BÁO CÁO BTL PYTHON

Câu 1: Thu thập dữ liệu cầu thủ:

Trình bày code:

1. Truy cập trang web và lấy bảng dữ liệu: Đoạn mã này truy cập trang web, tìm bảng chứa các câu lạc bộ và lưu các link href của từng câu lạc bộ vào danh sách.

```
# bắt đầu lấy dữ liệu web
link='https://fbref.com/en/comps/9/2023-2024/2023-2024-Premier-League-Stats'
r=requests.get(link)
soup=bs(r.content, 'html.parser')
# tìm bảng chứa các câu lạc bộ
soup=soup.find('table',{'id':'stats_squads_standard_for'})
# lưu các link href câu lạc bộ vòa danh sách
club=soup.find_all('th',{'scope':'row'})
```

2. Lấy dữ liệu của từng câu lạc bộ: Với mỗi link câu lạc bộ, đoạn mã này tiếp tục truy cập vào trang chi tiết của câu lạc bộ đó và tìm các bảng liên quan đến các thống kê khác nhau như Goalkeeping, Shooting, Passing, v.v.

```
# bắt đầu lấy dữ liệu của từng CLB
for item in club:
    # hàm chờ thời gian
    time.sleep(random.uniform(3,4))
    url='https://fbref.com' + item.a.get('href')
    tem=item.a.text.strip()
    print(tem)
    response = requests.get(url)
```

Tìm đến các bảng dữ liệu liên quan trong câu lạc bộ đó:

```
# tìm các bảng liên quan
soup=bs(response.text,'html.parser')
table=soup.find("table",{"id":"stats_standard_9"})
Goalkeeping  = soup.find("table", {"id": "stats_keeper_9"})
Shooting= soup.find("table",{"id":"stats_shooting_9"})
Passing=soup.find("table",{"id":"stats_passing_9"})
Pass_Types=soup.find("table",{"id":"stats_passing_types_9"})
Goal_and_Shot_Creation=soup.find("table",{"id":"stats_gca_9"})
Defensive_Actions=soup.find("table",{"id":"stats_defense_9"})
Possession=soup.find("table",{"id":"stats_possession_9"})
Playing_Time=soup.find("table",{"id":"stats_playing_time_9"})
Miscellaneous_Stats=soup.find("table",{"id":"stats_misc_9"})
```

3. Thu thập và xử lý dữ liệu: Đối với mỗi bảng, đoạn mã sử dụng hàm **get_stat(dât_stat , tbody)** để lấy giá trị của các ô trong bảng và lưu chúng vào dictionary **data_play**. Chỉ những cầu thủ có thời gian thi đấu trên 90 phút mới được thêm vào danh sách dữ liệu.

   Hàm lấy dữ liệu các ô:

```
def get_stat(data_stat,tbody):
    if data_stat=="nationality":
        cell = tbody.find("td", {"data-stat": data_stat})
        return cell.text.strip()[-3:] if cell else "N/a"

    elif data_stat=="player":
        cell=tbody.find("th")
        return cell.text.strip() if cell else "N/a"

    else:
        cell = tbody.find("td", {"data-stat": data_stat})
        return cell.text.strip() if cell and cell.text.strip() != "" else "N/a"
```

   Khi lấy dữ liệu sẽ cho vào từ điển **data_play{ }** và cập nhập giá trị các bảng khác vào từ điển

```
data_play={
    "name": get_stat("player", tbody),
    "Nation": get_stat("nationality", tbody),
    "Team": tem,
    "Position": get_stat("position", tbody),
    "Age": get_stat("age", tbody),

    # Playing time
    "Matches": get_stat("games", tbody),
    "Starts": get_stat("games_starts", tbody),
    "minutes_90s": get_stat("minutes_90s", tbody),

    # Performance
    "Non-Penalty Goals": get_stat("goals", tbody),
    "Penalty Goals": get_stat("pens_made", tbody),
    "Assists": get_stat("assists", tbody),
    "Yellow Cards": get_stat("cards_yellow", tbody),
    "Red Cards": get_stat("cards_red", tbody),

    # Expected
    "xG(Ex)": get_stat("xg", tbody),
    "npxG(Ex)": get_stat("npxg", tbody),
    "xAG(Ex)": get_stat("xg_assist", tbody),

    # Progression
```

