

Bài 1

Để lấy được tất cả các chỉ số theo yêu cầu ta sẽ truy cập vào mười url để lấy các dữ liệu lưu vào mười dataframe sau sử dụng hàm đó gộp chúng lại thành một bảng hoàn chỉnh

Ngoài các chỉ số cần thiết tại mỗi bảng ta sẽ lấy thêm các chỉ số name, team để làm các cột chung thuận tiện cho việc ghép bảng

Để lấy dữ liệu từ trang các trang web ta sử dụng các thư viện sau:

1. selenium: để tự động mở trình duyệt và truy cập trang web chứa dữ liệu.
2. webdriver_manager: tự động tải và cài đặt trình điều khiển ChromeDriver cho Selenium.
3. bs4 (BeautifulSoup): để phân tích cú pháp HTML và trích xuất thông tin từ trang web.
4. pandas: để lưu trữ và thao tác với dữ liệu dưới dạng bảng.

Lấy được các chỉ số :

+ Nation

+ Team

+ Position

+ Age

+ Playing time: matches played, starts, minutes

+ Performance: non-Penalty Goals, Penalty Goals, Assists, Yellow Cards, Red Cards

+ Expected: xG, npxG, xAG,

+ Progression: PrgC, PrgP, PrgR

+ Per 90 minutes: Gls, Ast, G+A, G-PK, G+A-PK, xG, xAG, xG + xAG, npxG, npxG + xAG

- Ta truy cập vào Truy cập URL: <https://fbref.com/en/comps/9/2023-2024/stats/2023-2024-Premier-League-Stats>
- Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_standard

- Duyệt qua từng hàng để lấy tất cả các thông tin trên dưới dạng text và đưa vào danh sách
- Chuyển đổi danh sách thành DataFrame có tên là bảng1

Lấy các chỉ số:

+ Goalkeeping:

. Performance: GA, GA90, SoTA, Saves, Save%, W, D, L, CS, CS%

. Penalty Kicks: PKatt, PKA, PKsv, PKm, Save%

- Ta truy cập vào URL: <https://fbref.com/en/comps/9/2023-2024/keepers/2023-2024-Premier-League-Stats>
- Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_keeper
- Duyệt qua từng hàng để lấy tất cả các thông tin trên dưới dạng text và đưa vào danh sách
- Chuyển đổi danh sách thành DataFrame có tên là Goalkeeping

Lấy được các chỉ số:

+ Shooting:

. Standard: Gls, Sh, SoT, SoT%, Sh/90, SoT/90, G/Sh, G/SoT, Dist, FK, PK, PKatt

. Expected: xG, npG, npG/Sh, G-xG, np:G-xG

- Ta truy cập vào URL : <https://fbref.com/en/comps/9/2023-2024/shooting/2023-2024-Premier-League-Stats>
- Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_shooting

- Duyệt qua từng hàng để lấy tất cả các thông tin trên dưới dạng text và đưa vào danh sách
- Chuyển đổi danh sách thành DataFrame có tên là Shooting

Lấy các chỉ số:

+ Passing:

- . Total: Cmp, Att, Cmp%, TotDist, PrgDist
- . Short: Cmp, Att, Cmp%
- . Medium: Cmp, Att, Cmp%
- . Long: Cmp, Att, Cmp%
- . Expected: Ast, xAG, xA, A-xAG, KP, 1/3, PPA, CrsPA, PrgP

- Ta truy cập vào URL: <https://fbref.com/en/comps/9/2023-2024/passing/2023-2024-Premier-League-Stats>
- Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_passing
- Duyệt qua từng hàng để lấy tất cả các thông tin trên và đưa vào danh sách
- Chuyển đổi danh sách thành DataFrame có tên là Passing

Lấy các chỉ số:

+ Pass Types:

- . Pass Types: Live, Dead, FK, TB, Sw, Crs, TI, CK
- . Corner Kicks: In, Out, Str
- . Outcomes: Cmp, Off, Blocks

