**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

**BÀI TẬP LỚN**

**PHÂN TÍCH DỮ LIỆU MỨC LƯƠNG CẦU THỦ CỦA CÁC ĐỘI BÓNG CHÂU ÂU**

**NGÀNH: KHOA HỌC MÁY TÍNH**

**CHUYÊN NGÀNH: TRÍ TUỆ NHÂN TẠO VÀ KHOA HỌC DỮ LIỆU**

**SINH VIÊN: NGUYỄN THÀNH HUY**

**MÃ LỚP: 124221**

**NGƯỜI HƯỚNG DẪN: TS. HOÀNG QUỐC VIỆT**

**HƯNG YÊN – 2024**

**NHẬN XÉT**

**Nhận xét của giáo viên hướng dẫn**

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

**GIÁO VIÊN HƯỚNG DẪN**

**Hoàng Quốc Việt**

**LỜI CAM ĐOAN**

Em xin cam đoan bài tập lớn môn lập trình Python nâng cao có tên là “Phân tích dữ liệu về mức lương cầu thủ của các câu lạc bộ bóng đá châu Âu” là sản phẩm của bản thân em.

Những phần sử dụng tài liệu tham khảo trong bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong bài tập lớn hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

*Hưng Yên, ngày … tháng 06 năm 2024*

**Sinh viên**

**Nguyễn Thành Huy**

**LỜI CẢM ƠN**

Để có thể hoàn thành bài tập lớn này, lời đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Khoa học máy tính, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện bài tập lớn môn học này.

Đặc biệt em xin chân thành cảm ơn thầy Hoàng Quốc Việt đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện bài tập lớn vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được bài tập lớn này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hy vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy cô về những kết quả triển khai trong bài tập lớn.

Em xin trân trọng cảm ơn

**MỤC LỤC**

[DANH MỤC HÌNH VẼ 2](#_Toc14411)

[CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN 5](#_Toc12977)

[1.1 Bài toán 5](#_Toc29375)

[1.2 Trình bày dữ liệu bài toán 5](#_Toc5385)

[1.3 Tiền xử lý dữ liệu 7](#_Toc30269)

[1.4 Thống kê dữ liệu 8](#_Toc27679)

[1.5 Trực quan hoá dữ liệu 8](#_Toc10478)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 9](#_Toc5710)

[2.1 Pandas 9](#_Toc9526)

[2.2 Matplotlib 12](#_Toc27673)

[CHƯƠNG 3: GIẢI PHÁP 25](#_Toc4745)

[3.1 Mã nguồn tiền xử lý dữ liệu 25](#_Toc27678)

[3.2 Mã nguồn chức năng Thống kê dữ liệu 27](#_Toc32295)

[3.3 Mã nguồn Trực quan hoá dữ liệu 33](#_Toc7010)

[3.4 Mô hình học máy (Machine Learning) 46](#_Toc31387)

[CHƯƠNG 4: TÀI LIỆU THAM KHẢO 52](#_Toc25954)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1 . Dữ liệu của file csv 6](#_Toc10130)

[Hình 1.2 . Mô tả các features có trong file dữ liệu 6](#_Toc20404)

[Hình 1.3 . Mô tả dữ liệu của file csv 7](#_Toc28334)

[HÌnh 2.1 . Cài đặt Pandas 11](#_Toc32712)

[HÌnh 2.2 . Sử dụng thư viện Pandas 11](#_Toc12566)

[Hình 2.3 . Ví dụ tạo Series 11](#_Toc24053)

[Hình 2.4 . Ví dụ tạo DataFrame từ dict 12](#_Toc32365)

[Hình 2.5 . Ví dụ chọn ô trong DataFrame 12](#_Toc15003)

[Hình 2.6 . Ví dụ tạo biểu đồ đơn giản với Matplotlib 15](#_Toc2715)

[Hình 2.7 . Tạo tiêu đề cho các trục biểu đồ 16](#_Toc20101)

[Hình 2.8 . Vẽ biểu đồ với kích thước tự chọn 16](#_Toc26482)

[Hình 2.9 . Định dạng kí hiệu dữ liệu trên biểu đồ 17](#_Toc25011)

[Hình 2.10 . Vẽ nhiều biểu đồ với phương thức subplots() 18](#_Toc402)

[Hình 2.11 . Tạo biểu đồ phân phối bằng Seaborn 21](#_Toc20316)

[Hình 2.12 . Tạo biểu đồ phân tán bằng Seaborn 22](#_Toc25737)

[Hình 2.13 . Tạo biểu đồ cột bằng Seaborn 23](#_Toc2867)

[Hình 2.14 . Tạo biểu đồ heatmap bằng Seaborn 24](#_Toc2623)

[Hình 3.1 . Thông tin của các feature 25](#_Toc3050)

[Hình 3.2 . Các feature có tổng số bị trống dữ liệu 25](#_Toc14738)

[Hình 3.3 . Xử lý datatype trong series 26](#_Toc16116)

[Hình 3.4 . Tổng số feature bị trùng lặp dữ liệu 26](#_Toc14397)

[Hình 3.5 . Dữ liệu sau khi xoá các dữ liệu bị trùng 26](#_Toc27024)

[Hình 3.6 . Bảng dữ liệu của “Apps” và “Caps” đều có giá trị 0 27](#_Toc21502)

[Hình 3.7 . Mã nguồn thống kê mức lương trung bình của các cầu thủ qua các độ tuổi 27](#_Toc6915)

[Hình 3.8 . Mã nguồn thống kê mức lương trung bình của các cầu thủ cho các vị trí 28](#_Toc1784)

[Hình 3.9 . Mã nguồn thống kê trung bình mức lương nhận được của các châu lục 29](#_Toc5156)

[Hình 3.10 . Mã nguồn thống kê tổng lương được trả qua các giải đấu 29](#_Toc15190)

[Hình 3.11 . Mã nguồn thống kê tổng lương nhận được trên và dưới 30 tuổi 30](#_Toc18462)

[Hình 3.12 . Mã nguồn thống kê trung bình mức lương của các vị trí trong các giải đấu 30](#_Toc29233)

[Hình 3.13 . Mã nguồn thống kê top 10 Club trả lương cầu thủ nhiều nhất 31](#_Toc23154)

[Hình 3.14 .Thống kê tổng lương phải trả nhiều nhất cho độ tuổi và vị trí của các Club trong top 15 32](#_Toc20915)

[Hình 3.15 . Mã nguồn thống kê tổng lương các cầu thủ không được sử dụng trong club và quốc gia của các giải đấu 32](#_Toc15931)

[Hình 3.16 . Mã nguồn hống kê top 10 tổng lương phải trả nhiều nhất của các cầu thủ không được sử dụng trong Club và quốc gia của các câu lạc bộ 33](#_Toc24189)

[Hình 3.17 . Thư viện hỗ trợ cho trực quan hoá dữ liệu 33](#_Toc23222)

[Hình 3.18 . Mã nguồn thể hiện top 10 câu lạc bộ trả lương nhiều nhất 34](#_Toc25671)

[Hình 3.19 . Biểu đồ thể hiện top 10 câu lạc bộ trả lương nhiều nhất 35](#_Toc22785)

[Hình 3.20 . Mã nguồn thể hiện mức lương trung bình cho từng vị trí 35](#_Toc20934)

[Hình 3.21 . Biểu đồ thể hiện mức lương trung bình cho từng vị trí 37](#_Toc32547)

[Hình 3.22 . Mã nguồn thể hiện tổng mức lương cho các vị trí trong các giải đấu 37](#_Toc3027)

[Hình 3.23 . Biểu đồ thể hiện tổng mức lương cho các vị trí trong các giải đấu 38](#_Toc15437)

[Hình 3.24 . Mã nguồn thể hiện tổng mức lương của các châu lục trong các giải đấu 38](#_Toc31689)

[Hình 3.25 . Biểu đồ thể hiện tổng mức lương của các châu lục trong các giải đấu 39](#_Toc19040)

[Hình 3.26 . Mã nguồn thể hiện mức lương nhận được dựa trên số lần đá cho quốc gia trong các giải đấu 39](#_Toc23855)

[Hình 3.27 .Biểu đồ thể hiện thể hiện mức lương nhận được dựa trên số lần đá cho quốc gia trong các giải đấu 40](#_Toc2222)

[Hình 3.28 . Mã nguồn thể hiện mức lương nhận được dựa trên số lần đá cho Club trong các giải đấu 40](#_Toc3735)

[Hình 3.29 . Biểu đồ thể hiện mức lương nhận được dựa trên số lần đá cho Club trong các giải đấu 41](#_Toc29701)

[Hình 3.30 . Mã nguồn top mức lương được trả cao nhất cho các vị trí trong top 10 Club trả lương cầu thủ nhiều nhất 41](#_Toc27077)

[Hình 3.31 .Biểu đồ top mức lương được trả cao nhất cho các vị trí trong top 10 Club trả lương cầu thủ nhiều nhất 42](#_Toc24554)

[Hình 3.32 . Mã nguồn ma trận của cầu thủ 42](#_Toc26388)

[Hình 3.33 .Biểu đồ ma trận của cầu thủ 43](#_Toc21794)

[Hình 3.34 . Mã nguồn top 20 Club có tổng lương cao nhất và tổng lương ít nhất 44](#_Toc4118)

[Hình 3.35 . Biểu đồ top 20 Club có tổng lương cao nhất và tổng lương ít nhất 46](#_Toc14711)

[Hình 3.36 . Cài đặt scikit - learn 47](#_Toc1403)

[Hình 3.37 . Cài đặt scikit - learn bằng conda 48](#_Toc9400)

[Hình 3.38 . Sử dụng các thư viện để ứng dụng mô hình học máy 49](#_Toc7220)

[Hình 3.39 . Mã hoá dữ liệu với dạng dữ liệu là văn bản 49](#_Toc1794)

[Hình 3.40 . Sử dụng OneHotEncoder() để chuyển giá trị phân loại thành vecto nhị phân 49](#_Toc21976)

[Hình 3.41 . Lấy giá trị Y bằng cột Wage để dự đoán Wage 50](#_Toc1086)

[Hình 3.42 . Lựa chọn số lượng train model và test model 50](#_Toc14406)

[Hình 3.43 . Xây dựng mô hình hồi quy tuyến tính 50](#_Toc10475)

[Hình 3.44 . Biểu đồ sự phân tán dữ liệu và dữ liệu dự đoán 51](#_Toc18572)

# GIỚI THIỆU BÀI TOÁN

## Bài toán

Hiện nay, việc phân tích dữ liệu về mức lương của các cầu thủ bóng đá châu Âu đang trở thành một lĩnh vực nghiên cứu quan trọng. Bằng cách phân tích những dữ liệu này, chúng ta có thể thu thập thông tin chi tiết về mức lương của các cầu thủ từ các câu lạc bộ khác nhau, từ đó cung cấp cái nhìn tổng quan về xu hướng lương và mức độ ưa chuộng của các cầu thủ trong lĩnh vực bóng đá.

Dữ liệu về mức lương của các cầu thủ bóng đá châu Âu thường được thu thập từ các nguồn tin cậy và có tính tin cậy cao. Ngoài ra, dữ liệu này thường được cập nhật thường xuyên, giúp đảm bảo tính sẵn có của dữ liệu và đáng tin cậy trong quá trình phân tích.

Thị trường bóng đá châu Âu có sự ảnh hưởng toàn cầu đến các thị trường khác trên thế giới. Việc phân tích dữ liệu về mức lương của các cầu thủ bóng đá châu Âu giúp hiểu rõ hơn về ảnh hưởng của xu hướng và thị trường bóng đá châu Âu đến các thị trường quốc tế khác, từ đó có thể áp dụng những kết quả phân tích để tối ưu hóa hoạt động kinh doanh và phát triển thị trường bóng đá.

Chúng ta đã có dữ liệu và bây giờ chúng ta sẽ phải khai thác dữ liệu đó giúp người dùng biết rõ hơn về thông tin của các cầu thủ bóng đá châu Âu, giúp người dùng dễ dàng lựa chọn được cầu thủ phù hợp nhất cho đội bóng của mình.

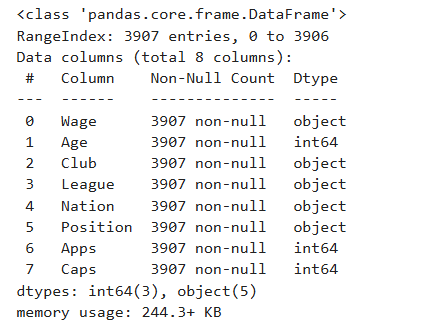
## Trình bày dữ liệu bài toán

Dữ liệu này được lấy từ trang web:

[**https://www.kaggle.com/datasets/ultimus/football-wages-prediction**](https://www.kaggle.com/datasets/ultimus/football-wages-prediction)

****

Hình 1.1. Dữ liệu của file csv

****

Hình 1.2. Mô tả các features có trong file dữ liệu

- Dữ liệu bài toán gồm các feature sau:

+ Wage : Mức lương của cầu thủ nhận được

+ Age : Độ tuổi của cầu thủ

+ Club : Tên của các đội bóng

+ Nation : Quốc tịch của cầu thủ

+ Position : Vị trí chơi bóng của cầu thủ

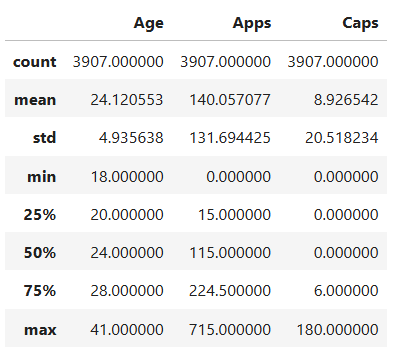
+ Apps : Số lần chơi bóng cho câu lạc bộ

+ Caps : Số lần chơi bóng cho quốc tịch của các cầu thủ

- Dữ liệu bài toán là 1 file csv gồm 3907 rows × 8 columns

+ Có 8 feature và mỗi feature có 3907 dữ liệu đầu vào

-Sau khi mô tả dữ liệu ta có:

****

Hình 1.3. Mô tả dữ liệu của file csv

+ Độ tuổi cao nhất là 41 và nhỏ nhất là 18

+ Số lần chơi bóng cho đội cao nhất là 715 và nhỏ nhất là 0

+ Số lần chơi bóng cho quốc gia cao nhất là 180 và nhỏ nhất là 0

## Tiền xử lý dữ liệu

+) Chuyển kiểu dữ liệu cột Wage từ object sang float

## Thống kê dữ liệu

1. Thống kê mức lương trung bình của các cầu thủ qua các độ tuổi
2. Thống kê mức lương trung bình của các cầu thủ cho các vị trí trong Club
3. Thống kê trung bình mức lương được nhận của các cầu thủ theo các châu lục khác nhau
4. Thống kê tổng lương được trả qua các giải đấu
5. Thống kê tổng lương của các cầu thủ trên và dưới 30 tuổi
6. Thống kê trung bình mức lương nhận được của các vị trí ở trong các giải đấu
7. Thống kê tổng lương phải trả nhiều nhất của các câu lạc bộ trong top 10
8. Thống kê tổng lương phải trả nhiều nhất cho độ tuổi và vị trí của các câu lạc bộ trong top 15
9. Thống kê tổng lương các cầu thủ không được sử dụng trong club và quốc gia của các giải đấu
10. Thống kê top 10 tổng lương phải trả nhiều nhất của các cầu thủ không được sử dụng trong Club và quốc gia của các câu lạc bộ

## Trực quan hoá dữ liệu

1. Biểu đồ thể hiện top 10 câu lạc bộ trả lương nhiều nhất
2. Biểu đồ thể hiện mức lương trung bình cho từng vị trí
3. Biểu đồ thể hiện tổng mức lương cho các vị trí trong các giải đấu
4. Biểu đồ thể hiện tổng mức lương của các châu lục trong các giải đấu
5. Biểu đồ thể hiện mức lương nhận được dựa trên số lần đá cho quốc gia trong các giải đấu
6. Biểu đồ thể hiện mức lương nhận được dựa trên số lần đá cho Club trong các giải đấu
7. Biểu đồ top mức lương được trả cao nhất cho các vị trí trong top 10 Club trả lương cầu thủ cao nhất
8. Biểu đồ ma trận của cầu thủ
9. Biểu đồ top 20 Club có tổng lương cao nhất và tổng lương thấp nhất

# CƠ SỞ LÝ THUYẾT

## Pandas

### Giới thiệu về Pandas

**Tổng quan về Pandas**

Pandas sẽ là một công cụ quan trọng trong phần lớn phần còn lại của cuốn sách. Nó chứa các cấu trúc dữ liệu và công cụ thao tác dữ liệu được thiết kế để làm cho việc làm sạch dữ liệu và phân tích nhanh chóng và dễ dàng trong Python. Pandas thường được sử dụng cùng với các công cụ tính toán số học như NumPy và SciPy, các thư viện phân tích như statsmodels và scikit-learn, và các thư viện trực quan hóa dữ liệu như matplotlib. Pandas tiếp nhận các phần quan trọng của phong cách lập trình dựa trên mảng của NumPy, đặc biệt là các hàm dựa trên mảng và sự ưu tiên cho việc xử lý dữ liệu mà không cần vòng lặp for.

Pandas cho phép chúng ta phân tích dữ liệu lớn và đưa ra kết luận dựa trên lý thuyết thống kê.

Pandas có thể dọn dẹp các tập dữ liệu lộn xộn và làm cho chúng dễ đọc và phù hợp.

Dữ liệu liên quan là rất quan trọng trong khoa học dữ liệu.

Pandas cung cấp cho bạn câu trả lời về dữ liệu. Giống:

* Có mối tương quan giữa hai hoặc nhiều cột không?
* Giá trị trung bình là gì?
* Giá trị tối đa?
* Giá trị tối thiểu?

Gấu trúc cũng có thể xóa các hàng không liên quan hoặc chứa các giá trị sai, như giá trị trống hoặc giá trị NULL. Điều này được gọi là làm sạch dữ liệu

**Vì sao bạn nên chọn pandas?**

* Pandas rất phù hợp với nhiều loại dữ liệu khác nhau:
* Dữ liệu dạng bảng với các cột được nhập không đồng nhất, như trong bảng SQL hoặc bảng tính Excel.
* Dữ liệu chuỗi thời gian theo thứ tự và không có thứ tự (không nhất thiết phải có tần số cố định).
* Dữ liệu ma trận tùy ý (được nhập đồng nhất hoặc không đồng nhất) với nhãn hàng và cột.
* Bất kỳ hình thức khác của các bộ dữ liệu quan sát / thống kê. Dữ liệu thực sự không cần phải được dán nhãn vào cấu trúc dữ liệu pandas.
* Pandas được xây dựng dựa trên NumPy. Hai cấu trúc dữ liệu chính của pandas là Series (1 chiều) và DataFrame (2 chiều) xử lý được phần lớn các trường hợp điển hình trong tài chính, thống kê, khoa học xã hội và nhiều lĩnh vực kỹ thuật.

