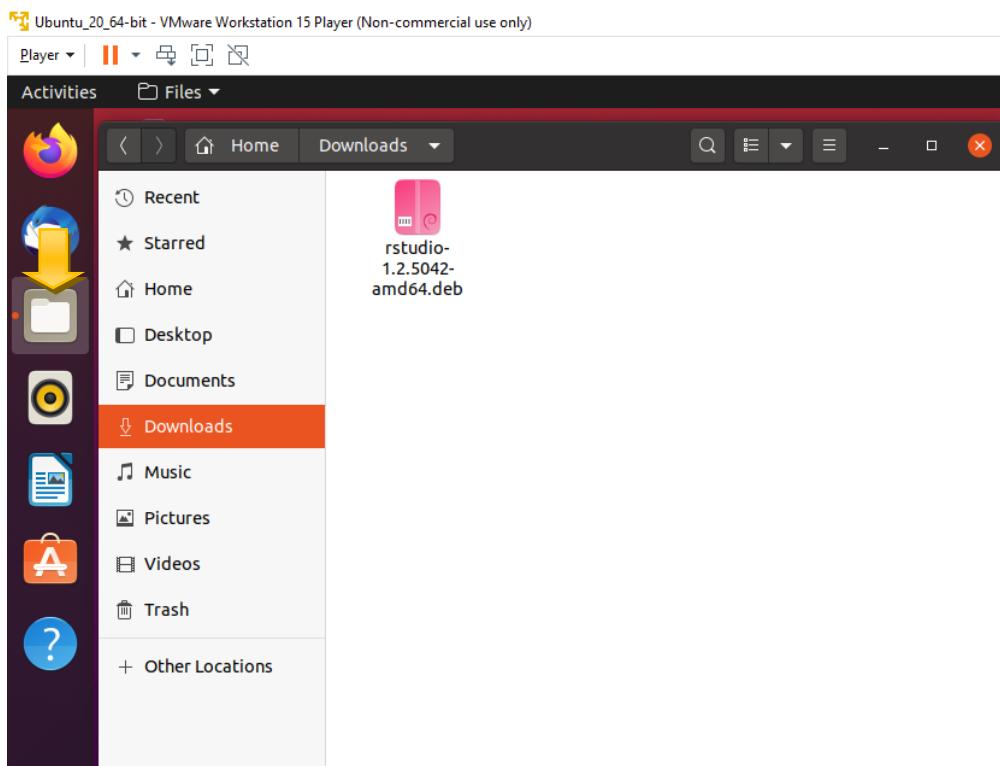
| OS | Download | Size | SHA-256 |
|---------------------|--|-----------|----------|
| Windows 10/8/7 | RStudio-1.2.5042.exe | 149.84 MB | 5d4cd644 |
| macOS 10.13+ | RStudio-1.2.5042.dmg | 126.89 MB | 74ea68eb |
| Ubuntu 14/Debian 8 | rstudio-1.2.5042-amd64.deb | 96.41 MB | 485e2757 |
| Ubuntu 16 | rstudio-1.2.5042-amd64.deb | 104.07 MB | e2f15cc2 |
| Ubuntu 18/Debian 10 | rstudio-1.2.5042-amd64.deb | 104.93 MB | 99e0f57b |
| Fedora 19/Red Hat 7 | rstudio-1.2.5042-x86_64.rpm | 119.75 MB | 5ab559e2 |
| Fedora 28/Red Hat 8 | rstudio-1.2.5042-x86_64.rpm | 120.39 MB | c9b62044 |
| Debian 9 | rstudio-1.2.5042-amd64.deb | 105.40 MB | 92684c84 |
| SLES/OpenSUSE 12 | rstudio-1.2.5042-x86_64.rpm | 98.88 MB | a419cef8 |
| OpenSUSE 15 | rstudio-1.2.5042-x86_64.rpm | 106.56 MB | c050eb25 |

Bên trong máy ảo Ubuntu, bạn mở trình duyệt Firefox rồi truy cập link "<https://rstudio.com/products/rstudio/download/#download>" để download gói "rstudio-1.2.5042-amd64.deb".