- Ta truy cập vào URL: https://fbref.com/en/comps/9/2023-2024/passing_types/2023-2024-Premier-League-Stats
- Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_passing_types
- Duyệt qua từng hàng để lấy tất cả các thông tin trên dưới dạng text và đưa vào danh sách
- Chuyển đổi danh sách thành DataFrame có tên là PassTypes

Lấy các chỉ số:

+ Goal and Shot Creation:

. SCA: SCA, SCA90

. SCA Types: PassLive, PassDead, TO, Sh, Fld, Def

. GCA: GCA, GCA90

. GCA Types: PassLive, PassDead, TO, Sh, Fld, Def

- Ta truy cập vào Truy cập URL: <https://fbref.com/en/comps/9/2023-2024/gca/2023-2024-Premier-League-Stats>
- Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_gca
- Duyệt qua từng hàng để lấy tất cả các thông tin trên dưới dạng text và đưa vào danh sách
- Chuyển đổi danh sách thành DataFrame có tên là GoalandShotCreation

Lấy các chỉ số:

+ Defensive Actions:

- . Tackles: Tkl, TklW, Def 3rd, Mid 3rd, Att 3rd
- . Challenges: Tkl, Att, Tkl%, Lost
- . Blocks: Blocks, Sh, Pass, Int, Tkl + Int, Clr, Err
- Ta truy cập vào Truy cập URL: <https://fbref.com/en/comps/9/2023-2024/defense/2023-2024-Premier-League-Stats>
- Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_defense
- Duyệt qua từng hàng để lấy tất cả các thông tin trên và đưa vào danh sách
- Chuyển đổi danh sách thành DataFrame có tên là DefensiveActionpsd

Lấy các chỉ số:

+ Possession:

- . Touches: Touches, Def Pen, Def 3rd, Mid 3rd, Att 3rd, Att Pen, Live
- . Take-Ons: Att, Succ, Succ%, Tkld, Tkld%
- . Carries: Carries, TotDist, ProDist, ProgC, 1/3, CPA, Mis, Dis
- . Receiving: Rec, PrgR
- Ta truy cập vào Truy cập URL: <https://fbref.com/en/comps/9/2023-2024/possession/2023-2024-Premier-League-Stats>
- Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_possession
- Duyệt qua từng hàng để lấy tất cả các thông tin trên dưới dạng text và đưa vào danh sách

- Chuyển đổi danh sách thành DataFrame có tên là Possession

Lấy các chỉ số

+ Playing Time

- . Starts: Starts, Mn/Start, Compl
 - . Subs: Subs, Mn/Sub, unSub
 - . Team Success: PPM, onG, onGA
 - . Team Success xG: onxG, onxGA
- Ta truy cập vào Truy cập URL: <https://fbref.com/en/comps/9/2023-2024/playingtime/2023-2024-Premier-League-Stats>
 - Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_playing_time
 - Duyệt qua từng hàng để lấy tất cả các thông tin trên dưới dạng text và đưa vào danh sách
 - Chuyển đổi danh sách thành DataFrame có tên là Playingtime

Lấy các chỉ số:

+ Miscellaneous Stats:

- . Performance: Fls, Fld, Off, Crs, OG, Recov
 - . Aerial Duels: Won, Lost, Won%
- Ta truy cập vào Truy cập URL: <https://fbref.com/en/comps/9/2023-2024/misc/2023-2024-Premier-League-Stats>
 - Sau khi truy cập vào trang, mã sử dụng BeautifulSoup để lấy mã nguồn HTML của trang và xác định bảng dữ liệu cần thiết bằng cách tìm bảng có thuộc tính id là stats_misc

- Duyệt qua từng hàng để lấy tất cả các thông tin trên dưới dạng text và đưa vào danh sách
- Chuyển đổi danh sách thành DataFrame có tên là MiscellaneousStats

Ghép các bảng

- Danh sách dataframes bao gồm nhiều bảng dữ liệu (DataFrame) chứa các thông tin khác nhau về cầu thủ (như Goalkeeping, Shooting, Passing, v.v.). Sử dụng hàm reduce với pd.merge để hợp nhất tất cả các bảng trong dataframes dựa trên hai cột chung là Name và Team. Phương pháp hợp nhất how='outer' được sử dụng để giữ lại tất cả các dữ liệu, kể cả khi có giá trị khuyết thiếu.