**Ưu điểm của pandas:**

* Dễ dàng xử lý dữ liệu mất mát, được biểu thị dưới dạng NaN, trong dữ liệu dấu phẩy động cũng như dấu phẩy tĩnh theo ý người dùng mong muốn: bỏ qua hoặc chuyển sang 0
* Khả năng thay đổi kích thước: các cột có thể được chèn và xóa khỏi DataFrame và các đối tượng chiều cao hơn
* Căn chỉnh dữ liệu tự động và rõ ràng: các đối tượng có thể được căn chỉnh rõ ràng với một bộ nhãn hoặc người dùng chỉ cần bỏ qua các nhãn và để Series, DataFrame, v.v. tự động căn chỉnh dữ liệu cho bạn trong các tính toán
* Chức năng group by mạnh mẽ, linh hoạt để thực hiện các hoạt động kết hợp phân tách áp dụng trên các tập dữ liệu, cho cả dữ liệu tổng hợp và chuyển đổi
* Dễ dàng chuyển đổi dữ liệu rời rạc (ragged), chỉ mục khác nhau (differently-indexed) trong các cấu trúc dữ liệu khác của Python và NumPy thành các đối tượng DataFrame
* Cắt lát (slicing) thông minh dựa trên nhãn, lập chỉ mục ưa thích (fancy indexing) và tập hợp lại (subsetting) các tập dữ liệu lớn
* Gộp (merging) và nối (joining) các tập dữ liệu trực quan
* Linh hoạt trong định hình lại (reshaping) và xoay (pivoting) các tập dữ liệu
* Dán nhãn phân cấp (hierarchical) của các trục (có thể có nhiều nhãn trên mỗi đánh dấu)
* Các công cụ IO mạnh mẽ để tải dữ liệu từ các tệp phẳng (flat file) như CSV và delimited, tệp Excel, cơ sở dữ liệu và lưu / tải dữ liệu từ định dạng HDF5 cực nhanh
* Chức năng theo chuỗi thời gian (time series) cụ thể: tạo phạm vi ngày và chuyển đổi tần số, thống kê cửa sổ di chuyển, dịch chuyển ngày và độ trễ.
* Tích hợp tốt với các thư viện khác của python như SciPy, Matplotlib, Plotly, v.v.
* Hiệu suất tốt

### Cài đặt và cách sử dụng Pandas

****

HÌnh 2.1. Cài đặt Pandas

Sau khi cài đặt xong chúng ta sẽ sử dụng Pandas bằng câu lệnh: import Pandas

****

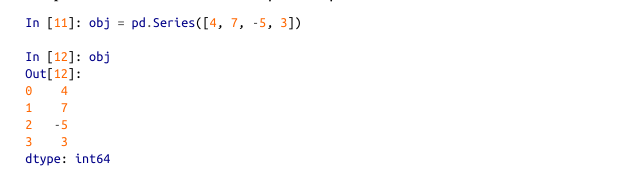
HÌnh 2.2. Sử dụng thư viện Pandas

### Các thành phần chính trong Pandas

* **Series**

Một Series là một đối tượng giống mảng một chiều chứa một chuỗi giá trị (tương tự như các loại của NumPy) và một mảng dữ liệu liên kết được gọi là chỉ mục của nó

Ví Dụ: Tạo 1 series



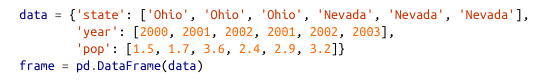
Hình 2.3. Ví dụ tạo Series

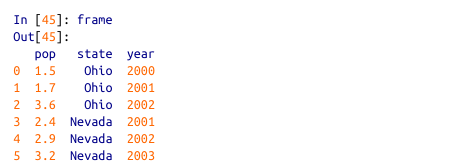
* **Dataframe**

DataFrame đại diện cho một bảng dữ liệu hình chữ nhật và chứa một tập hợp các cột có thứ tự, mỗi cột có thể là một loại giá trị khác nhau (số, chuỗi, boolean, v.v.). DataFrame có cả chỉ mục hàng và cột; nó có thể được coi là một mệnh lệnh của Series tất cả chia sẻ cùng một chỉ mục. Dưới mui xe, dữ liệu được lưu trữ dưới dạng một hoặc nhiều khối hai chiều thay vì danh sách, lệnh hoặc một số tập hợp các mảng một chiều khác.

Có nhiều cách để xây dựng DataFrame, mặc dù một trong những cách phổ biến nhất là từ một chuỗi các danh sách có độ dài bằng nhau hoặc mảng NumPy

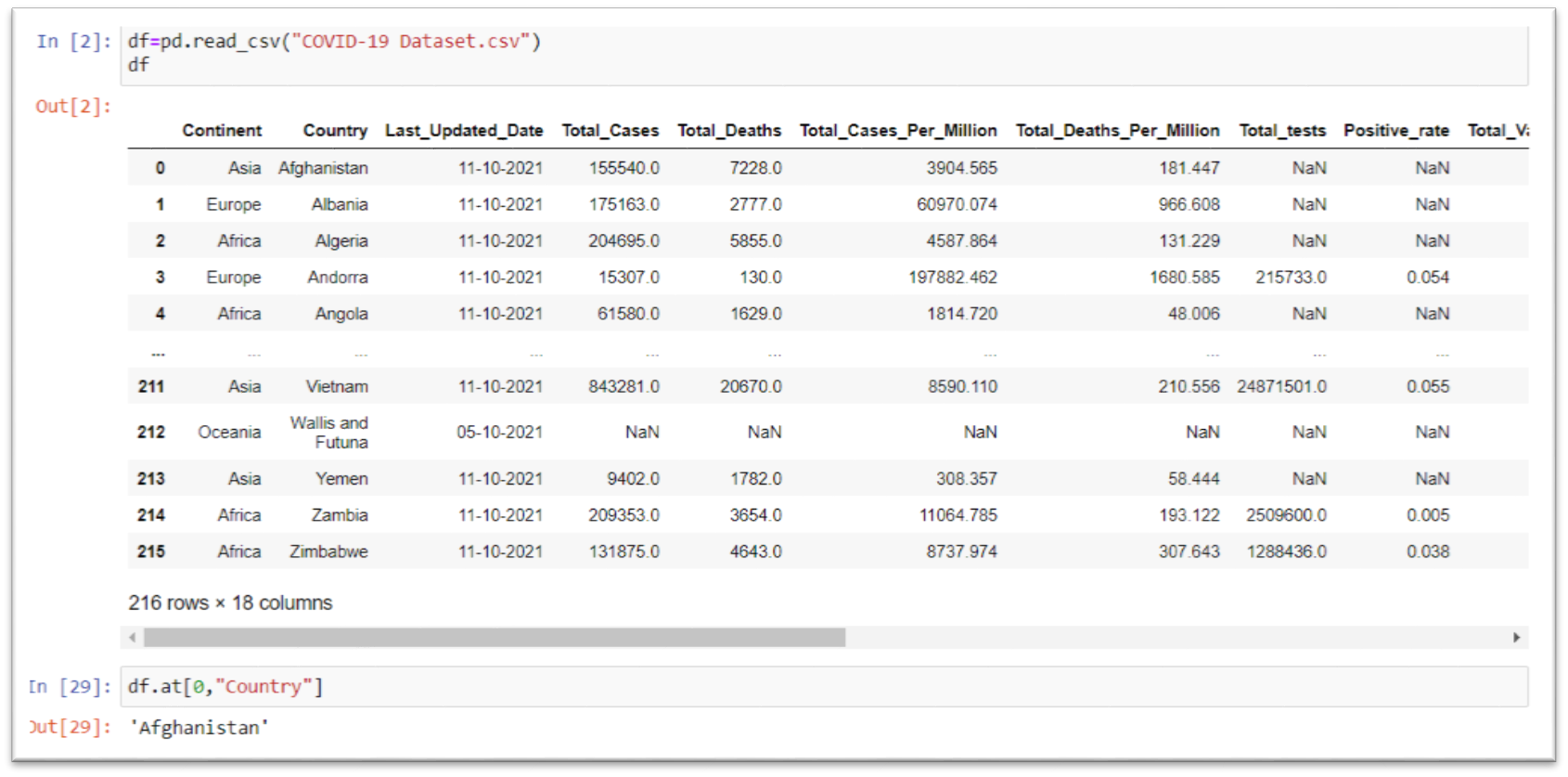
- Ví dụ: Tạo DataFram từ dict





Hình 2.4. Ví dụ tạo DataFrame từ dict

- Ví dụ: Truy xuất trong DataFrame (chọn ô tron DataFrame)



Hình 2.5. Ví dụ chọn ô trong DataFrame

## Matplotlib

Matplotlib là một thư viện vẽ đồ thị cho ngôn ngữ lập trình Python và được mở rộng từ thư viện toán học số học NumPy. Nó cung cấp một API hướng đối tượng để nhúng các biểu đồ vào ứng dụng sử dụng các toolkit GUI đa năng như Tkinter, wxPython, Qt hoặc GTK. Matplotlib cho phép bạn tạo và hiển thị các biểu đồ, hình ảnh và các hình vẽ khác, rất hữu ích cho việc trực quan hóa dữ liệu trong Python. Một phần quan trọng của Matplotlib là Pyplot, một module cung cấp các hàm đơn giản để thêm các thành phần plot như đường, hình ảnh, văn bản và nhiều hơn nữa vào các khung vẽ (axes).

Matplotlib là một trong những thư viện Python phổ biến nhất được sử dụng để trực quan hóa dữ liệu. Nó là một thư viện đa nền tảng để tạo các đồ thị 2D từ dữ liệu trong các mảng. Matplotlib được viết bằng Python và sử dụng NumPy, phần mở rộng toán học của Python. Nó cung cấp một API hướng đối tượng giúp nhúng các plot trong các ứng dụng và sử dụng bộ công cụ GUI Python như PyQt, WxPythonotTkinter. Ngoài ra có thể được sử dụng trong Python và IPython shell, Jupyter Notebook và các máy chủ web.

Matplotlib có giao diện được đặt tên là Pylab, được thiết kế giống với MATLAB - ngôn ngữ lập trình độc quyền được phát triển bởi MathWorks. Matplotlib cùng với NumPy có thể được coi là mã nguồn mở tương đương với MATLAB.

Matplotlib ban đầu được viết bởi John D. Hunter vào năm 2003. Phiên bản ổn định hiện tại là 2.2.0 được phát hành vào tháng 1 năm 2018

### Matplotlib dùng để làm gì?

Để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu của bạn và Matplotlib là một trong những giải pháp như vậy cho người dùng Python. Nó là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở.

Để cài đặt Matplotlib nếu bạn có Anaconda chỉ cần gõ conda install matplotlib hoặc sử dụng tools pip pip install matplotlib

### Khái niệm chung

Một Matplotlib figure có thể được phân loại thành nhiều phần như dưới đây:

Figure: Như một cái cửa sổ chứa tất cả những gì bạn sẽ vẽ trên đó.

Axes: Thành phần chính của một figure là các axes (những khung nhỏ hơn để vẽ hình lên đó). Một figure có thể chứa một hoặc nhiều axes. Nói cách khác, figure chỉ là khung chứa, chính các axes mới thật sự là nơi các hình vẽ được vẽ lên.

Axis: Chúng là dòng số giống như các đối tượng và đảm nhiệm việc tạo các giới hạn biểu đồ.

Artist: Mọi thứ mà bạn có thể nhìn thấy trên figure là một artist như Text objects, Line2D objects, collection objects. Hầu hết các Artists được gắn với Axes.

**3. Ưu điểm :**

Matplotlib là một thư viện giống như GNUplot. Ưu điểm chính so với GNUplot đó là Matplotlib là một module của Python. Do mức độ phổ biến của python ngày càng tăng, nên matplotlib cũng nhận được sự quan tâm tương tự.

Một lý do khác cho sự hấp dẫn của Matplotlib nằm ở chỗ nó được xem là một sự lựa chọn hoàn hảo thay thế cho MATLAB, nếu nó được sử dụng kết hợp với Numpy và Scipy. Trong khi MATLAB đắt đỏ và mã nguồn đóng, Matplotlib lại miễn phí và mã nguồn mở. Nó cũng là ngôn ngữ hướng đối tượng. Hơn nữa, nó có thể được sử dụng với bộ công cụ GUI mục đích chung như wxPython, Qt, và GTK+. Cũng có một thủ tục "pylab", được thiết kế để giống với MATLAB. Điều này có thể làm cho những người quen sử dụng MATLAB dễ dàng chuyển sang dùng matplotlib.

Matplotlib có thể được sử dụng để tạo ra những figures đủ chất lượng cho một loạt các định dạng hardcopy và môi trường tương tác trên nền tảng.

Một đặc điểm khác của matplotlib là tốc độ lĩnh hội, có nghĩa là người dùng thường đạt được tiến bộ nhanh chóng sau khi bắt đầu. Các trang web chính thức có thể nói những điều sau đây: "matplotlib cố gắng làm những điều khó khăn, phức tạp trở lên dễ dàng nhất có thể. Bạn có thể tạo ra các hình vẽ, histograms, phổ, biểu đồ thanh, errorcharts, scatterplots, vv, với chỉ một vài dòng mã."

Matplotlib có một số interfaces để tương tác với thư viện matplotlib: Object-Oriented API, The Scripting Interface (pyplot), The MATLAB Interface (pylab). Pyplot và pylab đều là lightweight interfaces, tuy nhiên Pyplot cung cấp một giao diện thủ tục các thư viện vẽ hướng đối tượng trong matplotlib. Các lệnh vẽ của nó được thiết kế tương tự với Matlab cả về cách đặt tên và ý nghĩa các đối số. Cách thiết kế này đã giúp cho việc sử dụng pyplot dễ dàng và dễ hiểu hơn vì vậy trong các bài viết về Matplotlib, tôi sẽ sử dụng giao diện pyplot thay vì hai giao diện còn lại. Nếu chúng ta muốn can thiệp sâu hơn, với nhiều tùy chỉnh hơn thì Object-Oriented API sẽ là lựa chọn thích hợp.

### Cài đặt Matplotlib

Để cài đặt Matplotlib nếu bạn có Anaconda chỉ cần gõ conda install matplotlib hoặc sử dụng tools pip

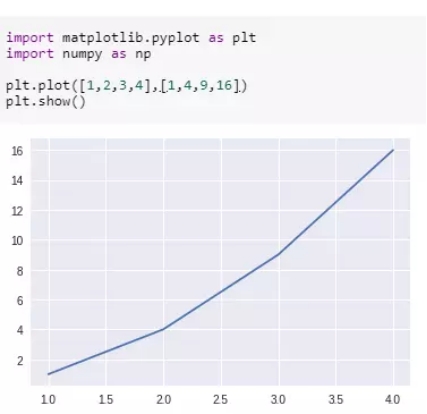
*pip install matplotlib*

### Tạo một biểu đồ đơn giản

*import matplotlib.pyplot as plt*

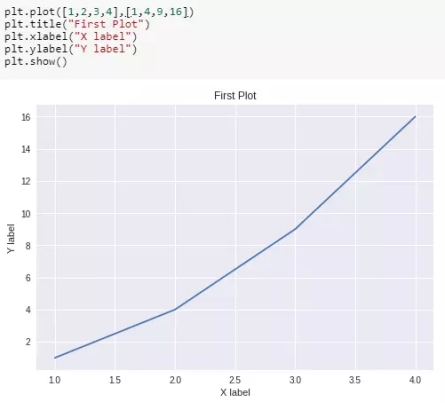
*import numpy as np*

Ở đây chúng ta import Matplotlib’s Pyplot module và thư viện Numpy vì hầu hết các dữ liệu mà ta sẽ làm việc sẽ chỉ ở dạng mảng.



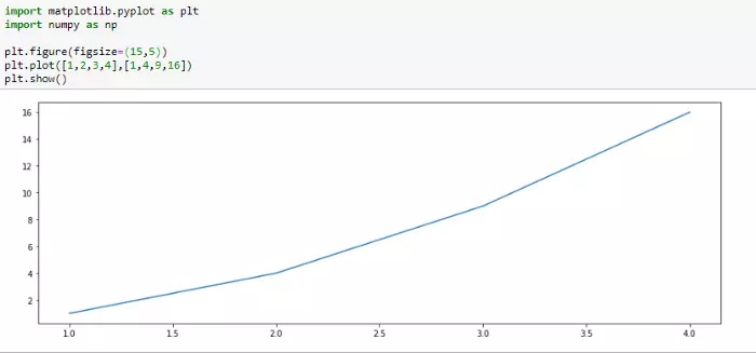
Hình 2.6. Ví dụ tạo biểu đồ đơn giản với Matplotlib

Chúng ta chuyển hai mảng làm đối số đầu vào cho phương thức plot() và sử dụng phương thức show() để gọi biểu đồ được yêu cầu. Ở đây lưu ý rằng mảng đầu tiên xuất hiện trên trục x và mảng thứ hai xuất hiện trên trục y của biểu đồ. Bây giờ, biểu đồ đầu tiên của chúng ta đã sẵn sàng, chúng ta hãy thêm tiêu đề và đặt tên trục x và trục y bằng cách sử dụng các phương thức title(), xlabel() và ylabel().



