# Báo cáo: Phân tích Đội Bóng Dẫn Đầu Các Chỉ Số Thống Kê

## Bài 2.1:

### 1. Mục tiêu

Mã này đọc dữ liệu cầu thủ từ file results.csv, xác định top 3 cầu thủ có chỉ số cao nhất và thấp nhất ở từng loại thống kê, và lưu kết quả vào file top_bottom_players.csv.

### 2. Các bước triển khai

- **Đọc dữ liệu**: Dùng pandas.read_csv() để đọc file results.csv vào DataFrame df.
- **Khởi tạo từ điển kết quả**: Tạo từ điển results để lưu danh sách top 3 cầu thủ cho từng chỉ số.
- **Vòng lặp qua các cột thống kê**: Với mỗi cột (bắt đầu từ cột thứ 8), xác định:
    - **Top 3 giá trị cao nhất**: Dùng nlargest(3, col) để tìm các cầu thủ có chỉ số cao nhất, lưu vào results.
    - **Top 3 giá trị thấp nhất**: Dùng nsmallest(3, col) để tìm các cầu thủ có chỉ số thấp nhất, lưu vào results.
- **Tạo DataFrame kết quả**: Chuyển đổi từ điển results thành DataFrame results_df.
- **Xuất file CSV**: Lưu DataFrame này vào file top_bottom_players.csv.

### 3. Kết quả

File top_bottom_players.csv chứa top 3 cầu thủ có thành tích cao nhất và thấp nhất theo từng chỉ số, giúp dễ dàng so sánh các chỉ số của các cầu thủ trong giải đấu.

## Bài 2.2:

### 1. Mục tiêu

Mã này tính toán các thống kê cơ bản (trung vị, trung bình, độ lệch chuẩn) cho từng chỉ số trong bảng dữ liệu results.csv, bao gồm:

- Các giá trị tổng quan cho toàn bộ cầu thủ.

- Các giá trị riêng cho từng đội bóng.

**2. Các bước triển khai**

- **Đọc dữ liệu**: Sử dụng pd.read_csv() để đọc dữ liệu từ results.csv và lưu vào DataFrame df.
- **Chuẩn bị từ điển kết quả**: Tạo từ điển output_data để lưu kết quả. Từ điển này bắt đầu với một mục chứa các giá trị tổng quan của toàn bộ cầu thủ ('Team': 'All').
- **Xác định các cột số liệu cần phân tích**: Sử dụng df.select_dtypes(include=['float64', 'int64']).columns để liệt kê các cột có kiểu dữ liệu số (cần tính trung bình, trung vị, và độ lệch chuẩn).
- **Tính toán các thống kê tổng quan**:
    - Với mỗi cột số, tính toán median (trung vị), mean (trung bình), và std (độ lệch chuẩn) cho toàn bộ cầu thủ và lưu kết quả vào output_data.
- **Tính toán thống kê cho từng đội bóng**:
    - Sử dụng df.groupby('Squad') để nhóm dữ liệu theo đội bóng.
    - Tính các giá trị median, mean, và std cho từng chỉ số của từng đội và lưu vào từ điển team_stats.
    - Thêm team_stats vào output_data dưới dạng DataFrame, nối dữ liệu từng đội vào sau dữ liệu tổng quan.
- **Xuất kết quả ra CSV**: Chuyển đổi output_data thành DataFrame và lưu vào file results2.csv.

**3. Kết quả**

File results2.csv chứa các thống kê tổng quan và riêng lẻ cho từng đội bóng, giúp phân tích các chỉ số theo cả tổng thể và từng đội trong giải đấu.

## Bài 2.3:

**1. Mục tiêu**

Đoạn mã này tạo các biểu đồ histogram để hiển thị phân phối của từng chỉ số cầu thủ cho toàn bộ giải đấu và từng đội bóng. Mỗi biểu đồ histogram đi kèm với đường phân bố mật độ (kde) để làm nổi bật đặc điểm phân phối của các chỉ số.

## 2. Các bước triển khai

- **Đọc dữ liệu**: Sử dụng pd.read_csv() để đọc file results.csv vào DataFrame df, với dấu phân cách là ;.
- **Chọn các cột chỉ số**: Xác định các cột chứa chỉ số cần phân tích. Đoạn mã giả định các chỉ số bắt đầu từ cột thứ 5 trở đi, với df.columns[4:] (giả sử cột đầu tiên là tên cầu thủ, đội bóng, và các thông tin cơ bản).
- **Vẽ histogram cho toàn giải**:
    - Với mỗi cột trong danh sách các cột chỉ số (stats_columns), sử dụng sns.histplot() để vẽ histogram thể hiện phân bố giá trị cho toàn bộ giải đấu.
    - Biểu đồ đi kèm với đường kde=True để thêm đường phân bố mật độ giúp dễ dàng quan sát xu hướng phân bố của chỉ số.
    - Thiết lập plt.title(), plt.xlabel(), và plt.ylabel() để biểu đồ trực quan và có ý nghĩa.

```
# Vẽ histogram phân bố cho toàn giải
for col in stats_columns:
    plt.figure(figsize=(10, 6))
    sns.histplot(df[col].dropna(), kde=True)  # vẽ histogram với đường phân bố
    plt.title(f'Phân bố {col} cho toàn giải')
    plt.xlabel(col)
    plt.ylabel('Số lượng cầu thủ')
    plt.show()
```