Chuẩn hóa dữ liệu

- Các dữ liệu là số lớn hơn 1000 sẽ chứa kí tự ',' nên t sẽ loại bỏ các kí tự đó nếu không bỏ các ô này sẽ không thể chuyển được về dạng số thực
- Sau đó chuyển về dạng số thực các cột thoải mái
- Loại các hàng(các cầu thủ) có cột Playing time_Minutes < 90
- Một cột phụ First Name được tạo ra để chứa tên đầu tiên của cầu thủ (họ). Dữ liệu được sắp xếp theo thứ tự Name (tên đầy đủ) và Age (theo thứ tự giảm dần). Sau đó, cột First Name phụ được loại bỏ để giữ dữ liệu gọn gàng.

Lưu dữ liệu vào tệp CSV

- Dữ liệu cuối cùng được lưu vào tệp results.csv, với cấu trúc và định dạng dễ đọc, thuận tiện cho các phân tích tiếp theo.

BÀI 2

Tìm top 3 cầu thủ có điểm cao nhất và thấp nhất ở mỗi chỉ số

Bài toán được giải quyết bằng cách sử dụng thư viện Pandas để xử lý dữ liệu từ file CSV. Các bước chính bao gồm:

1. Đọc dữ liệu từ file CSV: Dữ liệu từ file results.csv được đọc vào một DataFrame của Pandas để thuận tiện cho việc thao tác và tính toán.
2. Lọc các cột dạng số: Do bài toán yêu cầu tính toán trên các chỉ số (metrics) nên chúng ta chỉ quan tâm tới các cột có kiểu dữ liệu dạng số (float64 hoặc int64). Các cột không phải là số sẽ được bỏ qua để tránh lỗi khi thực hiện các phép so sánh và tìm kiếm.
3. Tìm top 3 và bottom 3 cầu thủ ở mỗi chỉ số:
 - Với mỗi cột dạng số ta thực hiện
 - Top 3: Dùng hàm nlargest() để tìm ra 3 giá trị cao nhất trong cột đó, giúp xác định 3 cầu thủ có điểm số cao nhất cho chỉ số.
 - Bottom 3: Tương tự, dùng hàm nsmallest() để tìm ra 3 giá trị thấp nhất trong cột đó, xác định 3 cầu thủ có điểm thấp nhất cho chỉ số.
 - Mỗi cầu thủ trong danh sách top 3 và bottom 3 sẽ được lưu vào danh sách top_bottom_results với các thông tin:
 - Metric: Tên chỉ số
 - Metric: Tên chỉ số
 - Player: Tên cầu thủ
 - Value: Giá trị của chỉ số
4. Tạo DataFrame từ kết quả:
 - Dữ liệu từ top_bottom_results được chuyển thành một DataFrame mới result_df để dễ xử lý và lưu trữ.
 - Mỗi hàng trong result_df chứa tên chỉ số, loại ("Top" hoặc "Bottom"), tên cầu thủ, và giá trị điểm số tương ứng.
5. Lưu kết quả ra file CSV:
 - Kết quả được lưu vào file result1.csv để có thể truy xuất và phân tích dễ dàng. File này chứa toàn bộ thông tin về top và bottom 3 cầu thủ ở từng chỉ số dạng số.