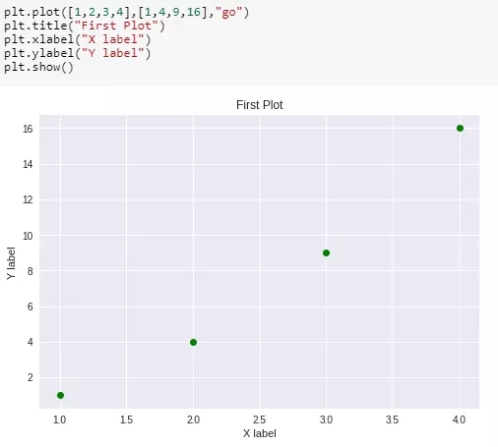
Hình 2.7. Tạo tiêu đề cho các trục biểu đồ

Chúng ta cũng có thể chỉ định kích thước của hình bằng cách sử dụng phương thức figure() và truyền các giá trị dưới dạng một tuple về độ dài của các hàng và cột cho đối số figsize



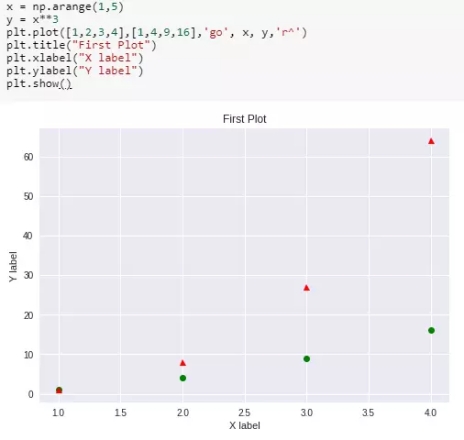
Hình 2.8. Vẽ biểu đồ với kích thước tự chọn

Với mỗi đối số X và Y, bạn cũng có thể chuyển một đối số thứ ba tùy chọn dưới dạng một chuỗi cho biết màu sắc và loại đường của biểu đồ. Định dạng mặc định là b- có nghĩa là một đường màu xanh lam đặc. Trong hình dưới đây, mình sử dụng go có nghĩa là vòng tròn màu xanh lá cây. Tương tự như vậy, chúng ta có thể thực hiện nhiều kết hợp như vậy để định dạng biểu đồ của mình.



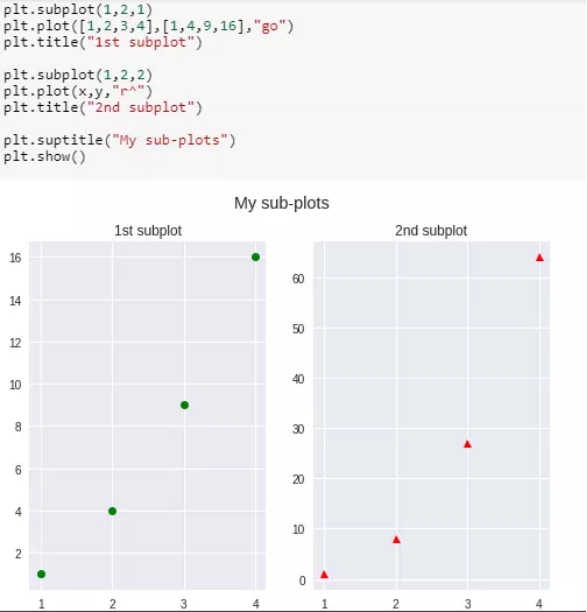
Hình 2.9. Định dạng kí hiệu dữ liệu trên biểu đồ

Chúng ta cũng có thể vẽ nhiều bộ dữ liệu bằng cách chuyển vào nhiều bộ đối số của trục X và Y trong phương thức plot() như bên dưới.



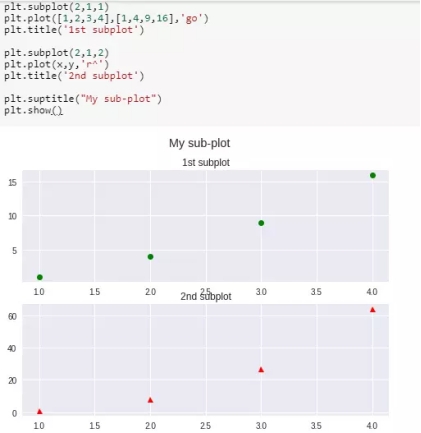
**Nhiều biểu đồ trong 1 figure**

Chúng ta có thể sử dụng phương thức subplot() để thêm nhiều plots trong một hình. Trong hình ảnh bên dưới, mình đã sử dụng phương pháp này để phân tách hai biểu đồ mà đã vẽ trên cùng một trục trong ví dụ trước. Phương thức subplot() có ba đối số: nrows, ncols và index. Chúng chỉ ra số lượng hàng, số cột và số index của sub-plot. Ví dụ, mình muốn tạo hai sub-plot trong một hình sao cho nó nằm trên một hàng và trên hai cột và do đó ta chuyển các đối số (1,2,1) và (1,2,2) trong phương thức subplot(). Lưu ý rằng ta đã sử dụng riêng phương thức title() cho cả các subplots. Ta sử dụng phương thức suptitle() để tạo một tiêu đề tập trung cho hình.



Hình 2.10. Vẽ nhiều biểu đồ với phương thức subplots()

Nếu ta muốn các sub-plots thành hai hàng và một cột, chúng ta có thể truyền các đối số (2,1,1) và (2,1,2)



Cách tạo ra subplots trên đây trở nên hơi tẻ nhạt khi chúng ta muốn có nhiều subplots trong hình. Một cách thuận tiện hơn là sử dụng phương thức subpltots(). Lưu ý sự khác biệt của các giá trị trong cả hai phương thức. Phương thức này lấy hai đối số nrows và ncols làm số lượng hàng và số cột tương ứng. Phương thức này tạo ra hai đối tượng: figure và axes mà chúng ta lưu trữ trong các biến fig và ax có thể được sử dụng để thay đổi các thuộc tính mức figure và axes tương ứng. Lưu ý rằng các tên biến này được chọn tùy ý.

## Seaborn

### Tổng quan về Seaborn

Seaborn là một thư viện trực quan hóa dữ liệu dựa trên matplotlib. Nó cung cấp một giao diện cấp cao để vẽ các biểu đồ thống kê hấp dẫn và nhiều công cụ mạnh mẽ để giúp bạn phân tích và trực quan hóa dữ liệu một cách dễ dàng và hiệu quả.

**Vì sao bạn nên chọn Seaborn?**

Dễ dàng tạo các biểu đồ phức tạp: Seaborn giúp bạn tạo ra các biểu đồ phức tạp chỉ với vài dòng mã.

Các biểu đồ thống kê phong phú: Seaborn cung cấp nhiều loại biểu đồ thống kê như biểu đồ phân phối (distribution plots), biểu đồ quan hệ (relational plots), biểu đồ phân loại (categorical plots) và biểu đồ ma trận (matrix plots).

Tích hợp tốt với Pandas: Seaborn làm việc rất tốt với DataFrame của Pandas, giúp việc thao tác và trực quan hóa dữ liệu trở nên đơn giản.

Kiểu dáng mặc định đẹp mắt: Các biểu đồ của Seaborn có kiểu dáng mặc định rất đẹp mắt và chuyên nghiệp.

Khả năng tùy chỉnh cao: Bạn có thể dễ dàng tùy chỉnh các biểu đồ để phù hợp với nhu cầu của mình.

**Ưu điểm của Seaborn**

Biểu đồ phân phối và thống kê dễ dàng: Bạn có thể dễ dàng tạo các biểu đồ phân phối như histograms, KDE plots, và box plots.

Biểu đồ quan hệ mạnh mẽ: Seaborn cung cấp các công cụ để tạo biểu đồ quan hệ như scatter plots và line plots, giúp bạn hiểu rõ hơn về mối quan hệ giữa các biến.

Các công cụ trực quan hóa phân loại: Bạn có thể sử dụng Seaborn để tạo các biểu đồ phân loại như bar plots và violin plots.

Biểu đồ ma trận: Seaborn cung cấp các công cụ để tạo heatmaps và cluster maps, giúp bạn phân tích dữ liệu ma trận một cách hiệu quả.

Tùy chỉnh giao diện: Seaborn cho phép bạn tùy chỉnh giao diện của biểu đồ một cách dễ dàng, từ màu sắc đến các chi tiết nhỏ như labels và ticks.

Khả năng tích hợp với các thư viện khác: Seaborn tích hợp tốt với các thư viện khác như Pandas và Matplotlib, giúp bạn dễ dàng kết hợp các công cụ để tạo ra các biểu đồ phức tạp.

### Cài đặt Seaborn

Sử dụng pip:

pip install seaborn

Sử dụng Anaconda:

conda install seaborn

### Ví dụ về Seaborn

Tạo biểu đồ phân phối (Distribution Plot)

import seaborn as sns

import matplotlib.pyplot as plt

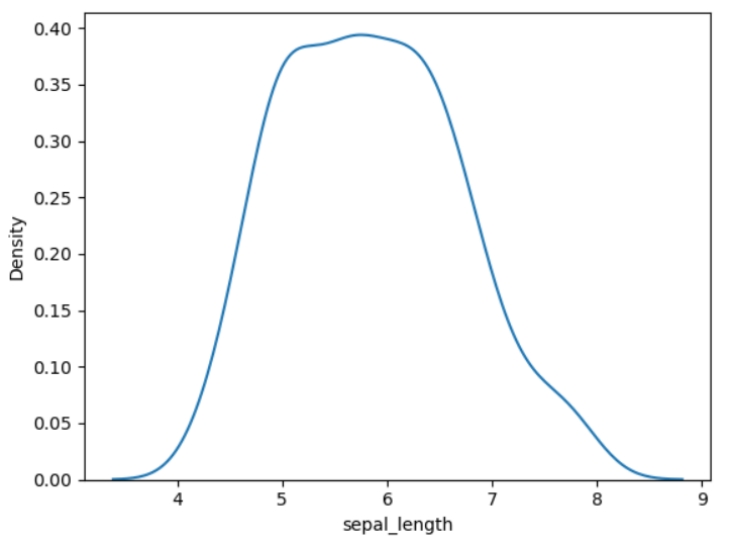
Tạo dữ liệu mẫu

data = sns.load\_dataset('iris')

Vẽ biểu đồ KDE

sns.kdeplot(data['sepal\_length'])

plt.show()



Hình 2.11. Tạo biểu đồ phân phối bằng Seaborn

Tạo biểu đồ quan hệ (Relational Plot)

import seaborn as sns

import matplotlib.pyplot as plt

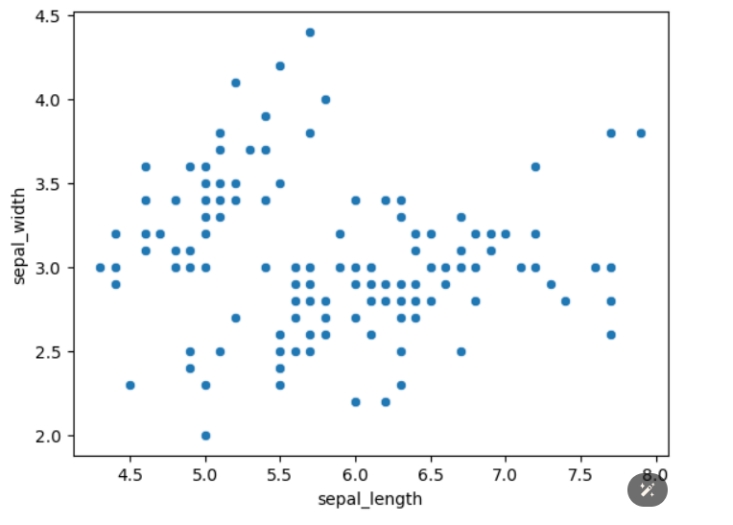
Tạo dữ liệu mẫu

data = sns.load\_dataset('iris')

Vẽ biểu đồ scatter

sns.scatterplot(x='sepal\_length', y='sepal\_width', data=data)

plt.show()



Hình 2.12. Tạo biểu đồ phân tán bằng Seaborn

Tạo biểu đồ phân loại (Categorical Plot)

import seaborn as sns

import matplotlib.pyplot as plt

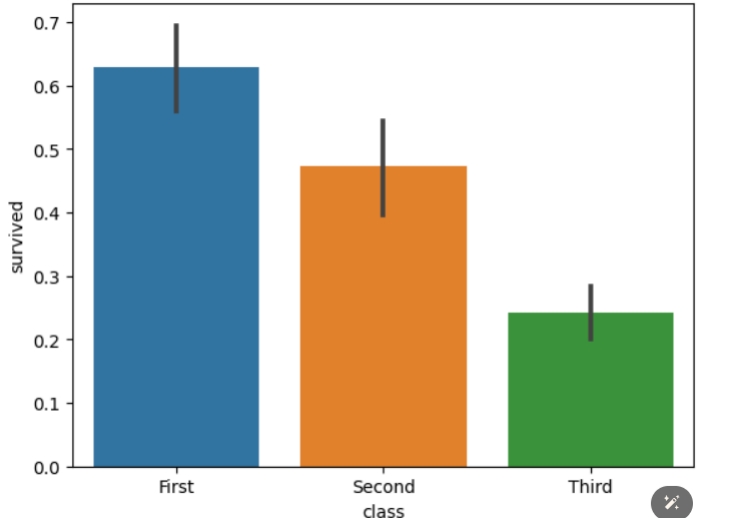
Tạo dữ liệu mẫu

data = sns.load\_dataset('titanic')

Vẽ biểu đồ bar

sns.barplot(x='class', y='survived', data=data)

plt.show()



Hình 2.13. Tạo biểu đồ cột bằng Seaborn

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

Tạo dữ liệu mẫu

data = sns.load\_dataset('flights')

Sắp xếp lại thứ tự tháng theo thứ tự thời gian

month\_order = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

data['month'] = pd.Categorical(data['month'], categories=month\_order, ordered=True)

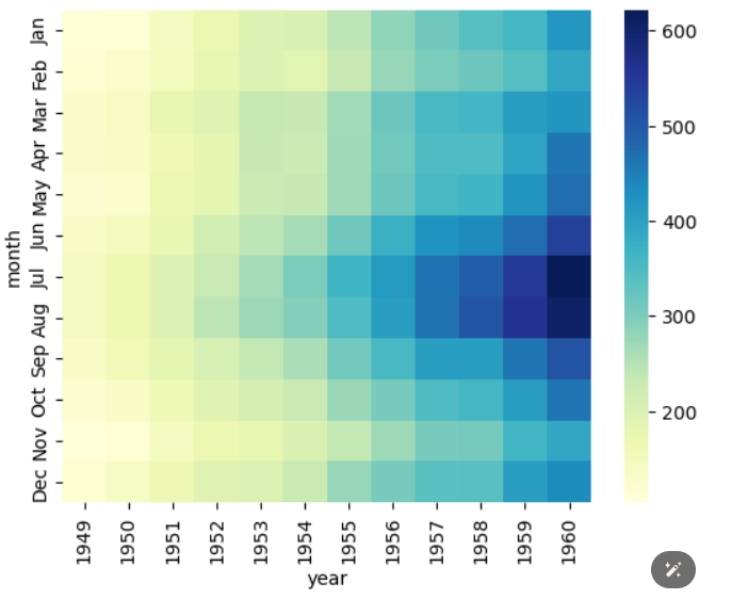
Tạo bảng pivot

flights = data.pivot\_table(values='passengers', index='month', columns='year')

Vẽ heatmap

sns.heatmap(flights, annot=True, fmt='d', cmap='YlGnBu')

plt.show()

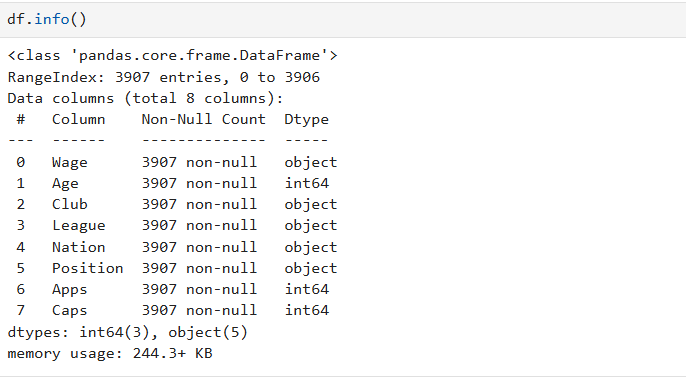


Hình 2.14. Tạo biểu đồ heatmap bằng Seaborn

# GIẢI PHÁP

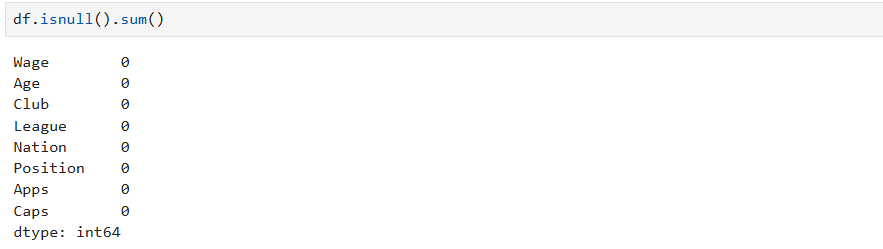
## Mã nguồn tiền xử lý dữ liệu

Đầu tiên em sử dụng hàm df.info() để kiểm tra các feature có kiểu dữ liệu thế nào.Em hiển thị lại hàm df và dùng lại hàm info để kiểm tra xem dữ liệu. Và hình ảnh dưới là kết quả sau khi đã điền các dữ liệu:



Hình 3.1. Thông tin của các feature

Sau đó em sử dụng df.isnull().sum() để kiểm tra xem các feature nào bị trống dữ liệu là bao nhiêu và hiển thị ra màn hình:



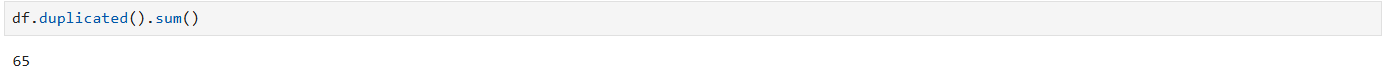
Hình 3.2. Các feature có tổng số bị trống dữ liệu

Chúng ta có thể thấy vì phần mô tả trên cột “Wage” có dạng kiểu dữ liệu là object và chúng ta cần phải đổi sang dạng kiểu int hoặc float, vì “Wage” đang là dạng string và giữa các con số có dấu “,” nên chúng ta sẽ sử dụng phương thức replace() với tham số “regex = True” để thay thế các dấu “,” thành kí tự “ ” và dùng phương thức astype() để chuyển sang kiểu dữ liệu vì dữ liệu của “Wage” khá lớn nên chúng ta sẽ chuyển sang kiểu float