- **Vẽ histogram cho từng đội bóng**:
    - Dùng df['Squad'].unique() để lấy danh sách các đội bóng.
    - Với mỗi đội, lọc dữ liệu theo đội (df[df['Squad'] == team]) để tạo DataFrame riêng df_team.
    - Với mỗi chỉ số, sử dụng sns.histplot() để vẽ biểu đồ histogram của chỉ số đó cho đội bóng cụ thể, cùng với đường phân bố mật độ.
    - Các thiết lập plt.title(), plt.xlabel(), và plt.ylabel() được tùy chỉnh theo đội bóng và chỉ số.

```
for team in teams:
    df_team = df[df['Squad'] == team]
    for col in stats_columns:
        plt.figure(figsize=(10, 6))
        sns.histplot(df_team[col].dropna(), kde=True)
        plt.title(f'Phân bố {col} cho đội {team}')
        plt.xlabel(col)
        plt.ylabel('Số lượng cầu thủ')
        plt.show()
```

**3. Kết quả**

Các biểu đồ histogram sẽ được tạo cho từng chỉ số, thể hiện phân phối dữ liệu:

- **Toàn giải đấu**: Biểu đồ cho toàn bộ giải đấu giúp hiểu tổng quát phân bố của từng chỉ số.
- **Từng đội bóng**: Biểu đồ cho từng đội bóng giúp thấy đặc điểm phân bố của các chỉ số trong từng đội, từ đó so sánh và đối chiếu với dữ liệu toàn giải.

**4. Tổng kết**

Đoạn mã giúp tạo các biểu đồ phân phối trực quan, hỗ trợ việc phân tích thống kê các chỉ số cầu thủ trong giải đấu và so sánh giữa các đội.

## Bài 2.4:

**1. Mục tiêu**

Đoạn mã này thực hiện phân tích để xác định đội bóng nào có điểm số cao nhất ở mỗi chỉ số trong file results.csv. Từ đó, xác định đội có phong độ tốt nhất dựa trên số lần dẫn đầu các chỉ số và đếm tần suất mỗi đội đạt được vị trí cao nhất ở từng chỉ số.

**2. Các bước triển khai**

- **Đọc dữ liệu**: Sử dụng pd.read_csv() để đọc dữ liệu từ file results.csv và lưu vào DataFrame df, với dấu phân cách là ;.
- **Xác định các cột chỉ số**: Giả định các cột chứa chỉ số bắt đầu từ cột thứ ba, do đó df.columns[2:] được sử dụng để lấy danh sách cột chỉ số (điều chỉnh nếu vị trí cột khác nhau).

- **Xác định đội dẫn đầu ở từng chỉ số**:
    - Tạo từ điển top_teams để lưu đội dẫn đầu cho từng chỉ số.
    - Với mỗi cột chỉ số (stats_columns), tìm giá trị lớn nhất bằng cách:
        - Dùng df[col].idxmax() để lấy chỉ số hàng có giá trị lớn nhất trong cột col.
        - Lấy tên đội tương ứng (df.loc[...] với cột 'Squad') và thêm vào từ điển top_teams.

```
for col in stats_columns:
    # Tìm hàng có giá trị lớn nhất trong cột, lấy tên đội từ cột 'Squad'
    top_team = df.loc[df[col].idxmax(), 'Squad']
    top_teams[col] = top_team
```

- **Tính số lần dẫn đầu của mỗi đội**:
    - Dùng Counter từ thư viện collections để đếm số lần mỗi đội dẫn đầu các chỉ số.
    - Lưu kết quả vào team_performance và tìm đội có phong độ tốt nhất dựa trên tần suất cao nhất (most_common(1)).

```
# Đếm tần suất xuất hiện của mỗi đội trong danh sách các chỉ số dẫn đầu
team_performance = Counter(top_teams.values())

# Tìm đội có phong độ tốt nhất dựa trên số lần đứng đầu các chỉ số
best_team = team_performance.most_common(1)[0]
print("\nĐội có phong độ tốt nhất:")
print(f"{best_team[0]} với {best_team[1]} lần dẫn đầu các chỉ số")
```

- **In kết quả**:
    - In tên đội dẫn đầu từng chỉ số từ từ điển top_teams.
    - In đội có phong độ tốt nhất (đội xuất hiện nhiều lần nhất) và số lần dẫn đầu.
    - In chi tiết tần suất số lần dẫn đầu của từng đội.

### 3. Kết quả

Kết quả đầu ra gồm:

- Đội có điểm số cao nhất ở từng chỉ số, cho phép so sánh và đánh giá hiệu suất các chỉ số khác nhau.
- Đội có phong độ tốt nhất dựa trên số lần dẫn đầu các chỉ số, giúp xác định đội bóng nổi bật nhất trong giải đấu.
- Số lần mỗi đội dẫn đầu từng chỉ số, giúp làm rõ mức độ cạnh tranh giữa các đội trong các chỉ số khác nhau.

**4. Tổng kết**

Đoạn mã giúp xác định đội bóng có phong độ nổi bật dựa trên các chỉ số khác nhau và cho thấy tần suất các đội dẫn đầu, hỗ trợ đánh giá toàn diện và so sánh hiệu suất đội bóng trong giải đấu.