Tìm trung vị của mỗi chỉ số. Tìm trung bình và độ lệch chuẩn của mỗi chỉ số cho các cầu thủ trong toàn giải và của mỗi đội

Ta thực hiện các bước sau:

1. Đọc dữ liệu từ file CSV:
 - File results.csv chứa dữ liệu các cầu thủ, được đọc vào DataFrame df.
2. Xác định các cột có kiểu số:
 - Để thực hiện tính toán thống kê, mã lọc các cột có kiểu dữ liệu số (float64 hoặc int64). Chỉ các cột dạng số mới cần thiết cho việc tính toán median, mean, và standard deviation (std).
3. Tính toán các chỉ số cho toàn giải:
 - Với mỗi cột dạng số, các chỉ số median, mean, và std của toàn giải được tính bằng các hàm median(), mean(), và std() trong Pandas.
 - Kết quả cho toàn giải của mỗi chỉ số được thêm vào danh sách
 - final_results
4. Tính toán các chỉ số cho từng đội:
 - Với mỗi cột dạng số, dữ liệu được nhóm (groupby) theo đội, và các chỉ số median, mean, và std của từng đội cho cột đó được tính toán.
 - Kết quả của từng đội được thêm vào final_results theo định dạng
 - team_name (Tên đội)
 - median, mean, và std của chỉ số
5. Kiểm tra và thêm dữ liệu cho từng đội:
 - Để đảm bảo không có dữ liệu bị trùng, mã kiểm tra xem đội đã có trong final_results hay chưa. Nếu đội đã có, kết quả của chỉ số mới sẽ được cập nhật vào dòng tương ứng. Nếu chưa, thông tin đội sẽ được thêm mới vào danh sách final_results.
6. Tạo DataFrame từ kết quả và đặt tên cột:
 - Dữ liệu từ final_results được chuyển thành DataFrame final_df, giúp dễ dàng lưu trữ và truy xuất.
 - Các tên cột được cập nhật để bao gồm Median of <col_name>, Mean of <col_name>, và Std of <col_name> tương ứng cho từng chỉ số trong file kết quả.

7. Lưu kết quả ra file CSV:

- DataFrame `final_df` được lưu vào file `results2.csv`, bao gồm các chỉ số thống kê cho toàn giải và từng đội.

Vẽ historgram phân bố của mỗi chỉ số của các cầu thủ trong toàn giải và mỗi đội

1 Mục tiêu:

- Mỗi chỉ số sẽ có một biểu đồ phân phối cho toàn giải và một loạt biểu đồ phân phối riêng biệt cho từng đội.

2. Đọc dữ liệu và xử lý tên cột:

- Dữ liệu được đọc từ file CSV `results.csv` với nhiều tiêu đề (`header=[0, 1, 2]`), do đó các cột sẽ có cấu trúc là bộ 3 (tuple), chứa các tiêu đề đa cấp cho từng chỉ số.
- Để làm cho tên cột dễ quản lý hơn, mã thay thế những tiêu đề không cần thiết (các tiêu đề `Unnamed` trong file gốc) bằng chuỗi rỗng (`"`). Việc này giúp làm sạch tên cột, giữ lại chỉ các tiêu đề quan trọng của chỉ số.

3. Lọc các cột có kiểu dữ liệu số:

- Mã chỉ tập trung vào các cột có dữ liệu số (float và int), vì chúng là các chỉ số cần phân tích phân phối.

4. Vẽ biểu đồ phân phối cho toàn giải:

- Với từng chỉ số dạng số, mã sử dụng biểu đồ histogram (biểu đồ tần số) với `sns.histplot` để hiển thị sự phân phối của chỉ số đó trong toàn bộ giải đấu.
- Các biểu đồ này có dạng chuẩn với:
 - Trục x: Chỉ số của cầu thủ.
 - Trục y: Tần suất (số lượng cầu thủ).
- Mỗi biểu đồ đi kèm với `kde=True`, giúp hiển thị đường cong mật độ, giúp dễ nhận ra các mẫu và sự phân phối của dữ liệu.