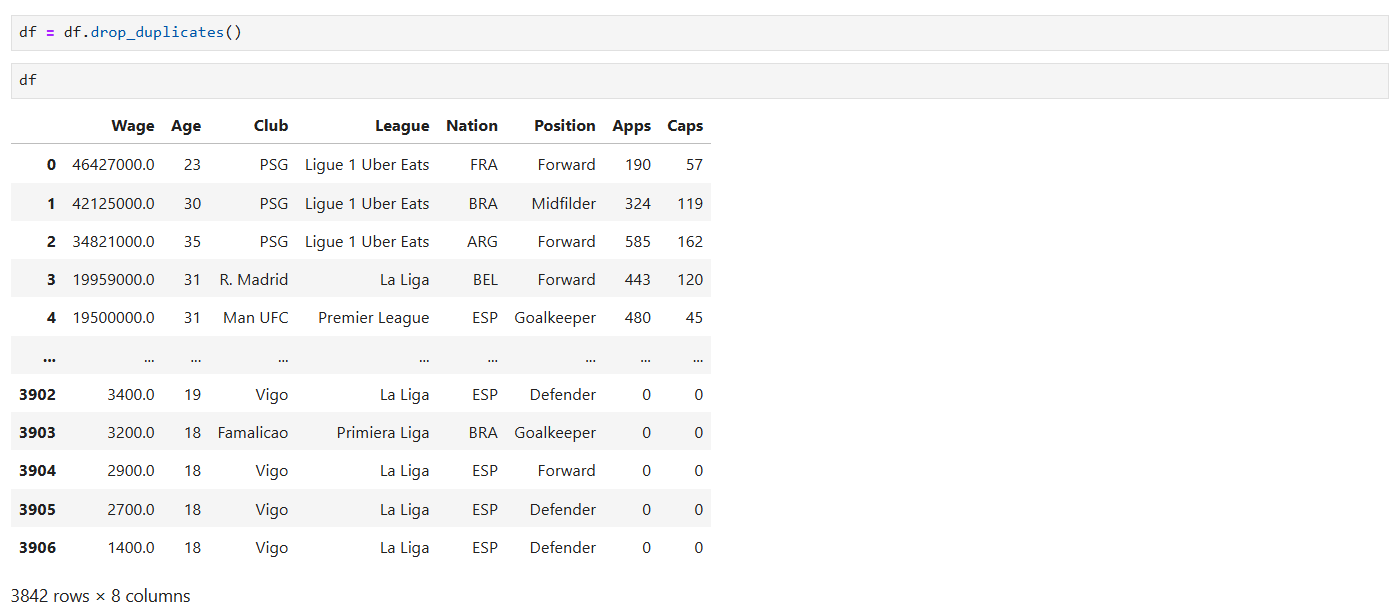
Hình 3.3. Xử lý datatype trong series

Vì có 3907 dữ liệu nên chắc chắn sẽ có vài trường hợp sẽ bị trùng dữ liệu của các feature và chúng ta sẽ sử dụng phương thức duplicated() và sum() để kiểm tra số feature bị trùng dữ liệu với nhau



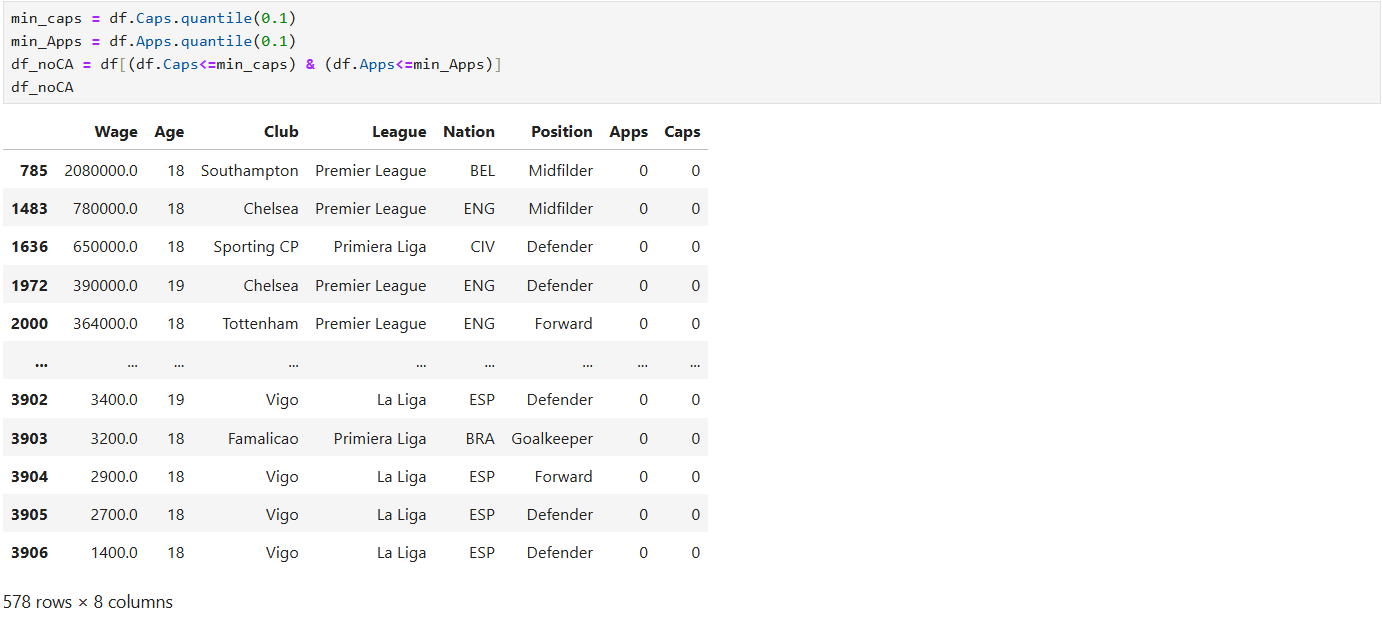
Hình 3.4. Tổng số feature bị trùng lặp dữ liệu

Có thể thấy có tổng 65 dữ liệu trùng lặp nên chúng ta sẽ sử dụng phương thức drop\_duplicates() để xoá các feature bị trùng lặp và sau khi xoá các dữ liệu bị trùng lặp chúng ta còn lại 3842 dữ liệu



Hình 3.5. Dữ liệu sau khi xoá các dữ liệu bị trùng

Chúng ta có thể thấy giữa 2 cột “Apps” và “Caps” đều có dữ liệu là 0 chúng ta sẽ cho các dữ liệu này trở thành 1 DataFrame dữ liệu của “Apps” và “Caps” đều là 0 bằng cách sử dụng phương thức quantile() và tham số lượng tử là 0.1, chúng ta sẽ có bảng dữ liệu như sau

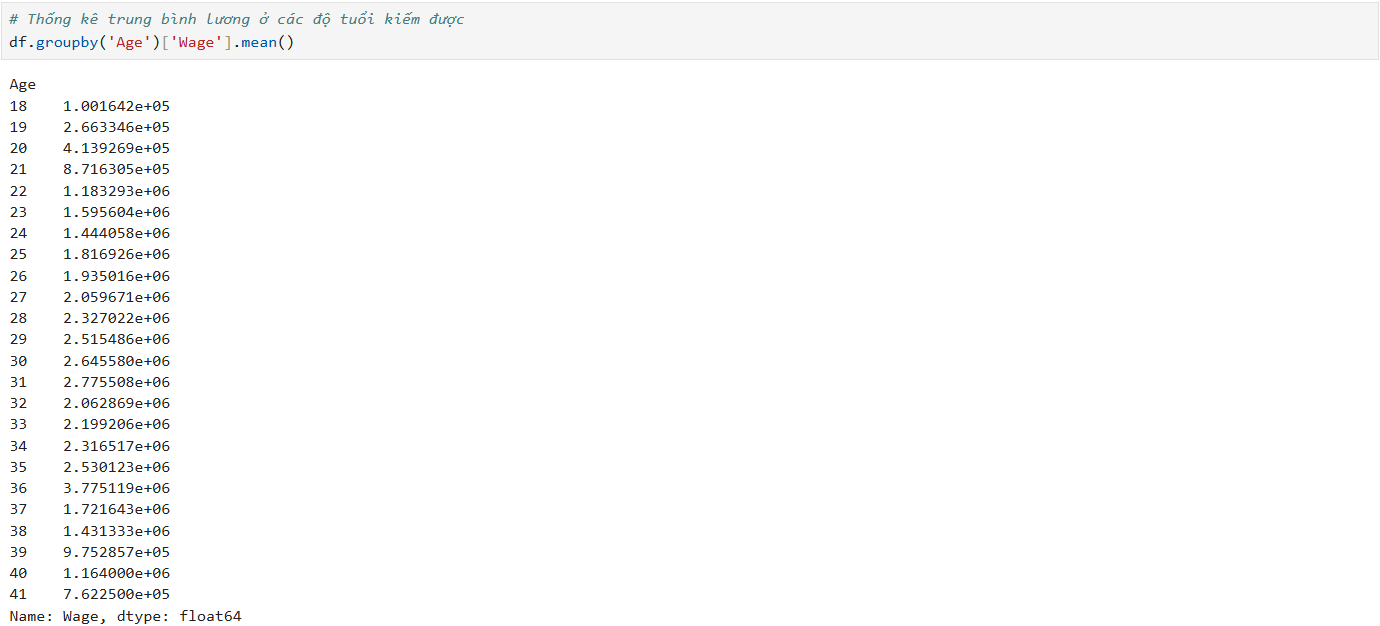


Hình 3.6. Bảng dữ liệu của “Apps” và “Caps” đều có giá trị 0

## Mã nguồn chức năng Thống kê dữ liệu

### Thống kê mức lương trung bình của các cầu thủ qua các độ tuổi

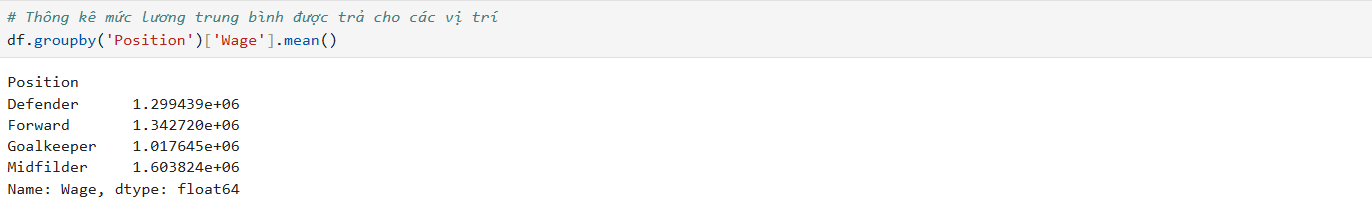
Để thống kê mức lương trung bình của các độ tuổi, chúng ta sẽ cần phải tính giá trị mức lương trung bình của các độ tuổi đó. Chúng ta sẽ sử dụng phương thức groupby() và phương thức mean(). Phương thức groupby() cho phép chúng ta nhóm các dữ liệu giống nhau theo nhãn feature mà ta chọn trong DataFrame, vì ở đây là mức lương trung bình của các độ tuổi nên chúng ta sẽ nhóm dữ liệu ở cột “Age” và mean() được sử dụng để tính giá trị trung bình lương cầu thủ của từng độ tuổi “Wage”



Hình 3.7. Mã nguồn thống kê mức lương trung bình của các cầu thủ qua các độ tuổi

### Thống kê mức lương trung bình của các cầu thủ cho các vị trí trong Club

Để thống kê mức lương trung bình của các vị trí, chúng ta sẽ cần phải tính giá trị mức lương trung bình của các vị trí đó. Chúng ta sẽ sử dụng phương thức groupby() và phương thức mean(). Phương thức groupby() cho phép chúng ta nhóm các dữ liệu giống nhau theo nhãn feature mà ta chọn trong DataFrame, vì ở đây là mức lương trung bình của các vị trí nên chúng ta sẽ nhóm dữ liệu ở cột “Position” và mean() được sử dụng để tính giá trị trung bình lương cầu thủ của từng vị trí “Wage”



Hình 3.8. Mã nguồn thống kê mức lương trung bình của các cầu thủ cho các vị trí

### Thống kê trung bình mức lương được nhận của các cầu thủ theo các châu lục khác nhau

Chúng ta có thể thấy khi đọc file dữ liệu trong DataFrame không có cột các châu lục cho các nước vì vậy chúng ta sẽ phải gán các quốc gia thuộc các châu lục khác nhau và tạo 1 cột “Continent” để lữu trữ các châu lục theo từng các quốc gia khác nhau. Để có thể gán dữ liệu các châu lục tương ứng với các quốc gia chúng ta cần 1 dict, trong dict là các quốc gia được gán theo các châu lục tương ứng và sử dụng phương thức map() để thay thế các quốc gia thành châu lục nhưng lại lưu các giá trị được thay vào “Continent”

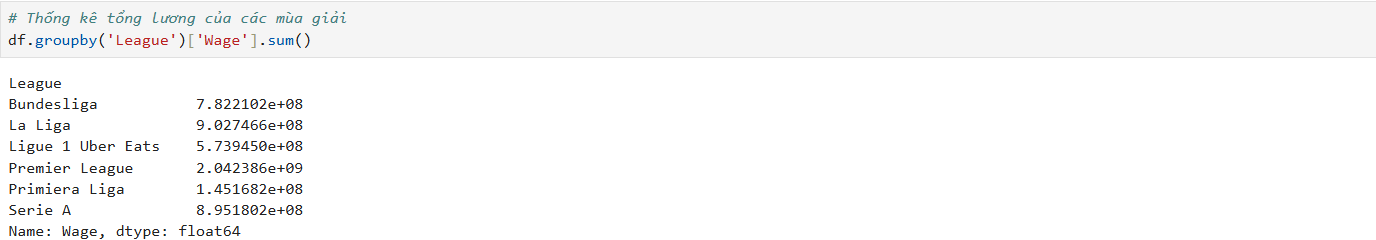
Sau khi có các dữ liệu của các châu lục chúng ta sẽ cần phải tính giá trị mức lương trung bình của các châu lục đó. Chúng ta sẽ sử dụng phương thức groupby() và phương thức mean(). Phương thức groupby() cho phép chúng ta nhóm các dữ liệu giống nhau theo nhãn feature mà ta chọn trong DataFrame, vì ở đây là mức lương trung bình của các châu lục nên chúng ta sẽ nhóm dữ liệu ở cột “Continent” và mean() được sử dụng để tính giá trị trung bình lương cầu thủ của từng châu lục “Wage”



Hình 3.9. Mã nguồn thống kê trung bình mức lương nhận được của các châu lục

### Thống kê tổng lương được trả qua các giải đấu

Để thống kê tổng mức lương được trả qua các giải đấu, chúng ta sẽ cần phải tính tổng mức lương của các giải đấu đó. Chúng ta sẽ sử dụng phương thức groupby() và phương thức sum(). Phương thức groupby() cho phép chúng ta nhóm các dữ liệu giống nhau theo nhãn feature mà ta chọn trong DataFrame, vì ở đây là tổng mức lương của các mùa giải nên chúng ta sẽ nhóm dữ liệu ở cột “League” và sum() được sử dụng để tính tổng mức lương được trả cho các giải đấu “Wage”



Hình 3.10. Mã nguồn thống kê tổng lương được trả qua các giải đấu

### Thống kê tổng lương của các cầu thủ trên và dưới 30 tuổi

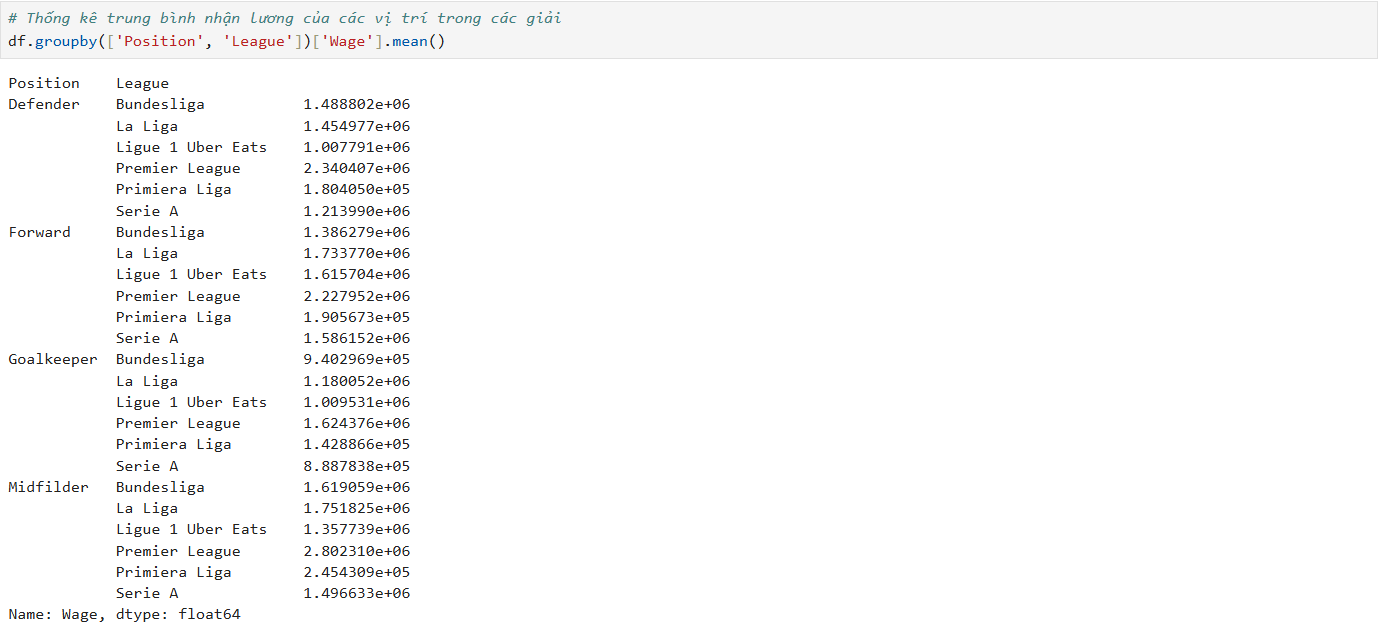
Để thống kê tổng lương của các cầu thủ trên và dưới 30 tuổi, việc đầu tiên chúng ta cần lấy dữ liệu “Age” trên dưới 30 bằng cách chọn hàng sử dụng phương thức loc[] và lưu giá trị các hàng vào các biến “tren\_30” và ”duoi\_30” , kết hợp sử dụng phương thức sum() để tính tổng tiền lương mà nhận được dựa trên cột “Wage”