Chạm tới AI trong 10 ngày

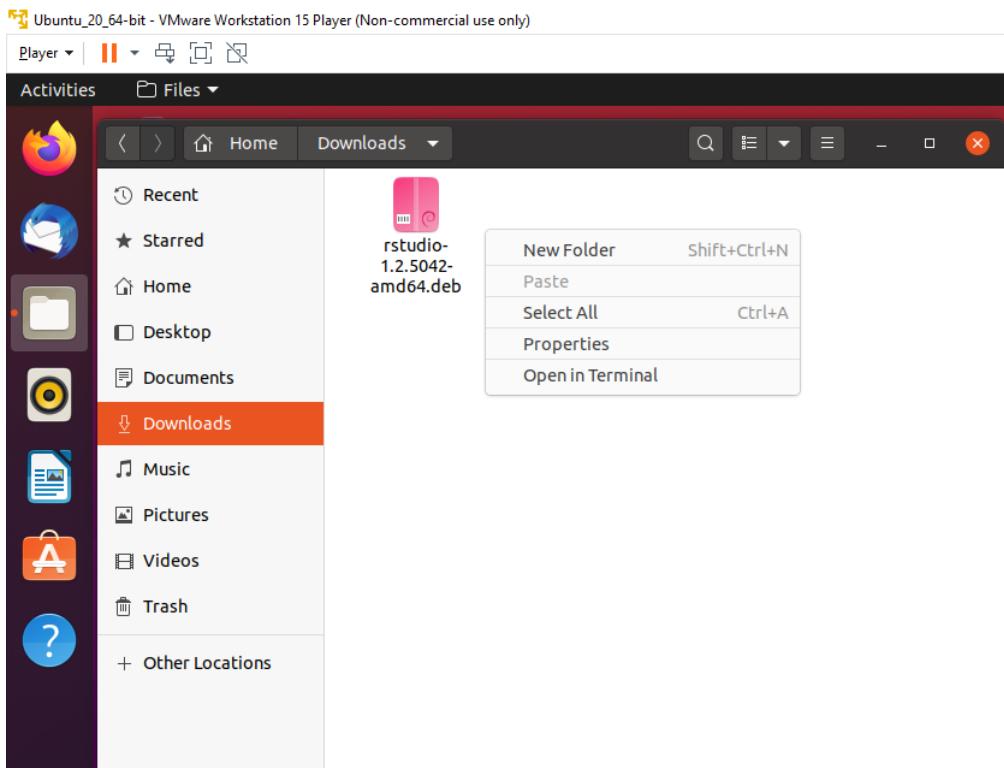


Bạn có thể mở trình quản lý thư mục, vào thư mục Downloads.



Sau đó nhập phải chuột lên vùng trống để hiển thị menu pop up. Chọn Open Terminal để mở cửa sổ lệnh.

Chạm tới AI trong 10 ngày



Cách này có điểm tiện lợi là Terminal được mở với thư mục làm việc là Downloads luôn. Thủ gõ lệnh pwd để biết đường dẫn đầy đủ:

```
pwd
```

```
/home/thachln/Downloads
```