5. Vẽ biểu đồ phân phối của từng chỉ số theo đội:

- Mã sử dụng FacetGrid để tạo biểu đồ histogram của từng chỉ số cho từng đội trong giải.
 - FacetGrid tạo ra nhiều biểu đồ nhỏ, mỗi biểu đồ tương ứng với một đội, cho phép dễ dàng so sánh sự phân phối của chỉ số giữa các đội.
 - Trong mỗi biểu đồ:
 - Trục x: Chỉ số đang phân tích
 - Trục y: Tần suất của chỉ số trong đội.
 - Các biểu đồ được hiển thị theo cách sắp xếp tự động (col_wrap=4), giúp dễ quan sát và tối ưu không gian hiển thị.
6. Thiết lập nhãn và tiêu đề:
- Mỗi biểu đồ được đặt tiêu đề và nhãn theo tên của chỉ số
 - Mã cũng định dạng tiêu đề để tên chỉ số và đội bóng được hiển thị rõ ràng trong biểu đồ, giúp người đọc dễ dàng phân biệt và so sánh

Tìm đội bóng có chỉ số điểm số cao nhất ở mỗi chỉ số

Ta sẽ sử dụng các chỉ số trung bình của các cầu thủ trong đội bóng nên ta sử dụng dữ liệu từ file results2.csv

1. Đọc dữ liệu

- Sử dụng thư viện pandas, chúng tôi đã nạp dữ liệu từ file results2.csv vào một DataFrame để tiến hành phân tích. Dữ liệu này bao gồm các thông tin về đội bóng và các chỉ số thống kê khác nhau (số lần ghi bàn, phòng thủ, v.v.).

2. Loại bỏ hàng tổng hợp

- Một số dòng dữ liệu có giá trị 'All' trong cột Team, đại diện cho các tổng hợp hoặc giá trị trung bình của giải đấu. Những hàng này không phù hợp cho việc tìm đội bóng dẫn đầu vì chúng không phản ánh hiệu suất của một đội bóng cụ thể nào. Do đó, chúng tôi đã loại bỏ những dòng này khỏi DataFrame.

3. Xác định đội bóng có thành tích cao nhất cho từng chỉ số

- Sau khi loại bỏ các hàng tổng hợp, chúng tôi đặt cột Team làm chỉ mục (index) để dễ dàng thao tác dữ liệu. Sau đó, chúng tôi sử dụng hàm idxmax() để tìm ra đội bóng có giá trị cao nhất cho mỗi chỉ số. Điều này cho phép xác định đội dẫn đầu trong các khía cạnh cụ thể, như ghi bàn, phòng thủ hoặc tốc độ.

Đội phong độ nhất

Cộng giá trị của cột Mean of Performance_non-Penalty Goals và Mean of Performance_Penalty Goals ta được tổng bàn thắng phạt đền và không phạt đền. Sau đó tìm đội có giá trị này lớn nhất

Ta thấy Manchester City là đội có giá trị này lớn nhất nên Manchester City sẽ là đội phong độ nhất

BÀI 3

Sử dụng thuật toán K-means để phân loại các cầu thủ thành các nhóm có chỉ số giống nhau

1. Xử lý Dữ liệu

- Dữ liệu được tải từ tệp results.csv. Để chuẩn bị cho phân cụm, các bước xử lý được thực hiện:
 - Lọc các cột số: Chỉ chọn những cột chứa dữ liệu dạng số (float, int) vì KMeans yêu cầu dữ liệu đầu vào phải là dạng số.
 - Xử lý giá trị thiếu (NaN): Các giá trị thiếu được thay thế bằng giá trị trung bình của từng cột. Việc này giúp đảm bảo rằng dữ liệu đầy đủ và không gây ảnh hưởng tới quá trình phân cụm.

2. Xác định số cụm tối ưu bằng Phương pháp Elbow

- Phương pháp Elbow được sử dụng để xác định số lượng cụm tối ưu cho dữ liệu. Ý tưởng là:
 - Thực hiện phân cụm với các giá trị k từ 2 đến 9.
 - Tính tổng bình phương khoảng cách từ các điểm dữ liệu đến tâm cụm của chúng (SSE - Sum of Squared Errors) cho mỗi k.
 - Vẽ biểu đồ SSE theo số cụm k.
- Trên biểu đồ, điểm gấp khúc ("khủy tay") thường là giá trị k tối ưu vì tại điểm này, SSE bắt đầu giảm chậm hơn, cho thấy sự giảm lợi ích khi tăng số cụm.
- Nhìn hình ta chọn k=4 với đoạn này sự giảm lợi ích khi tăng số cụm

Sử dụng thuật toán PCA, giảm số chiều dữ liệu xuống 2 chiều, vẽ hình phân cụm các điểm dữ liệu trên mặt 2D.