Hình 3.11. Mã nguồn thống kê tổng lương nhận được trên và dưới 30 tuổi

### Thống kê trung bình mức lương nhận được của các vị trí ở trong các giải

Để thống kê mức lương trung bình của các vị trí ở trong các giải bóng, chúng ta sẽ cần phải tính giá trị mức lương trung bình của các vị trí và ở trong các giải đấu đó. Chúng ta sẽ sử dụng phương thức groupby() và phương thức mean(). Phương thức groupby() cho phép chúng ta nhóm các dữ liệu giống nhau theo nhãn feature mà ta chọn trong DataFrame, vì ở đây là mức lương trung bình của các vị trí nên chúng ta sẽ nhóm dữ liệu ở cột “Position, League” và mean() được sử dụng để tính giá trị trung bình lương cầu thủ của từng vị trí ở các giải đấu khác nhau“Wage”



Hình 3.12. Mã nguồn thống kê trung bình mức lương của các vị trí trong các giải đấu

### Thống kê tổng lương phải trả nhiều nhất của các câu lạc bộ trong top 10

Để thống kê tổng lương phải trả nhiều nhất của các câu lạc bộ trong top 10 chúng ta cần tính tổng lương cho các câu lạc bộ bằng cách sử dụng phương thức groupby() và phương thức sum(). Phương thức groupby() dùng để nhóm các dữ liệu giống nhau vì vậy ở đây chúng ta sẽ nhóm “Club” và sử dụng sum() để tính tổng mức lương cho các câu lạc bộ. Kết hợp sử dụng phương thức rename\_axis() có chuỗi “Club” chuyền vào để thay đổi tên feature thành “Club” và phương thức reset\_index() để trở thành DataFrame chứa 2 cột “Club” và “Wage”, truyền tham số name =”Sum Age” để đổi tên cột dữ liệu “Wage”

Sau đó chúng ta sử dụng phương thức sort\_values() để sắp xếp các dữ liệu và truyền tham số by = “Sum Wage” để sắp xếp dữ liệu ở cột đó và tham số ascending = Fasle để được sắp xếp dữ liệu từ cao đến thấp kết hợp phương thức head() có giá trị là 10 để in ra tổng lương trả nhiều nhất của các câu lạc bộ trong top 10



Hình 3.13. Mã nguồn thống kê top 10 Club trả lương cầu thủ nhiều nhất

### Thống kê tổng lương phải trả nhiều nhất cho độ tuổi và vị trí của các câu lạc bộ trong top 15

Để Thống kê tổng lương phải trả nhiều nhất cho độ tuổi và vị trí của các câu lạc bộ bằng cách sử dụng phương thức groupby() và phương thức sum(). Phương thức groupby() dùng để nhóm các dữ liệu giống nhau vì vậy ở đây chúng ta sẽ nhóm ['Club', 'Age', 'Position'] và sử dụng sum() để tính tổng mức lương cho các câu lạc bộ. Kết hợp sử dụng phương thức rename\_axis() có chuỗi ['Club', 'Age', 'Position'] chuyền vào để thay đổi tên feature thành ['Club', 'Age', 'Position'] và phương thức reset\_index() để trở thành DataFrame chứa các cột ['Club', 'Age', 'Position'] và “Wage”, truyền tham số name = ”Sum Age” để đổi tên cột dữ liệu “Wage”

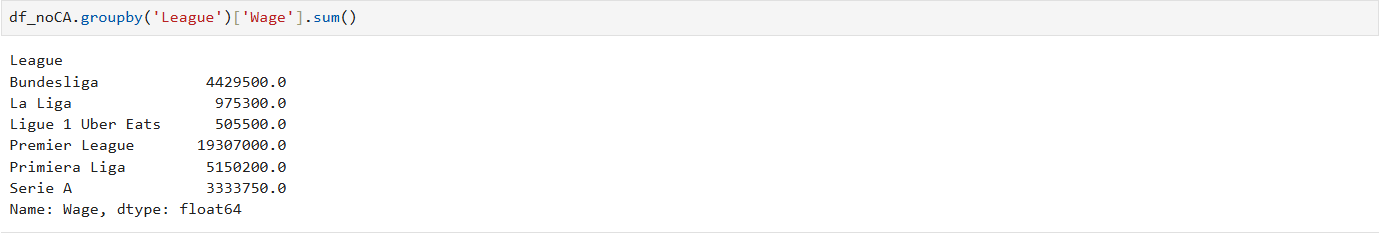
Sau đó chúng ta sử dụng phương thức sort\_values() để sắp xếp các dữ liệu và truyền tham số by = “Sum Wage” để sắp xếp dữ liệu ở cột đó và tham số ascending = Fasle để được sắp xếp dữ liệu từ cao đến thấp kết hợp phương thức head() có giá trị là 15 để in ra tổng lương trả nhiều nhất của các câu lạc bộ trong top 15



Hình 3.14.Thống kê tổng lương phải trả nhiều nhất cho độ tuổi và vị trí của các Club trong top 15

### Thống kê tổng lương các cầu thủ không được sử dụng trong club và quốc gia của các giải đấu

Để thống kê tổng mức lương được trả qua các giải đấu, chúng ta sẽ cần phải tính tổng mức lương của các giải đấu đó. Chúng ta sẽ sử dụng phương thức groupby() và phương thức sum(). Phương thức groupby() cho phép chúng ta nhóm các dữ liệu giống nhau theo nhãn feature mà ta chọn trong DataFrame, vì ở đây là tổng mức lương của các mùa giải nên chúng ta sẽ nhóm dữ liệu ở cột “League” và sum() được sử dụng để tính tổng mức lương được trả cho các giải đấu “Wage”



Hình 3.15. Mã nguồn thống kê tổng lương các cầu thủ không được sử dụng trong club và quốc gia của các giải đấu

### Thống kê top 10 tổng lương phải trả nhiều nhất của các cầu thủ không được sử dụng trong Club và quốc gia của các câu lạc bộ

Để thống kê top 10 tổng lương phải trả nhiều nhất của các cầu thủ không được sử dụng trong Club và quốc gia của các câu lạc bộ chúng ta cần tính tổng lương cho các câu lạc bộ bằng cách sử dụng phương thức groupby() và phương thức sum(). Phương thức groupby() dùng để nhóm các dữ liệu giống nhau vì vậy ở đây chúng ta sẽ nhóm “Club” và sử dụng sum() để tính tổng mức lương cho các câu lạc bộ. Kết hợp sử dụng phương thức rename\_axis() có chuỗi “Club” chuyền vào để thay đổi tên feature thành “Club” và phương thức reset\_index() để trở thành DataFrame chứa 2 cột “Club” và “Wage”, truyền tham số name =”NoCa\_Sum Wage” để đổi tên cột dữ liệu “Wage”

Sau đó chúng ta sử dụng phương thức sort\_values() để sắp xếp các dữ liệu và truyền tham số by = “NoCa\_Sum Wage” để sắp xếp dữ liệu ở cột đó và tham số ascending = Fasle để được sắp xếp dữ liệu từ cao đến thấp kết hợp phương thức head() có giá trị là 10 để in ra tổng lương phải trả nhiều nhất của các cầu thủ không được sử dụng trong Club và quốc gia của các câu lạc bộ chúng ta cần tính tổng lương cho các câu lạc bộ trong top 10



Hình 3.16. Mã nguồn hống kê top 10 tổng lương phải trả nhiều nhất của các cầu thủ không được sử dụng trong Club và quốc gia của các câu lạc bộ

## Mã nguồn Trực quan hoá dữ liệu

Đầu tiên, chúng ta cần import một vài thư viện hỗ trợ cho trực quan hoá dữ liệu như Matplotlib, Seaborn:



Hình 3.17. Thư viện hỗ trợ cho trực quan hoá dữ liệu

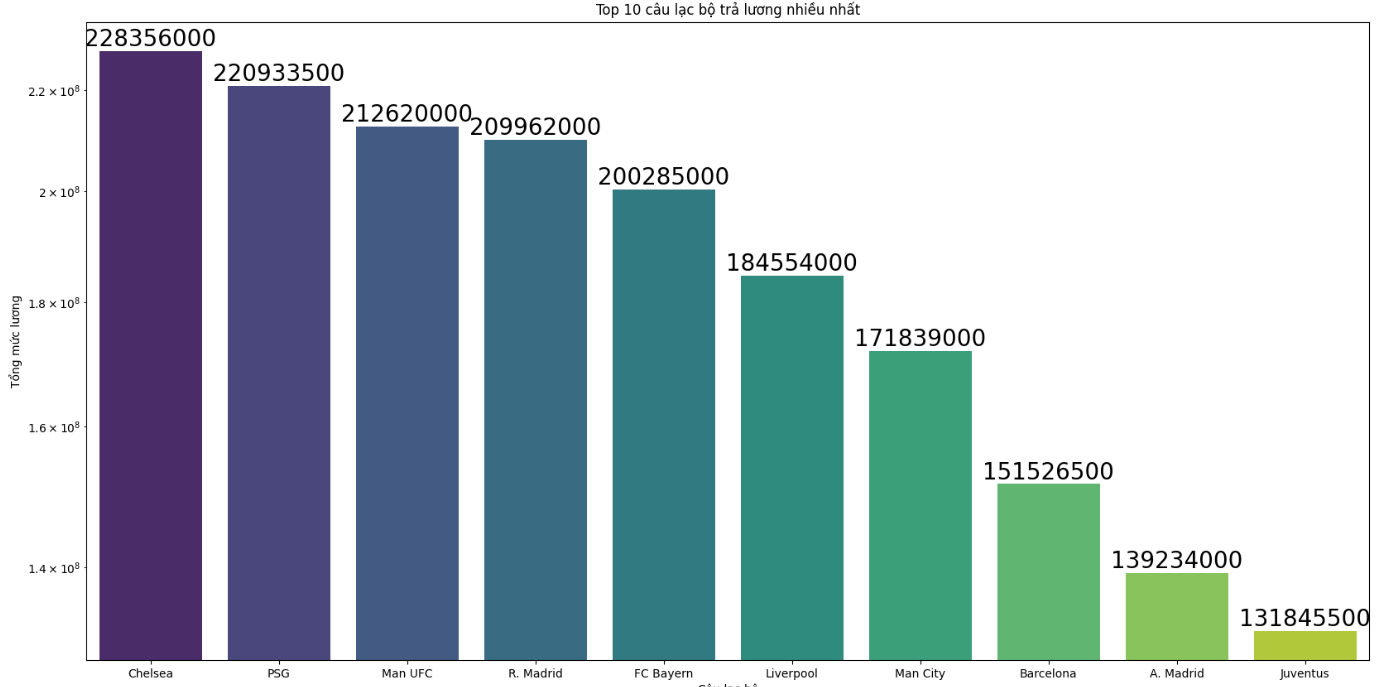
### Mã nguồn, biểu đồ thể hiện top 10 câu lạc bộ trả lương nhiều nhất



Hình 3.18. Mã nguồn thể hiện top 10 câu lạc bộ trả lương nhiều nhất

Do biểu đồ cột chứa các con số dữ liệu tổng lương phải trả của các đội bóng khá lớn nên chúng ta cần chỉnh lại kích thước figure bằng câu lệnh plt.figure(figsize = (20,10)) để đảm bảo đủ không gian cho các dữ liệu được hiển thị trên biểu đồ. Câu lệnh sns.barplot(x = “Club”, y = “Sum Wage”, data = df\_club\_wage) được sử dụng để vẽ biểu đồ cột từ data là df\_club\_wage, trục x sẽ hiển thị tên các dội bóng và trục y hiển thị tổng mức lương các đội bóng phải trả cho các cầu thủ tương ứng với mỗi đội bóng. Chúng ta gán tên trục x, y và đặt tên cho biểu đồ đó lần lượt các phương thức set\_xlabel(), set\_ylabel(), set\_title() và kết hợp sử dụng set\_yscale() với tham số là “log” để trực quan hoá dữ liệu nhìn rõ các cột hơn

Để dễ dàng quan sát về các giá trị tổng mức lương phải trả của các đội bóng chúng ta sử dụng vòng lặp for để duyệt qua từng cột của biểu đồ, câu lệnh ax.annotate(format(p.get\_height(), '.0f'), (p.get\_x() + p.get\_width() / 2., p.get\_height()), ha = 'center', va = 'center', xytext = (0, 10), textcoords = 'offset points', fontsize=20) để chú thích hiển thị giá trị của cột lên đỉnh toạ độ tương ứng. Cụ thể format(p.get\_height(), '.0f'): Lấy chiều cao của cột (tức là giá trị y) và định dạng nó thành số nguyên (không có số thập phân); (p.get\_x() + p.get\_width() / 2., p.get\_height()): Đây là tọa độ nơi chú thích sẽ được đặt. p.get\_x() + p.get\_width() / 2. tính toán vị trí x ở giữa cột, và p.get\_height() đặt vị trí y ở đỉnh cột; ha = 'center', va = 'center': Các tham số này đặt căn chỉnh ngang (ha) và căn chỉnh dọc (va) của chú thích. Trong trường hợp này, chú thích sẽ được căn giữa cả ngang và dọc; xytext = (0, 10): Đặt vị trí của chú thích tương đối so với điểm được chú thích. Trong trường hợp này, chú thích sẽ được đặt 10 điểm phía trên điểm được chú thích; textcoords = 'offset points': Đặt hệ thống tọa độ cho xytext. Trong trường hợp này, xytext sẽ được hiểu là một offset trong điểm từ vị trí được chú thích.



Hình 3.19. Biểu đồ thể hiện top 10 câu lạc bộ trả lương nhiều nhất

### Mã nguồn, biểu đồ thể hiện mức lương trung bình cho từng vị trí



Hình 3.20. Mã nguồn thể hiện mức lương trung bình cho từng vị trí

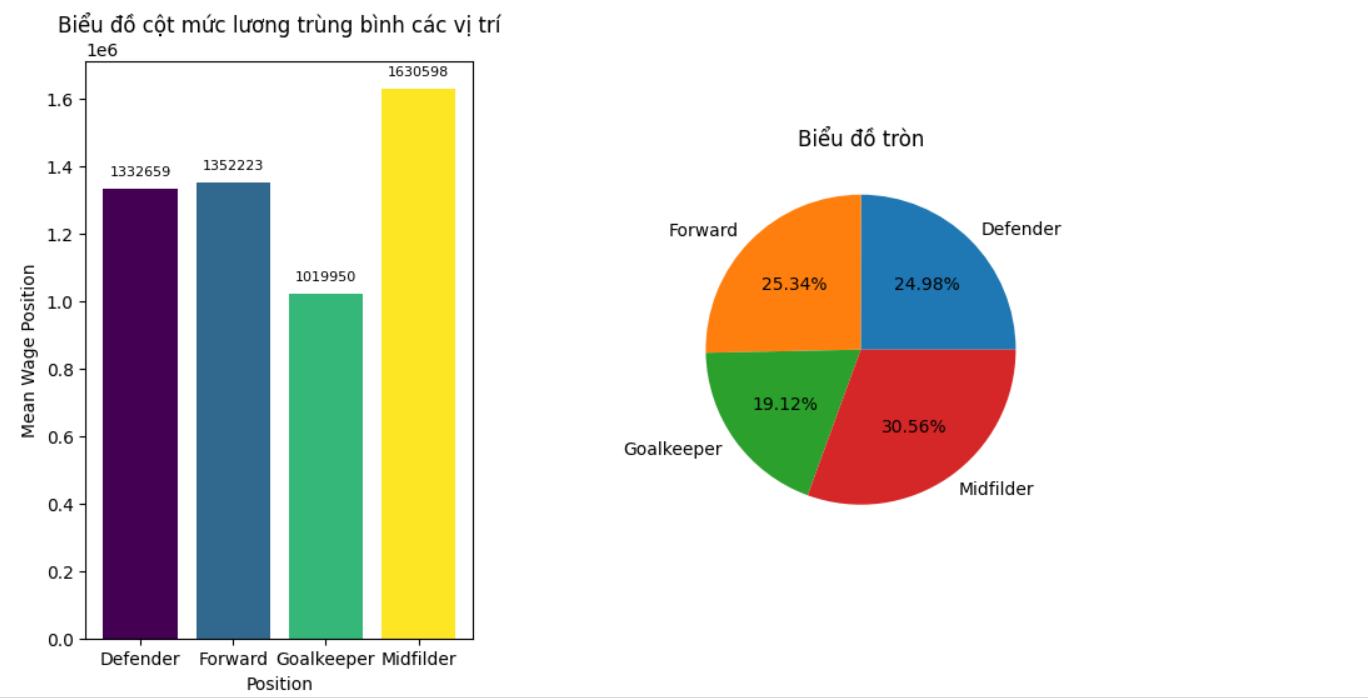
Do thể hiện này chúng ta vẽ 2 biểu đồ nên chúng ta sẽ dụng phương thức subplot() với tham số chuyền vào tương ứng là figure gồm 1 hàng và 2 cột với size của figure là (10,6) để đảm bảo đủ không gian cho 2 biểu đồ. Sau đó chúng ta sử dụng phương thức subplots\_adjust() truyền tham số wspace = 0.5 có nghĩa là khoảng các giữa các biểu đồ là 0.5.

Sau đó sử dụng axes[0] với tương ứng vị trí của figure là 0: axes[0].bar(df\_mean\_position['Position'], df\_mean\_position['Mean Wage']) tương ứng với trục x là các vị trí của cầu thủ và trục y là trung bình lương của các vị trí đó.

Để dễ dàng quan sát biểu đồ hơn chúng ta cần hiển thị giá trị trung bình lương tương ứng với các vị trí đó bằng cách sử dụng vòng lặp for để duyệt qua từng cột của biểu đồ, câu lệnh ax.annotate(format(p.get\_height(),'.0f'),(p.get\_x()+p.get\_width() / 2., p.get\_height()), ha = 'center', va = 'center', xytext = (0, 10), textcoords = 'offset points', fontsize=20) để chú thích hiển thị giá trị của cột lên đỉnh toạ độ tương ứng. Cụ thể format(p.get\_height(), '.0f'): Lấy chiều cao của cột (tức là giá trị y) và định dạng nó thành số nguyên (không có số thập phân); (p.get\_x() + p.get\_width() / 2., p.get\_height()): Đây là tọa độ nơi chú thích sẽ được đặt. p.get\_x() + p.get\_width() / 2. tính toán vị trí x ở giữa cột, và p.get\_height() đặt vị trí y ở đỉnh cột; ha = 'center', va = 'center': Các tham số này đặt căn chỉnh ngang (ha) và căn chỉnh dọc (va) của chú thích. Trong trường hợp này, chú thích sẽ được căn giữa cả ngang và dọc; xytext = (0, 10): Đặt vị trí của chú thích tương đối so với điểm được chú thích. Trong trường hợp này, chú thích sẽ được đặt 10 điểm phía trên điểm được chú thích; textcoords = 'offset points': Đặt hệ thống tọa độ cho xytext. Trong trường hợp này, xytext sẽ được hiểu là một offset trong điểm từ vị trí được chú thích. Sử dụng axes[0] tức là biểu đồ ở giá trị figure là 0 sử dụng lần lượt các phương thức để tạo tiêu đề cho trục x, y và tiêu đề của biểu đồ: axes[0].set\_xlabel(), axes[0].set\_ylabel(), axes[0].set\_title().