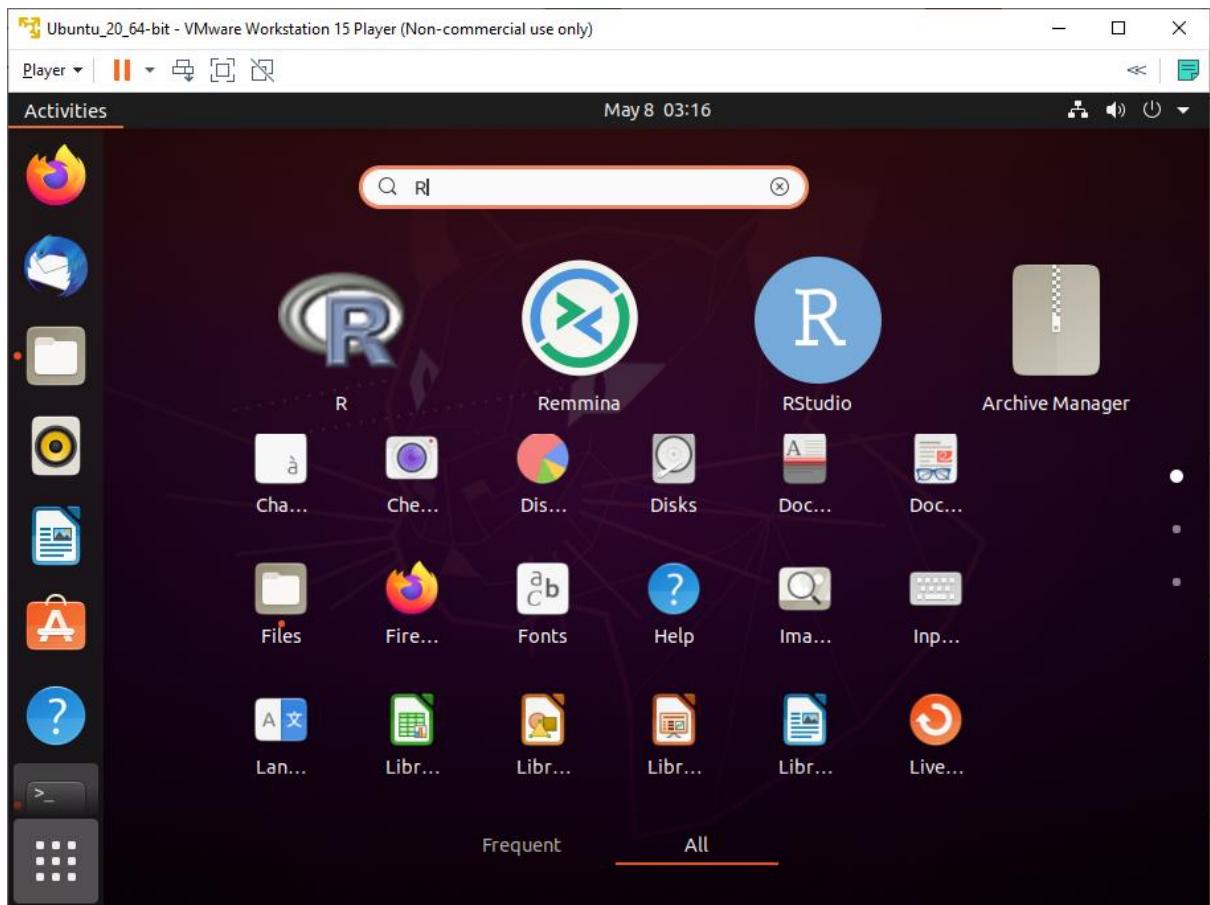
Sau đó gõ “sudo apt install ./” rồi nhấn Tab, bạn sẽ có lệnh đầy đủ:

```
sudo apt install ./rstudio-1.2.5042-amd64.deb
```

Nhấn Enter để cài đặt. Nhập mật khẩu của bạn nếu bị hỏi. Trong quá trình cài đặt nếu có câu hỏi Y/n thì gõ Y và nhấn Enter để tiếp tục.

Sau khi cài xong, vào biểu tượng Show Applications, rõ R trong mục tìm kiếm sẽ thấy biểu tượng của ứng dụng RStudio. Click vào nó để mở RStudio.