1. Dữ liệu được tải từ tệp results.csv và lưu vào biến data

2. Xử lý dữ liệu thiếu: Chọn các cột số trong dữ liệu (data_numeric) và điền các giá trị thiếu (NaN) bằng giá trị trung bình của mỗi cột. Việc này đảm bảo dữ liệu không bị mất giá trị và các phép tính phân cụm không bị ảnh hưởng.

3. Chuẩn hóa dữ liệu: Dữ liệu được chuẩn hóa với StandardScaler, chuyển đổi các giá trị thành dạng có phân phối chuẩn với giá trị trung bình bằng 0 và phương sai bằng 1. Việc này làm cho các đặc trưng có tầm quan trọng đồng đều khi phân cụm.

4. Dữ liệu sau khi chuẩn hóa được chuyển thành dạng 2 chiều với PCA, giúp trực quan hóa dữ liệu sau khi phân cụm.

5. Phương pháp KMeans với 4 cụm (n_clusters=4) được áp dụng trên dữ liệu đã chuẩn hóa. Kết quả phân cụm (clusters) chứa nhãn cụm cho mỗi điểm dữ liệu.

Sử dụng kết quả từ PCA (dữ liệu 2D) và KMeans (nhãn cụm), biểu đồ phân cụm trong không gian 2 chiều được tạo ra. Mỗi điểm dữ liệu trên biểu đồ thể hiện một quan sát trong dữ liệu gốc, với màu sắc thể hiện cụm mà nó thuộc về

Viết chương trình python vẽ biểu đồ rada (radar chart) so sánh cầu thủ

1. Chọn cầu thủ và chỉ số so sánh

- Đầu tiên, ta chọn hai cầu thủ muốn so sánh và các chỉ số cụ thể đại diện cho hiệu suất của họ.
 - Cầu thủ: Max Aarons và Joshua Acheampong
 - Chỉ số: Playing time_Starts, Playing time_Matches played và Performance_Penalty Goals

2. Trích xuất dữ liệu

- Dữ liệu của mỗi cầu thủ được trích xuất từ tệp CSV result.csv dựa trên tên cầu thủ và các chỉ số đã chọn. Mỗi cầu thủ sẽ có một tập hợp các giá trị chỉ số tương ứng.

3. Chuẩn bị góc cho các trục

- Vì biểu đồ radar có cấu trúc tròn, mỗi chỉ số cần được đặt tại một góc cụ thể. Với num_vars là số lượng chỉ số:
 - Các góc cho trục của các chỉ số được tính bằng cách chia đều góc 360 độ.
 - Các góc này giúp bố trí các chỉ số quanh hệ tọa độ tròn, tạo thành hình đa giác biểu diễn dữ liệu.

4. Đóng vòng điểm đầu

- Để khép kín hình đa giác của biểu đồ, giá trị đầu tiên của mỗi cầu thủ được thêm lại vào cuối chuỗi dữ liệu, đảm bảo rằng biểu đồ được khép kín khi vẽ.

5. Vẽ biểu đồ radar

- Biểu đồ radar được tạo trên hệ tọa độ phân cực (`polar=True`) với mỗi cầu thủ là một hình đa giác khác nhau.
- Dữ liệu của từng cầu thủ được tô màu (màu xanh dương cho Max Arons và màu đỏ cho Joshua Acheampong) và có độ trong suốt (`alpha`) để dễ phân biệt các vùng dữ liệu chồng lên nhau.
- Các nhãn trục là các chỉ số được đặt trên các góc tương ứng, tạo thành một vòng tròn hoàn chỉnh.

6. Hiển thị biểu đồ