4. Sắp xếp và lưu dữ liệu: Cuối cùng, đoạn mã sắp xếp dữ liệu theo tên cầu thủ và tuổi từ lớn đến nhỏ trước khi lưu vào file CSV result.csv.

```
# Lưu DataFrame vào file CSV
df = pd.DataFrame(data)
df = df.sort_values(by=["name", "Age"], ascending=[True, False])
df.to_csv('result.csv', index=False,encoding='utf-8-sig')  # Lưu vào file result.csv
print("Dữ liệu đã được lưu vào file result.csv.")
print(df)
```

Kết quả:

```
Nott'ham Forest
đang lấy dữ liệu cầu thủ trong bảng: Nott'ham Forest
Sheffield Utd
đang lấy dữ liệu cầu thủ trong bảng: Sheffield Utd
Tottenham
đang lấy dữ liệu cầu thủ trong bảng: Tottenham
West Ham
đang lấy dữ liệu cầu thủ trong bảng: West Ham
Wolves
đang lấy dữ liệu cầu thủ trong bảng: Wolves
Dữ liệu đã được lưu vào file result.csv.
                 name Nation           Team Position Age Matches Starts minutes_90s Non-Penalty Goals Penalty Goals  ... onxGA Fls ^Fld ^Off Crs OG Recov Won ^Lost   Won%
461    Aaron Cresswell    ENG       West Ham    DF,FW  33      11     4         4.8                0             0  ...   5.6   2    3    1  11  0    18   6     3   66.7
87       Aaron Hickey    SCO      Brentford       DF  21       9     9         7.9                0             0  ...  10.3  10   16    0   5  0    42   1     9   10.0
17      Aaron Ramsdale    ENG        Arsenal       GK  25       6     6         6.0                0             0  ...   5.8   0    1    0   0  0     6   0     0    N/a
145      Aaron Ramsey    ENG        Burnley    MF,FW  20      14     5         5.9                0             0  ...  11.5   7    3    1   2  0    24   4     5   44.4
319  Aaron Wan-Bissaka    ENG  Manchester Utd       DF  25      22    20        19.8                0             0  ...  37.9  17    9    0  14  0    94  21    19   52.5
..                 ...    ...            ...      ...  ..     ...   ...         ...              ...           ...  ...   ...  ..  ...  ...  .. ..   ...  ..   ...    ...
429      Yves Bissouma    MLI       Tottenham       MF  26      28    26        23.0                0             0  ...  37.8  35   31    0   1  0   128  14    15   48.3
126       Zeki Amdouni    SUI        Burnley       FW  22      34    27        21.7                5             1  ...  36.2  22   39    2   8  0    79  15    56   21.1
35         Álex Moreno    ESP    Aston Villa       DF  30      21    11        11.5                2             0  ...  18.2  11   16    2  37  1    40   4    10   28.6
158     Đorđe Petrović    SRB        Chelsea       GK  23      23    22        22.1                0             0  ...  34.3   0    1    0   0  0    17   5     0  100.0
457    Łukasz Fabiański    POL       West Ham       GK  38      10     7         8.0                0             0  ...  16.8   0    0    0   0  0     3   2     0  100.0

[491 rows x 171 columns]
PS C:\Users\ADMIN\Desktop\python code\beautifulsoup>
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	name	Nation	Team	Position	Age	Matches	Starts	minutes_9	Non-Penal	Penalty Go	Assists	Yellow Car	Red Cards	xG(Ex)	npxG(Ex)	xAG(E
2	Aaron Cres	ENG	West Ham	DF,FW	33	11	4	4.8	0	0	0	1	0	0	0	
3	Aaron Hick	SCO	Brentford	DF	21	9	9	7.9	0	0	0	5	0	0.2	0.2	
4	Aaron Ram	ENG	Arsenal	GK	25	6	6	6	0	0	0	0	0	0	0	
5	Aaron Ram	ENG	Burnley	MF,FW	20	14	5	5.9	0	0	0	1	0	0.3	0.3	
6	Aaron War	ENG	Manchester	DF	25	22	20	19.8	0	0	2	4	0	0.1	0.1	
7	Abdoulaye	MLI	Everton	FW,MF	30	32	32	29.2	7	0	1	7	0	8.8	8.8	
8	Adam Lalla	ENG	Brighton	MF,FW	35	25	13	9.4	0	0	1	2	0	0.8	0.8	
9	Adam Smit	ENG	Bournemo	DF	32	28	25	23.9	0	0	2	6	0	0.1	0.1	
10	Adam Web	ENG	Brighton	DF	28	15	13	12.7	0	0	0	2	0	0.4	0.4	
11	Adam Wha	ENG	Crystal Pal	MF	19	16	15	14.4	0	0	3	2	0	0.3	0.3	
12	Adama Tra	ESP	Fulham	FW,MF	27	17	1	4.2	2	0	3	2	0	1.5	1.5	
13	Albert Sam	BEL	Luton Tow	MF	23	17	16	14.5	1	0	3	4	0	0.6	0.6	
14	Alejandro (	ARG	Manchester	FW	19	36	30	28.5	7	0	4	4	0	8.4	8.3	
15	Alex Iwobi	NGA	Everton	MF	27	2	2	1.6	0	0	0	0	0	0.3	0.3	
16	Alex Iwobi	NGA	Fulham	FW,MF	27	30	25	24.4	5	0	2	2	0	5.3	5.3	
17	Alex Scott	ENG	Bournemo	MF	19	23	11	11.3	1	0	1	3	0	0.7	0.7	
18	Alexander	SWE	Newcastle	FW	23	30	27	25.1	21	5	2	1	0	20.3	15.6	
19	Alexis Mac	ARG	Liverpool	MF	24	33	31	28.9	5	1	5	7	1	3.7	2.9	
20	Alfie Dougl	ENG	Luton Tow	DF	23	37	34	32.5	2	0	8	5	0	1.3	1.3	

Câu 2:

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

1. Đọc dữ liệu từ file CSV result.csv với mã hóa utf-8-sig