Tiếp theo sử dụng axes[1] tương ứng với figure có vị trí là 1: axes[1].pie() để vẽ biểu đồ tròn,câu lệnh:

axes[1].pie(df\_mean\_position['MeanWage'],labels=df\_mean\_position['Position'], autopct='%0.2f%%') tương ứng các giá trị là trung bình lương của các vị trí và có các labels tương giá trị trung bình mức lương của các vị trí, autopct='%0.2f%%' lấy giá trị phần trăm của từng phần làm tròn đến số thập phân thứ 2, axes[1].set\_title() để ghi tên của biểu đồ tương ứng với vị trí figure là 1



Hình 3.21. Biểu đồ thể hiện mức lương trung bình cho từng vị trí

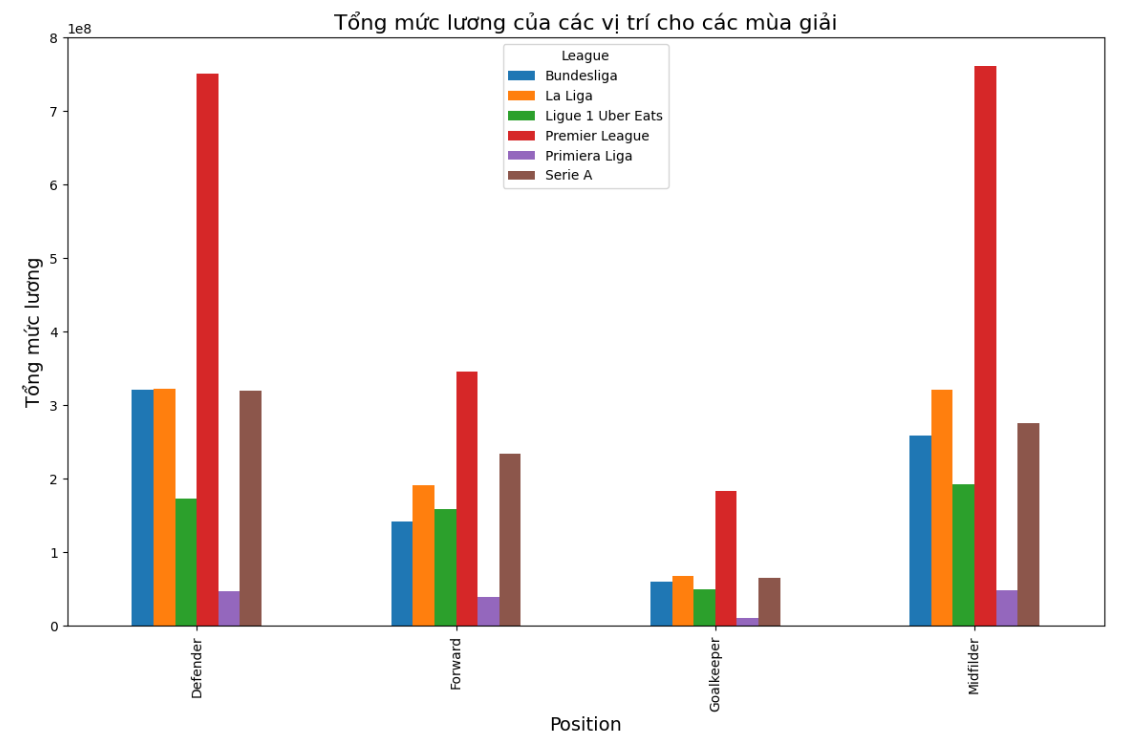
### Mã nguồn, biểu đồ thể hiện tổng mức lương cho các vị trí trong các giải đấu



Hình 3.22. Mã nguồn thể hiện tổng mức lương cho các vị trí trong các giải đấu

Đầu tiên chúng ta sử dụng phương thức pivot() để chuyển đổi các dữ liệu với các tham số tương ứng: index = “position”, columns = “League”, values = “Wage” có nghĩa index ban đầu của DataFrame trở thành index dữ liệu các vị trí, đổi các cột trở thành cột dữ liệu là league và values ban đầu trở thành các values chỉ có wage

Tiếp theo sử dụng phương thức subplots để định dạng kích thước của biểu đồ, câu lệnh df\_sum\_wage\_positon\_league.plot(kind = “bar”, ax = ax) có ý nghĩa DataFrame được vẽ dưới dạng biểu đồ cột thông qua tham số kind và vẽ với tương ứng ax. Sử dụng các phương thức lần lượt title(), xlabel(), ylabel(), lengend(), show() tương ứng nghĩa: đặt tên cho biểu đồ, đặt tên trục x, đặt tên trục y, chú thích các dữ liệu, hiển thị biểu đồ đó.



Hình 3.23. Biểu đồ thể hiện tổng mức lương cho các vị trí trong các giải đấu

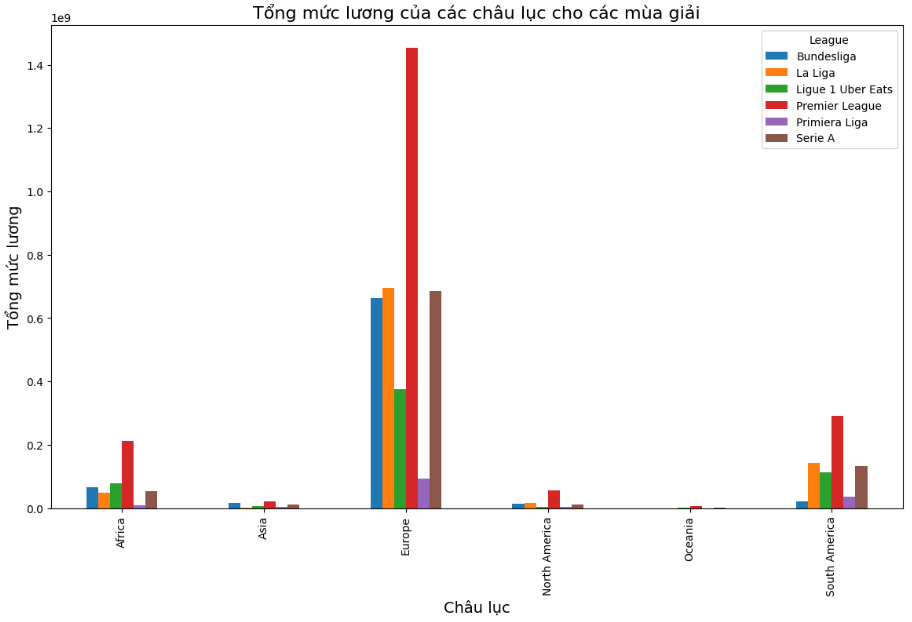
### Mã nguồn, biểu đồ thể hiện tổng mức lương của các châu lục trong các giải đấu



Hình 3.24. Mã nguồn thể hiện tổng mức lương của các châu lục trong các giải đấu

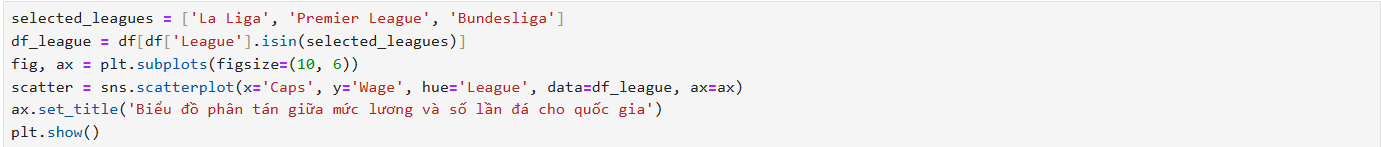
Đầu tiên chúng ta sử dụng phương thức pivot() để chuyển đổi các dữ liệu với các tham số tương ứng: index = “Continent”, columns = “League”, values = “Wage” có nghĩa index ban đầu của DataFrame trở thành index dữ liệu các vị trí, đổi các cột trở thành cột dữ liệu là league và values ban đầu trở thành các values chỉ có wage

Tiếp theo sử dụng phương thức subplots để định dạng kích thước của biểu đồ, câu lệnh df\_sum\_wage\_positon\_league.plot(kind = “bar”, ax = ax) có ý nghĩa DataFrame được vẽ dưới dạng biểu đồ cột thông qua tham số kind và vẽ với tương ứng ax. Sử dụng các phương thức lần lượt title(), xlabel(), ylabel(), lengend(), show() tương ứng nghĩa: đặt tên cho biểu đồ, đặt tên trục x, đặt tên trục y, chú thích các dữ liệu, hiển thị biểu đồ đó.



Hình 3.25. Biểu đồ thể hiện tổng mức lương của các châu lục trong các giải đấu

### Mã nguồn, biểu đồ thể hiện mức lương nhận được dựa trên số lần đá cho quốc gia trong các giải đấu

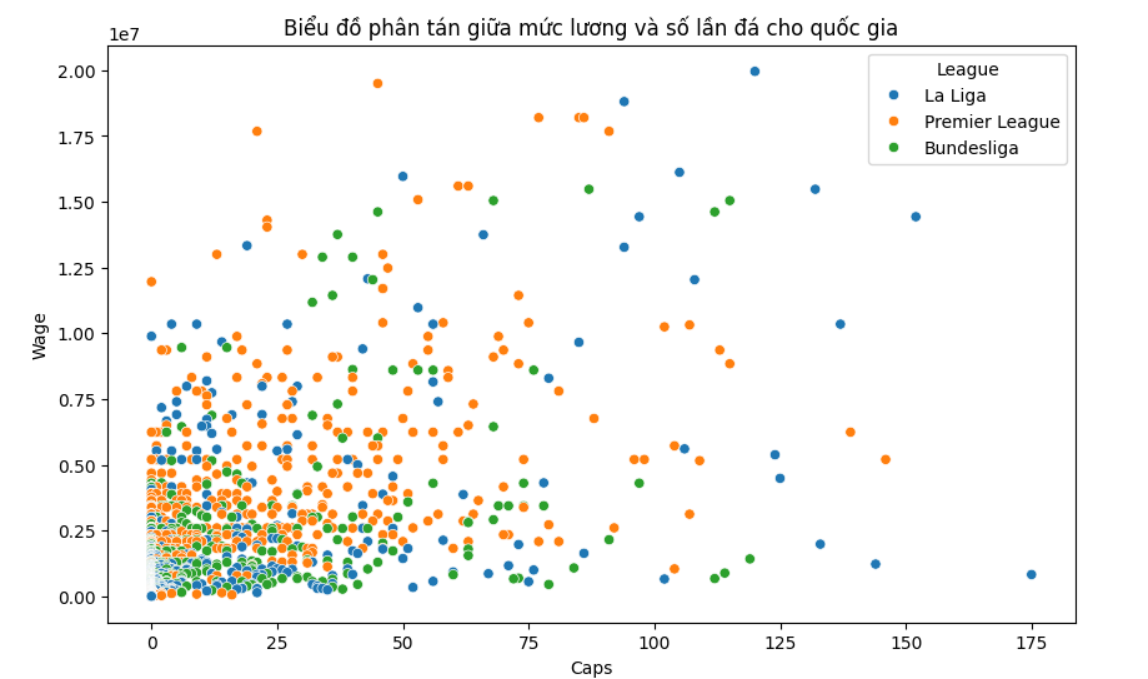


Hình 3.26. Mã nguồn thể hiện mức lương nhận được dựa trên số lần đá cho quốc gia trong các giải đấu

Chúng ta cần chọn các giải đấu mà chúng ta muốn trực quan hoá quan hệ giữa mức lương và số lần đá cho quốc gia của cầu thủ đó vào 1 list. Sử dụng kết hợp phương thức isin() để kiểm tra xem các phần tử trong list có tồn tại trong cột giải đấu không nếu có nó sẽ trả cho bản ghi là True

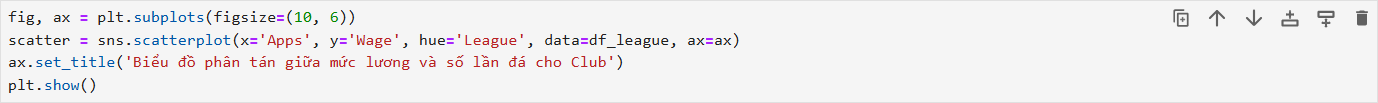
Sau đó chúng ta sử dụng phương thức subplots() để tạo các ô con của figure và có tham số kích thước của figure là (10,6) có chiều rộng là 10 và chiều dài là 6.

Sử dụng phương thức scatterplot() để vẽ biểu đồ phân tán với câu lệnh sns.scatterplot(x = “Caps”, y = “Wage”,hue =”League”, data = df\_league, ax = ax) có nghĩa tương ứng lần lượt như sau: trục x là các dữ liệu của Caps, trục y là các dữ liệu của wage, tham số hue là tham số phân biệt các điểm dữ liệu trên 1 biến cụ thể và dữ liệu được lấy từ df\_league, ax được vẽ tại vị trí chính nó.



Hình 3.27.Biểu đồ thể hiện thể hiện mức lương nhận được dựa trên số lần đá cho quốc gia trong các giải đấu

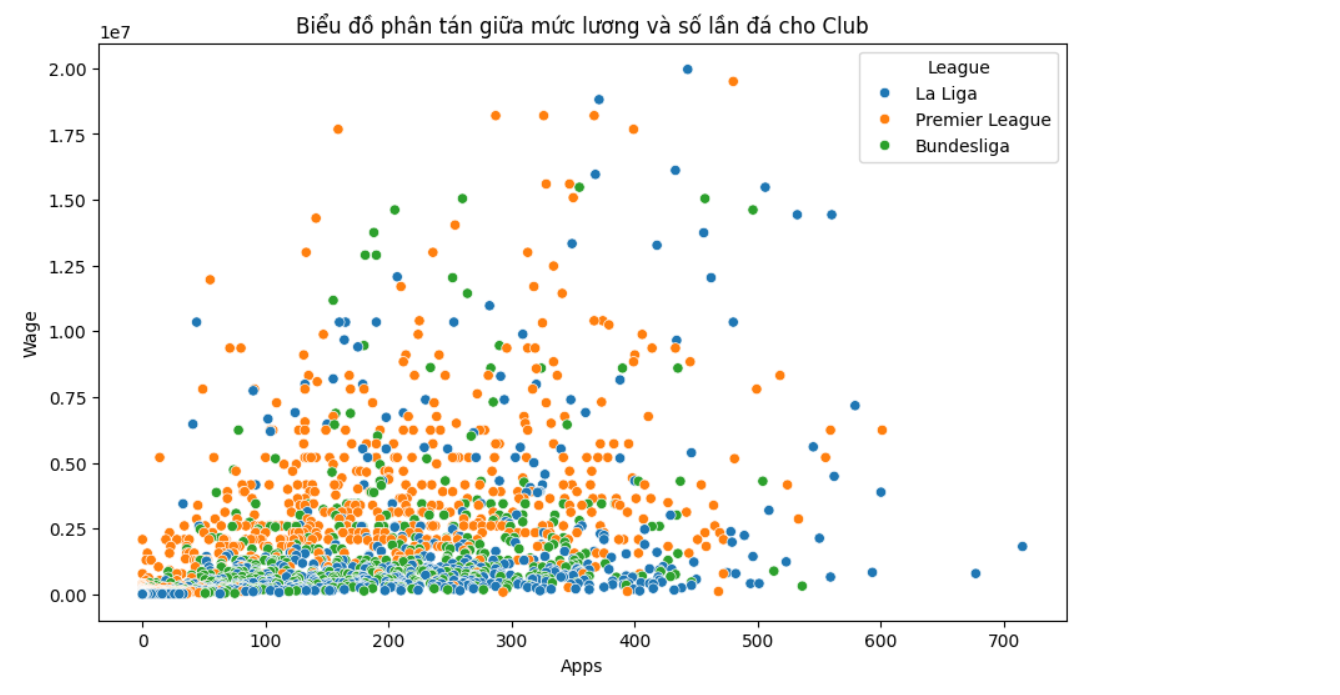
### Mã nguồn, biểu đồ thể hiện mức lương nhận được dựa trên số lần đá cho Club trong các giải đấu



Hình 3.28. Mã nguồn thể hiện mức lương nhận được dựa trên số lần đá cho Club trong các giải đấu

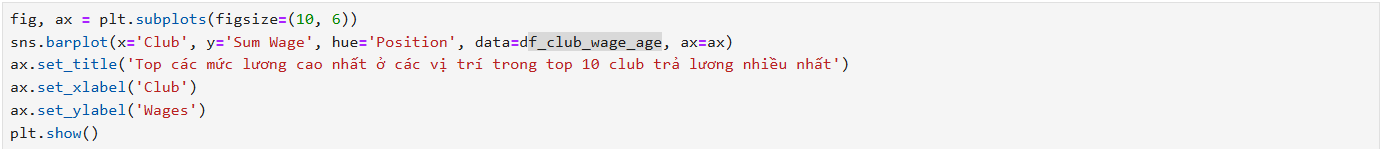
Chúng ta sử dụng phương thức subplots() để tạo các ô con của figure và có tham số kích thước của figure là (10,6) có chiều rộng là 10 và chiều dài là 6.

Sử dụng phương thức scatterplot() để vẽ biểu đồ phân tán với câu lệnh sns.scatterplot(x = “Apps”, y = “Wage”,hue =”League”, data = df\_league, ax = ax) có nghĩa tương ứng lần lượt như sau: trục x là các dữ liệu của Apps, trục y là các dữ liệu của wage, tham số hue là tham số phân biệt các điểm dữ liệu trên 1 biến cụ thể và dữ liệu được lấy từ df\_league, ax được vẽ tại vị trí chính nó.