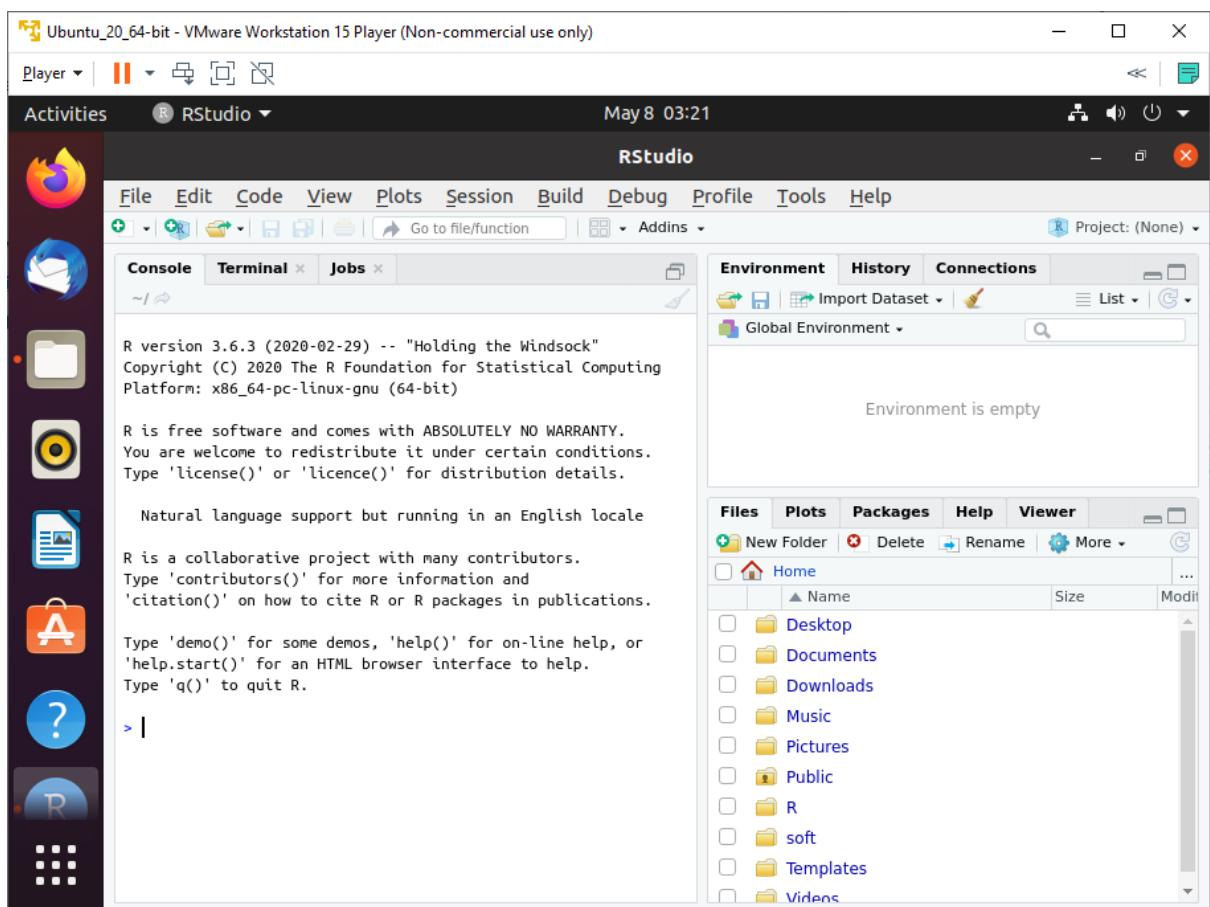
Chạm tới AI trong 10 ngày



Như vậy bạn có thêm một môi trường RStudio trên Ubuntu để khám phá.

Trong trường hợp máy tính của bạn đang sử dụng Windows không có bản quyền thì có thể nghĩ đến việc sử dụng Ubuntu để thay thế Windows. Giao diện RStudio trong Ubuntu cũng giống như trong Windows thôi.

Cài đặt RHadoop



Hãy kết hợp gõ lệnh và phím TAB để tìm đường dẫn của lệnh hadoop trong Terminal:

```
ls /opt/hadoop/bin/hadoop
```

Từ Terminal, thiết lập môi trường Java cho R bằng lệnh:

```
sudo R CMD javareconf
```

Sau đó chạy R từ Terminal luôn:

```
R
```

Kiểm tra biến môi trường HADOOP_CMD và HADOOP_COMMON_LIB_NATIVE_DIR

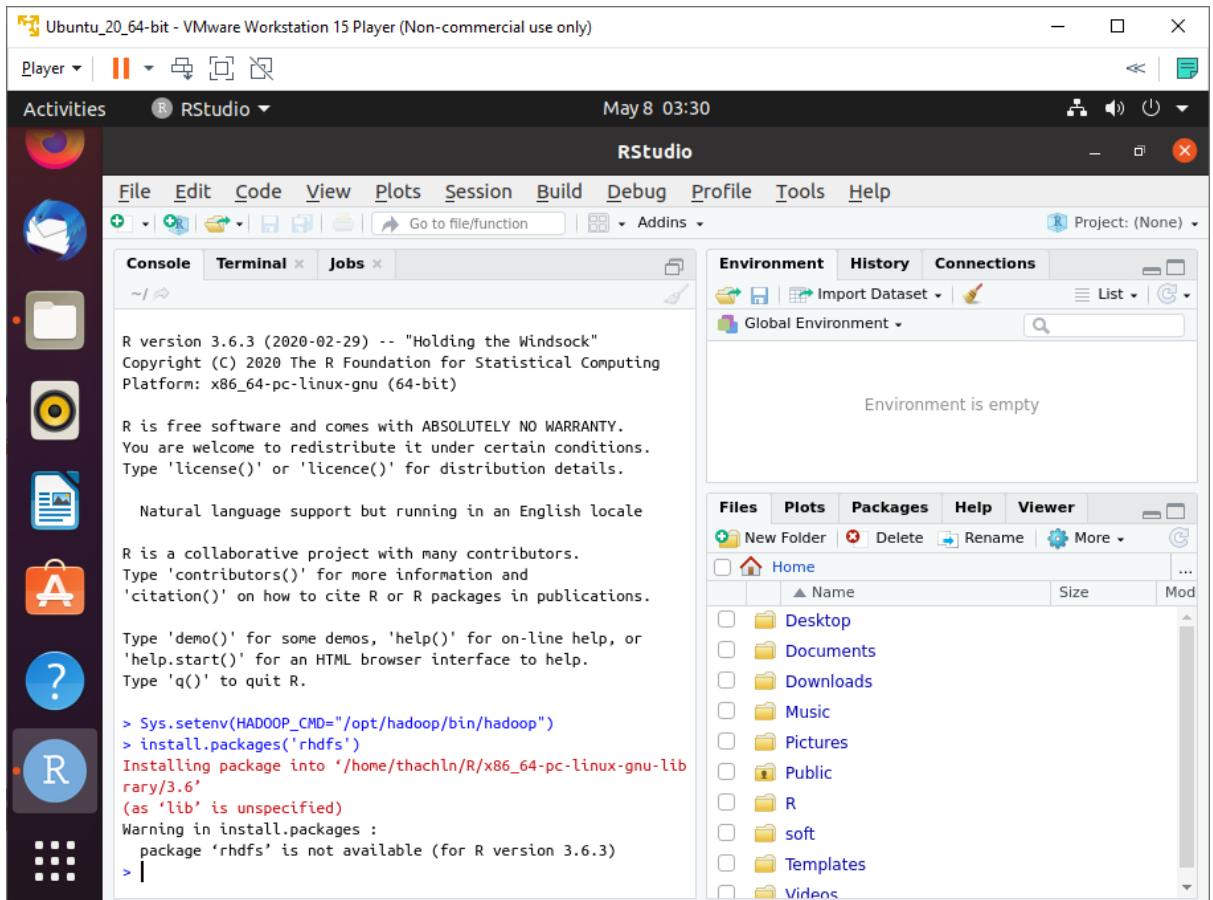
```
Sys.getenv("HADOOP_CMD")
Sys.getenv("HADOOP_COMMON_LIB_NATIVE_DIR")
```

Chạm tới AI trong 10 ngày

Cài đặt thư viện rhdfs cho R:

```
install.packages('rhdfs')
```

Không may là bị báo lỗi:



Thử cài đặt trực tiếp từ github bằng cách sử dụng thư viện “devtools” xem sao?

Cài các thư viện cho Ubuntu trong Terminal với lệnh sau:

```
sudo apt install libxml2 libxml2-dev libcurl4-openssl-dev  
libssl-dev
```

Cài devtools cho RStudio:

```
install.packages('devtools')
```

Cài thư viện rhdfs trực tiếp từ Github bằng cách chạy lệnh sau trong RStudio:

```
devtools::install_github("RevolutionAnalytics/rhdfs", subdir = "pkg")
```

Sử dụng Rhadoop

```
library('rhdbs')
hdfs.init()
hdfs.ls('/user/root/datasets/')
```

Kết quả:

```
permission owner      group    size      modtime
file
1 -rwxr-xr-x root supergroup 5423882 2020-05-08 08:22 /user/root/datasets/bank-
-additional-full.csv
```

Có một số cảnh báo (warning) tạm thời chưa cần để ý!