Hình 3.29. Biểu đồ thể hiện mức lương nhận được dựa trên số lần đá cho Club trong các giải đấu

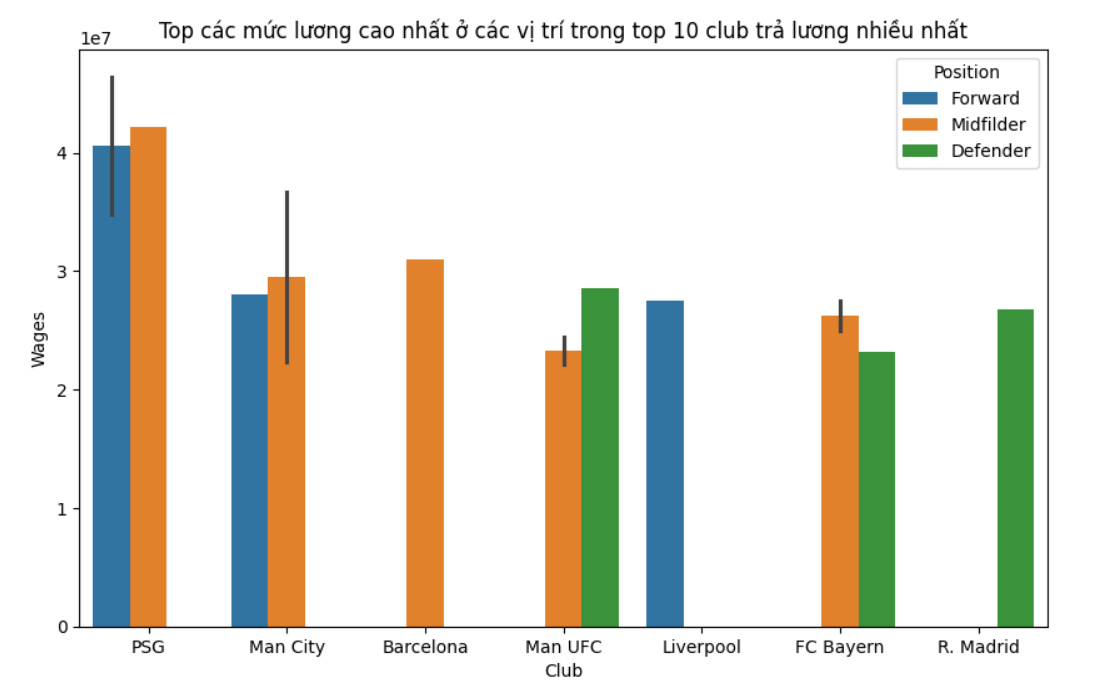
### Mã nguồn, biểu đồ top mức lương được trả cao nhất cho các vị trí trong top 10 Club trả lương cầu thủ cao nhất



Hình 3.30. Mã nguồn top mức lương được trả cao nhất cho các vị trí trong top 10 Club trả lương cầu thủ nhiều nhất

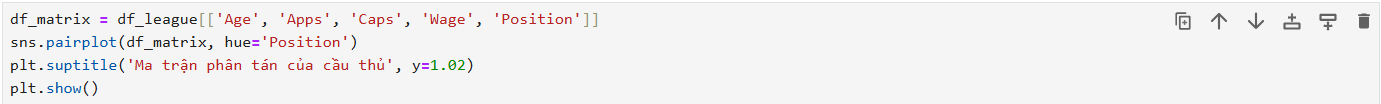
Chúng ta sử dụng phương thức subplots() để tạo các ô con của figure và có tham số kích thước của figure là (10,6) có chiều rộng là 10 và chiều dài là 6.

Sử dụng phương thức scatterplot() để vẽ biểu đồ phân tán với câu lệnh sns.scatterplot(x = “Club”, y = “Sum Wage”,hue =”Position”, data = df\_club\_wage\_age, ax = ax) có nghĩa tương ứng lần lượt như sau: trục x là các dữ liệu của Club, trục y là các dữ liệu của Sum Wage, tham số hue là tham số phân biệt các điểm dữ liệu trên 1 biến cụ thể và dữ liệu được lấy từ df\_club\_wage\_age, ax được vẽ tại vị trí chính nó.



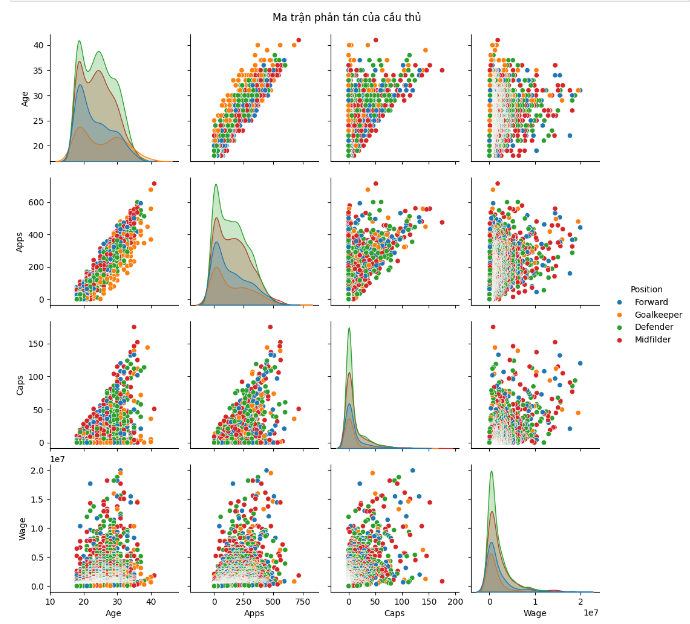
Hình 3.31.Biểu đồ top mức lương được trả cao nhất cho các vị trí trong top 10 Club trả lương cầu thủ nhiều nhất

### Mã nguồn, biểu đồ ma trận của cầu thủ



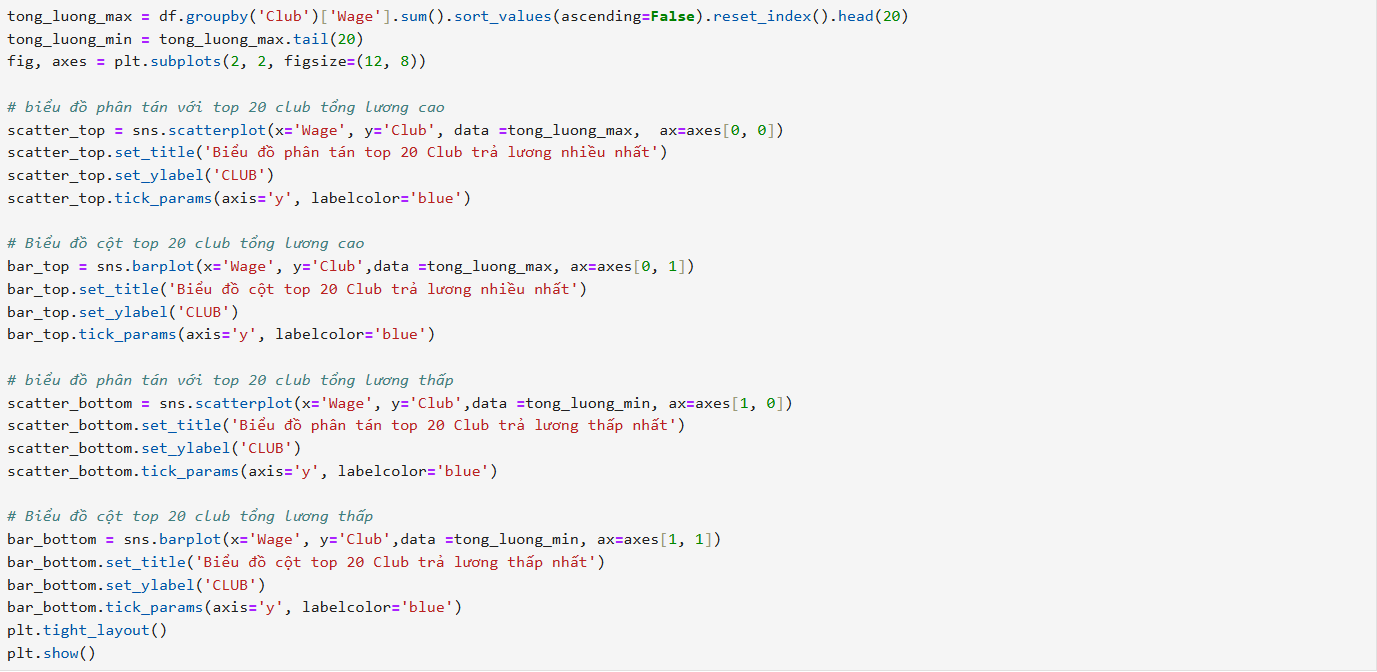
Hình 3.32. Mã nguồn ma trận của cầu thủ

Đầu tiên tạo 1 DataFrame dựa trên các cột mà ta chọn để vẽ biểu đồ mà trận. Sử dụng phương thức pairplot() để vẽ được biểu đồ ma trận, câu lệnh sns.pairplot(df\_matrix, hue = “Position”) có nghĩa lần lượt là dữ liệu được vẽ bởi DataFrame chúng ta muốn vẽ biểu đồ ma trận, hue là tham số phân biệt dữ liệu, sử dụng phương thức suptitle() để viết tiêu đề ở trên đầu biểu đồ



Hình 3.33.Biểu đồ ma trận của cầu thủ

### Mã nguồn, biểu đồ top 20 Club có tổng lương cao nhất và tổng lương thấp nhất



Hình 3.34. Mã nguồn top 20 Club có tổng lương cao nhất và tổng lương ít nhất

Đầu tiên chúng ta phải tính tổng lương phải trả nhiều nhất của các câu lạc bộ trong top 10 chúng ta cần tính tổng lương cho các câu lạc bộ bằng cách sử dụng phương thức groupby() và phương thức sum(). Phương thức groupby() dùng để nhóm các dữ liệu giống nhau vì vậy ở đây chúng ta sẽ nhóm “Club” và sử dụng sum() để tính tổng mức lương cho các câu lạc bộ. Kết hợp sử dụng phương thức rename\_axis() có chuỗi “Club” chuyền vào để thay đổi tên feature thành “Club” và phương thức reset\_index() để trở thành DataFrame chứa 2 cột “Club” và “Wage”, truyền tham số name =”Sum Age” để đổi tên cột dữ liệu “Wage”

Sau đó chúng ta sử dụng phương thức sort\_values() để sắp xếp các dữ liệu để sắp xếp dữ liệu ở cột đó và tham số ascending = Fasle để được sắp xếp dữ liệu từ cao đến thấp kết hợp phương thức head() có giá trị là 10 để lấy ra tổng lương trả cao nhất của các câu lạc bộ trong top 20 và sử dụng phương thức tail() để tổng lương trả ít nhát của các câu lạc bộ trong top 20

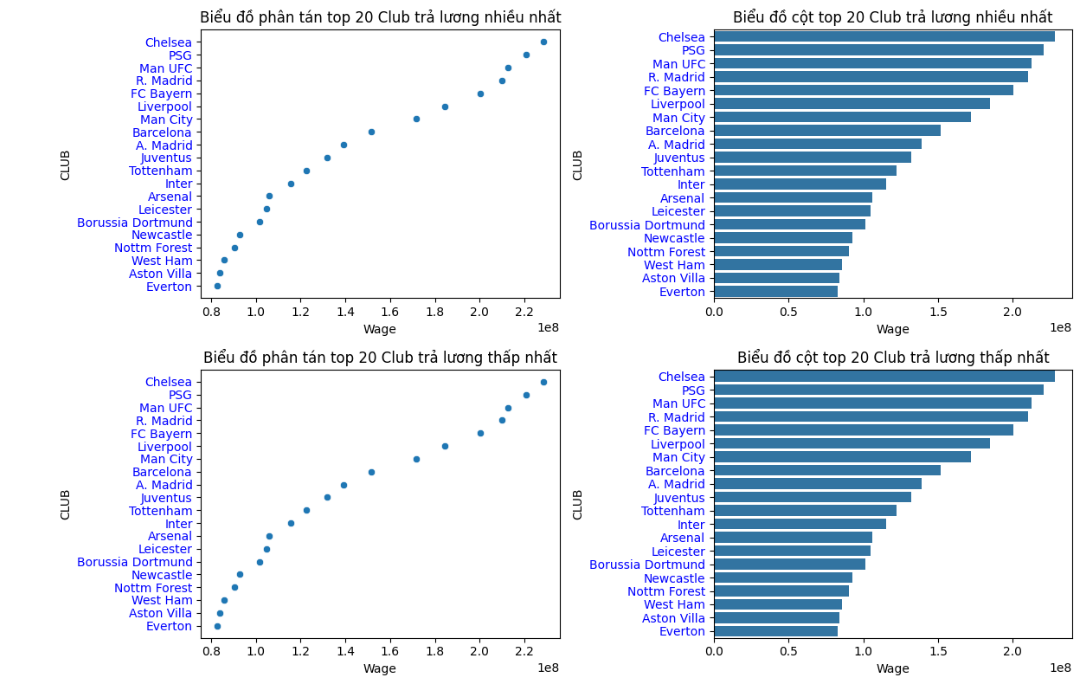
Tiếp theo chúng ta sẽ sử dụng phương thức subplots() để tạo các axes con, câu lệnh sns.subplots(2,2, figsize = (12,8)) có nghĩa tương ứng là gồm 2 hàng, 2 cột và figture có kích thước là rộng 12 và dài 8.

Sử dụng phương thức scatterplot() để vẽ biểu đồ phân tán với câu lệnh sns.scatterplot(x = “Wage”, y = “Club”, data = tong\_luong\_max, ax = ax[0,0]) có nghĩa tương ứng lần lượt như sau: trục x là các dữ liệu của Apps, trục y là các dữ liệu của wage, tham số hue là tham số phân biệt các điểm dữ liệu trên 1 biến cụ thể và dữ liệu được lấy từ tong\_luong\_max, ax được vẽ tại vị trí hàng 0 cột 0, sử dụng phương thức set\_title(), set\_ylabel(), có nghĩa tương ứng để đặt tên biểu đồ, đặt tên trục y, phương thức tick\_params() dùng để đánh dấu và có tham số axis = “y” đánh dấu tại trục y và sử đụng màu blue

Sử dụng phương thức scatterplot() để vẽ biểu đồ phân tán với câu lệnh sns.scatterplot(x = “Wage”, y = “Club”, data = tong\_luong\_min, ax = ax[1,0]) có nghĩa tương ứng lần lượt như sau: trục x là các dữ liệu của Wage, trục y là các dữ liệu của Club, tham số hue là tham số phân biệt các điểm dữ liệu trên 1 biến cụ thể và dữ liệu được lấy từ tong\_luong\_min, ax được vẽ tại vị trí hàng 1 cột 0 sử dụng phương thức set\_title(), set\_ylabel(), có nghĩa tương ứng để đặt tên biểu đồ, đặt tên trục y, phương thức tick\_params() dùng để đánh dấu và có tham số axis = “y” đánh dấu tại trục y và sử đụng màu blue

Sử dụng phương thức barplot() để vẽ biểu đồ phân tán với câu lệnh sns.scatterplot(x = “Wage”, y = “Club”, data = tong\_luong\_max, ax = ax[0,1]) có nghĩa tương ứng lần lượt như sau: trục x là các dữ liệu của Wage, trục y là các dữ liệu của Club, tham số hue là tham số phân biệt các điểm dữ liệu trên 1 biến cụ thể và dữ liệu được lấy từ tong\_luong\_max, ax được vẽ tại vị trí hàng 0 cột 1 sử dụng phương thức set\_title(), set\_ylabel(), có nghĩa tương ứng để đặt tên biểu đồ, đặt tên trục y, phương thức tick\_params() dùng để đánh dấu và có tham số axis = “y” đánh dấu tại trục y và sử đụng màu blue

Sử dụng phương thức barplot() để vẽ biểu đồ phân tán với câu lệnh sns.scatterplot(x = “Wage”, y = “Club”, data = tong\_luong\_min, ax = ax[1,1]) có nghĩa tương ứng lần lượt như sau: trục x là các dữ liệu của Wage, trục y là các dữ liệu của Club, tham số hue là tham số phân biệt các điểm dữ liệu trên 1 biến cụ thể và dữ liệu được lấy từ tong\_luong\_min, ax được vẽ tại vị trí hàng 1 cột 1 sử dụng phương thức set\_title(), set\_ylabel(), có nghĩa tương ứng để đặt tên biểu đồ, đặt tên trục y, phương thức tick\_params() dùng để đánh dấu và có tham số axis = “y” đánh dấu tại trục y và sử đụng màu blue



Hình 3.35. Biểu đồ top 20 Club có tổng lương cao nhất và tổng lương ít nhất

## Mô hình học máy (Machine Learning)

### Giới thiệu thư viện scikit - learn

Scikit-learn là một thư viện học máy mã nguồn mở cho Python. Nó cung cấp một loạt các công cụ hiệu quả và dễ sử dụng cho khai phá dữ liệu và phân tích dữ liệu. Scikit-learn được xây dựng trên các thư viện khoa học của Python như NumPy, SciPy, và matplotlib.

**Lý do sử dụng scikit - learn**

Dễ sử dụng và tài liệu phong phú: Scikit-learn có giao diện nhất quán và dễ sử dụng. Tài liệu của thư viện này rất phong phú và chi tiết.

Hỗ trợ nhiều thuật toán học máy: Scikit-learn cung cấp nhiều thuật toán học máy từ cơ bản đến nâng cao như hồi quy, phân loại, cụm (clustering), và giảm kích thước (dimensionality reduction).

Tích hợp tốt với các thư viện khoa học khác: Scikit-learn tích hợp tốt với các thư viện khoa học khác như NumPy và Pandas, giúp việc phân tích và xử lý dữ liệu trở nên dễ dàng.

Hiệu suất cao: Các thuật toán trong Scikit-learn được tối ưu hóa để có hiệu suất cao và sử dụng hiệu quả tài nguyên hệ thống.

**Ưu điểm của Scikit-learn**

Đa dạng thuật toán: Scikit-learn cung cấp nhiều thuật toán học máy bao gồm các thuật toán phân loại (classification), hồi quy (regression), cụm (clustering), và giảm kích thước (dimensionality reduction).

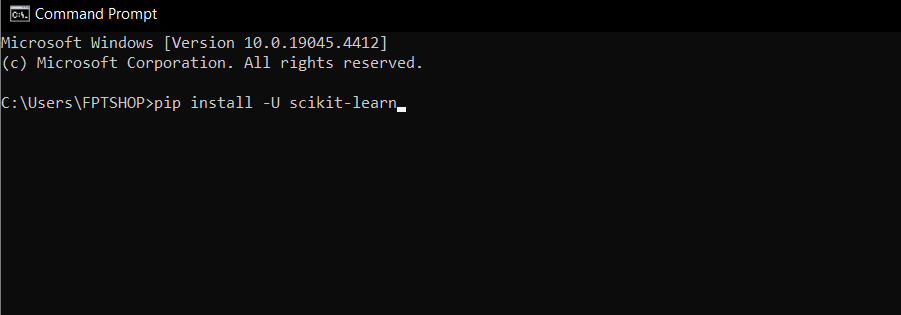
Giao diện nhất quán: Các thuật toán trong Scikit-learn có giao diện nhất quán, giúp bạn dễ dàng chuyển đổi giữa các thuật toán khác nhau.