Thử đọc file csv trực tiếp từ Hadoop bằng cách sử dụng thư viện `data.table` và lệnh `hadoop`:

```
packages <- c('data.table')
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}

library(data.table)
hadoop_cmd = Sys.getenv('HADOOP_CMD')
df = fread(paste(hadoop_cmd, "fs -text /user/root/datasets/bank-
additional-full.csv"))
summary(df)

> df = fread(paste(hadoop_cmd, "fs -text /user/root/datasets/bank-additional-fu
11.csv"))
Taking input= as a system command ('/opt/hadoop/bin/hadoop fs -text /user/root/
datasets/bank-additional-full.csv') and a variable has been used in the express
ion passed to `input=`. Please use fread(cmd=....). There is a security concern
if you are creating an app, and the app could have a malicious user, and the ap
p is not running in a secure environment; e.g. the app is running as root. Plea
se read item 5 in the NEWS file for v1.11.6 for more information and for the op
tion to suppress this message.
20/05/08 21:43:59 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
> summary(df)
      V1           age          job        marital      educati
on
      default      housing
      Min.   : 0   Min.   :17.00  Length:41188   Length:41188   Length:4
1188   Length:41188   Length:41188
      1st Qu.:10297  1st Qu.:32.00  Class :character  Class :character  Class :c
haracter  Class :character  Class :character
      Median :20594  Median :38.00  Mode  :character  Mode  :character  Mode  :c
haracter  Mode  :character  Mode  :character
      Mean   :20594  Mean   :40.02
      3rd Qu.:30890  3rd Qu.:47.00
      Max.   :41187  Max.   :98.00
      loan
duration      contact
      campaign      month
      pdays
      Length:41188  Length:41188  Length:41188  Length:41188
n.   : 0.0   Min.   : 1.000  Min.   : 0.0   Length:41188
      day_of_week
      Mi
```

Chạm tới AI trong 10 ngày

```
  class :character  class :character  class :character  class :character  1s
t Qu.: 102.0  1st Qu.: 1.000  1st Qu.:999.0
  Mode :character  Mode :character  Mode :character  Mode :character  Me
dian : 180.0  Median : 2.000  Median :999.0
  Me
an : 258.3  Mean : 2.568  Mean :962.5
  3r
d Qu.: 319.0  3rd Qu.: 3.000  3rd Qu.:999.0
  Ma
x. :4918.0  Max. :56.000  Max. :999.0
  previous poutcome emp.var.rate cons.price.idx cons.con
f.idx  euribor3m nr.employed y
  Min. :0.000  Length:41188  Min. :-3.40000  Min. :92.20  Min. :
-50.8  Min. :0.634  Min. :4964  Length:41188
  1st Qu.:0.000  Class :character  1st Qu.:-1.80000  1st Qu.:93.08  1st Qu.:
-42.7  1st Qu.:1.344  1st Qu.:5099  Class :character
  Median :0.000  Mode :character  Median : 1.10000  Median :93.75  Median :
-41.8  Median :4.857  Median :5191  Mode :character
  Mean :0.173  Mean : 0.08189  Mean : 93.58  Mean :
-40.5  Mean :3.621  Mean :5167  3rd Qu.: 1.40000  3rd Qu.:93.99  3rd Qu.:
-36.4  3rd Qu.:4.961  3rd Qu.:5228  Max. : 1.40000  Max. :94.77  Max. :
  Max. :7.000  Max. :5228
-26.9  Max. :5.045  Max. :5228
```

Có một số cảnh báo (warning) tạm thời chưa cần để ý!

Tóm tắt:

Trong trường hợp tổ chức của bạn có nhu cầu phân tích dữ liệu lớn thì giải pháp khai thác Hadoop, R và Python là một trong các giải pháp khả thi. Việc vận dụng linh hoạt các thư viện này sẽ giúp cho ý tưởng khai thác dữ liệu lớn có tính khả thi cao với chi phí bản quyền phần mềm gần như bằng 0. Hy vọng các ví dụ sơ khai ở trên sẽ giúp bạn nhìn ra hướng đi cho nhu cầu thực tế của mình.