Tiền xử lý dữ liệu: Scikit-learn cung cấp các công cụ mạnh mẽ cho tiền xử lý dữ liệu như chuẩn hóa (normalization), chuẩn hóa lại dữ liệu (scaling), và biến đổi dữ liệu (transformations).

Đánh giá mô hình: Scikit-learn có các công cụ để đánh giá hiệu suất của mô hình như cross-validation và các metric đánh giá.

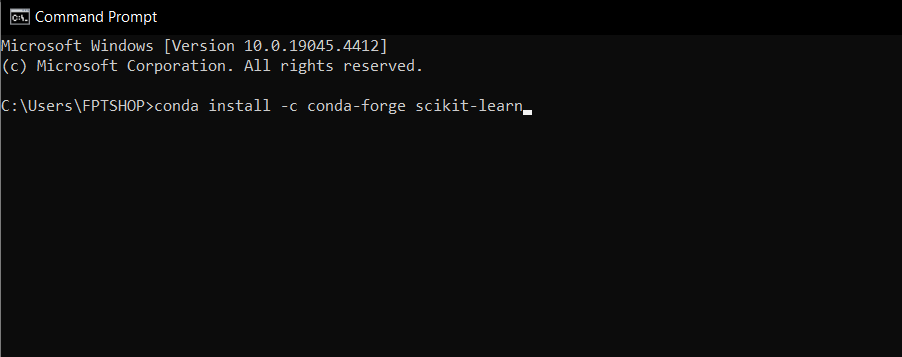
Tích hợp với các thư viện khác: Scikit-learn tích hợp tốt với các thư viện khoa học khác của Python như NumPy, Pandas, và Matplotlib.

### Cài đặt scikit - learn



Hình 3.36. Cài đặt scikit - learn

**Cài đặt scikit - learn bằng anaconda**



Hình 3.37. Cài đặt scikit - learn bằng conda

### Ví dụ về Scikit - learn

**Hồi quy tuyến tính (Linear Regression):**

import numpy as np

import pandas as pd

from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

Tạo dữ liệu mẫu

x = np.array([1, 2, 3, 4, 5]).reshape((-1, 1))

y = np.array([1, 4, 9, 16, 25])

Tạo mô hình hồi quy tuyến tính

model = LinearRegression()

model.fit(x, y)

Dự đoán:

y\_pred = model.predict(x)

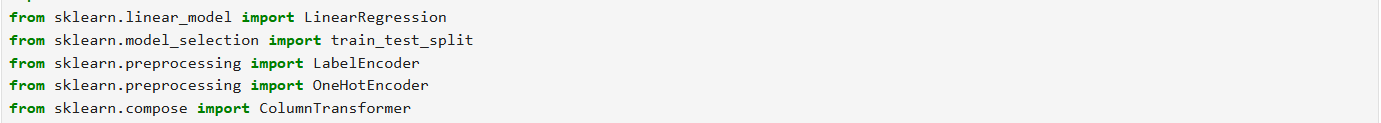
Vẽ biểu đồ

plt.scatter(x, y, color='blue')

plt.plot(x, y\_pred, color='red')

plt.show()

### Ứng dụng mô hình học máy

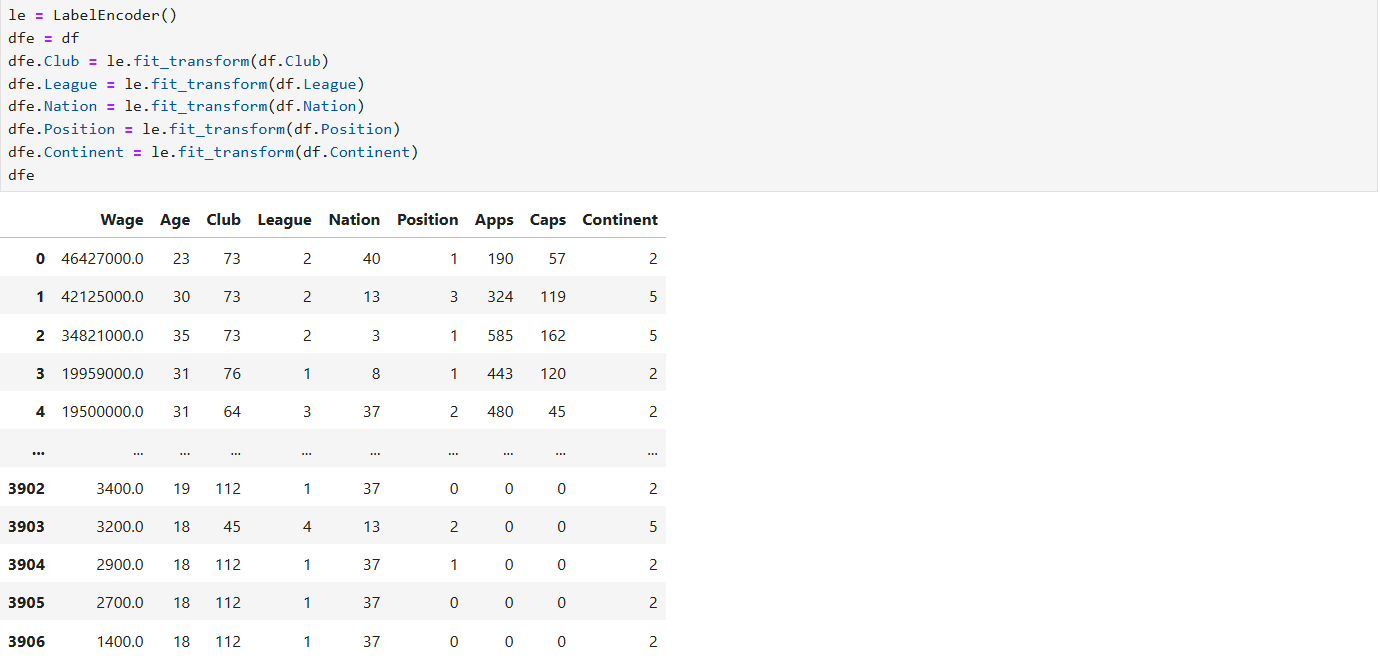


Hình 3.38. Sử dụng các thư viện để ứng dụng mô hình học máy

Để sử dụng mô hình học máy chúng ta cần sử dụng thư viện sklearn bao gồm: linear\_model, model\_selection, preprocessing, compose có nghĩa tương ứng như sau: mô hình hồi quy tuyến tính, mô hình lựa chọn dữ liệu, tiền xử lý dữ liệu, dự đoán tự động

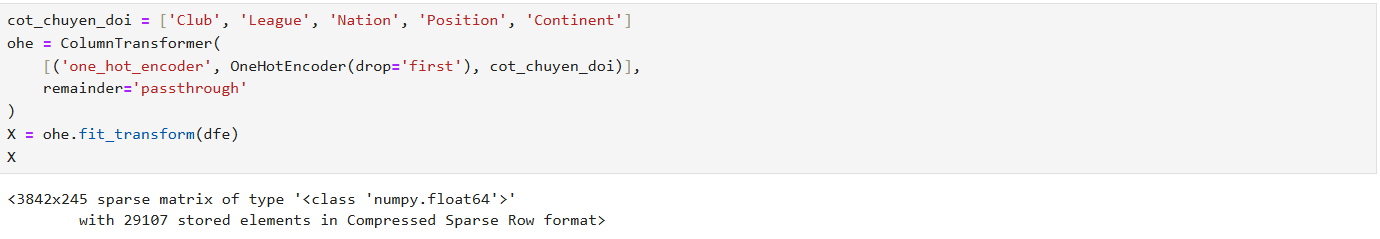
#### Mô hình hồi quy tuyến tính (LinearRegression)

Để sử dụng mô hình hồi quy tuyến tính bước đầu tiên chúng ta cần phải tiền xử lý dữ liệu bằng cách mã hoá dữ liệu với các dữ liệu văn bản sử dụng công cụ tiền xử lý dữ liệu LabelEncoder(). Sử dụng phương thức fit\_transform() để mã hoá dữ liệu với dạng dữ liệu là văn bản trở thành các giá trị được phân loại



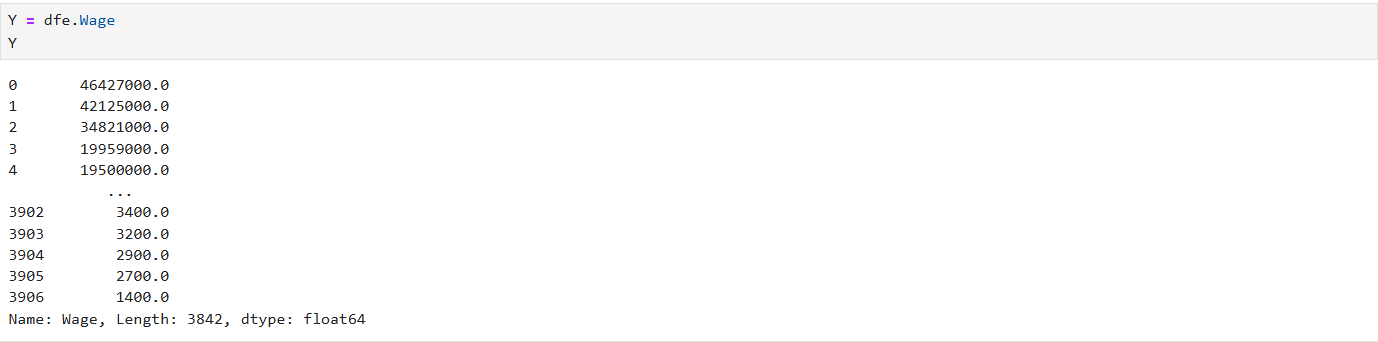
Hình 3.39. Mã hoá dữ liệu với dạng dữ liệu là văn bản

Tiếp theo chúng ta sẽ sử dụng công cụ tiền xử lý dữ liệu OneHotEncoder() để chuyển các giá trị được phân loại thành các vecto nhị phân, mỗi vecto chỉ có 1 lớp duy nhất



Hình 3.40. Sử dụng OneHotEncoder() để chuyển giá trị phân loại thành vecto nhị phân

Chúng ta chọn tên các cột vừa chuyển thành các giá trị phân loại lưu vào 1 list và sử dụng công cụ ColumnTransfomer() để chuyển đổi các dữ liệu mà liên quan đến list chúng ta vừa có. Câu lệnh: ColumnTransfomer([‘one\_hot\_encoder’, OneHotEncoder(drop = “first”), cot\_chuyen\_doi], remaider = “passthourgh”) có nghĩa tương ứng như sau: tên của OneHotEncoder, OneHotEncoder() chuyển đổi giá trị phân loại thành các vecto nhị phân với tham số drop = “first” xoá cột đầu của vecto sau khi chuyển đổi các giá trị phân loại,list liệt kê các cột cần biến đổi, tham số remaider = “passthourgh” chỉ định các cột không được liệt kê ở trong list sẽ giữ nguyên và không bị biến đổi. Sau đó chúng ta sử dụng phương thức fit\_transform() để thay thế và chuyển đổi dữ liệu và lưu vào 1 biến X coi như là giá trị X để train model

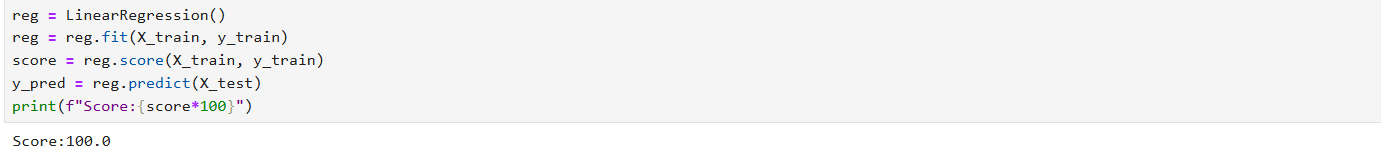


Hình 3.41. Lấy giá trị Y bằng cột Wage để dự đoán Wage

Tiếp theo chúng ta lấy cột cần dự đoán lưu vào 1 biến có tên là biến Y  

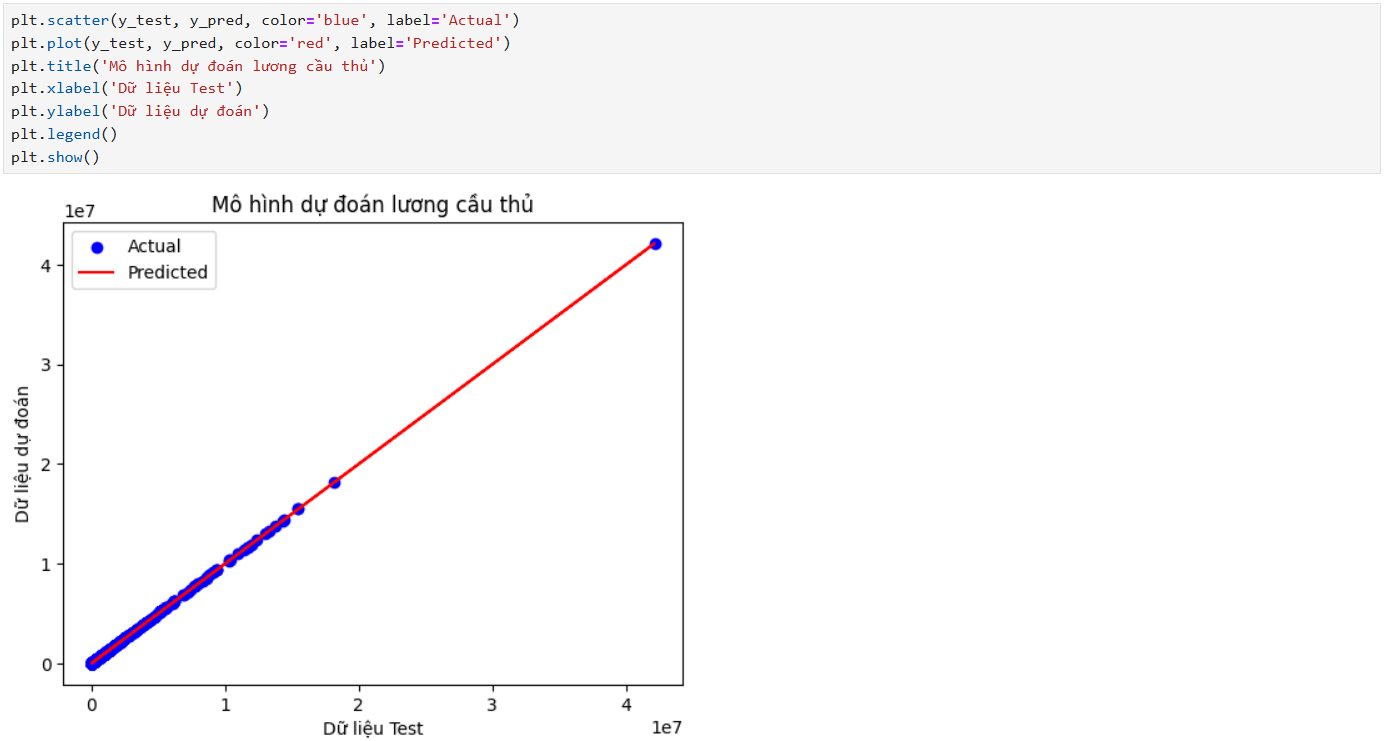

Hình 3.42. Lựa chọn số lượng train model và test model

Sau khi chọn xong X và Y chúng ta cần lựa chọn số lượng để train model và test model. Ở đây câu lệnh train\_test\_split(X, Y, test\_size=0.2, random\_state=10) có nghĩa tương ứng: Tham số X, Y, test\_size là tham số chỉ số lượng test cho model là bao nhiêu ở đây giá trị là 0.2 vậy tức là 20% giá trị để test\_model và 80% để trainning model, tham số random\_state = 10 giữ nguyên giá trị train của random được lưu vào thứ tự các biến X\_train, X\_test, y\_train, y\_test



Hình 3.43. Xây dựng mô hình hồi quy tuyến tính

Chúng ta đã có dữ liệu để train và dữ liệu để test chúng ta sẽ xây dựng mô hình, câu lệnh trên có nghĩa tương ứng như sau: Khởi tạo lớp đối tượng model mới, phương thức fit() dùng để huấn luyện LinearRegression Model, phương thức score() dùng để xem số điểm mô hình có tính hợp lệ là bao nhiêu, và phương thức predict() có nghĩa là dự đoán các dữ liệu cần dự đoán và hiển thị cho người dùng mô hình này phù hợp 100% cho file dữ liệu trên



Hình 3.44. Biểu đồ sự phân tán dữ liệu và dữ liệu dự đoán

# TÀI LIỆU THAM KHẢO

* 1. **Dữ liệu**

[**https://www.kaggle.com/datasets/ultimus/football-wages-prediction**](https://www.kaggle.com/datasets/ultimus/football-wages-prediction)

* 1. **Tài liệu tham khảo**

• Các tài liệu slide của thầy Nguyễn Văn Quyết

• Choosing color palettes, Seaborn, Choosing color palettes — seaborn

0.12.2 documentation (pydata.org)

• Building structured multi-plot grids, Seaborn, Building structured

multi-plot grids — seaborn 0.12.2 documentation (pydata.org)

• matplotlib.pyplot.bar, Matplotlib, matplotlib.pyplot.bar — Matplotlib

3.7.1 documentation

• pandas.DataFrame.sort\_values, Pandas,

pandas.DataFrame.sort\_values — pandas 2.0.2 documentation

(pydata.org)

• Indexing and selecting data, Pandas, Indexing and selecting data —

pandas 2.0.2 documentation (pydata.org)

• pandas.DataFrame.dropna, Pandas, pandas.DataFrame.dropna —

pandas 2.0.2 documentation (pydata.org)

• Merge, join, concatenate and compare, Pandas, Merge, join,

concatenate and compare — pandas 2.0.2 documentation (pydata.org)

• BoringPPL, Youtube, BoringPpl - Ứng Dụng Lập Trình Python -

YouTube