```
import pandas as pd

# Đọc dữ liệu từ file CSV
df = pd.read_csv('result.csv', encoding='utf-8-sig')
```

2. Lọc ra các cột có chỉ số bằng cách loại bỏ các cột 'name', 'Nation', 'Team', 'Position'. Chuyển đổi các cột này về kiểu số. Nếu có lỗi, giá trị sẽ là NaN.

```
# Lọc ra các cột có chỉ số (loại bỏ cột không cần thiết như 'name', 'Nati
numeric_cols = df.columns.drop(['name', 'Nation', 'Team', 'Position'])
# Chuyển đổi các cột về kiểu số nếu có thể
for col in numeric_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')
```

3. Tạo hai dictionary top_3_highest và top_3_lowest để lưu danh sách top 3 cầu thủ có điểm cao nhất và thấp nhất ở mỗi chỉ số. Dùng df.nlargest(3, attr) để tìm 3 giá trị lớn nhất và df.nsmallest(3, attr) để tìm 3 giá trị nhỏ nhất ở cột attr. Sử dụng apply và lambda để tạo chuỗi ký tự dạng "tên cầu thủ: giá trị chỉ số".

```
# Tìm top 3 cầu thủ có điểm cao nhất và thấp nhất ở mỗi chỉ số
top_3_highest = {}
top_3_lowest = {}
for attr in numeric_cols:
    top_3_highest[attr] = df.nlargest(3, attr).apply(lambda x: f"{x['name']}: {x[attr]}", axis=1).tolist()
    top_3_lowest[attr] = df.nsmallest(3, attr).apply(lambda x: f"{x['name']}: {x[attr]}", axis=1).tolist()
```

4. Chuyển đổi hai dictionary top_3_highest và top_3_lowest thành DataFrame. Lưu dữ liệu vào hai file CSV result2_highest.csv và result2_lowest.csv.

```
# Chuyển đổi thành DataFrame để lưu vào file CSV
top_3_highest_df = pd.DataFrame.from_dict(top_3_highest, orient='index').transpose()
top_3_lowest_df = pd.DataFrame.from_dict(top_3_lowest, orient='index').transpose()

# Lưu dữ liệu vào file result2_highest.csv và result2_lowest.csv
top_3_highest_df.to_csv('result2_highest.csv', index=False, encoding='utf-8-sig')
top_3_lowest_df.to_csv('result2_lowest.csv', index=False, encoding='utf-8-sig')
```

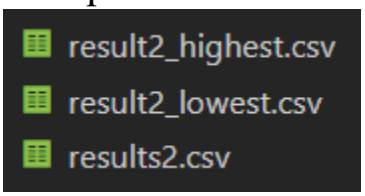
5. Tạo danh sách stats để lưu các giá trị thống kê. Tính trung vị (median), trung bình (mean) và độ lệch chuẩn (std) cho mỗi chỉ số trên toàn giải và cho từng đội.

```
# Tính trung vị, trung bình và độ lệch chuẩn cho mỗi chỉ số
stats = []
teams = df['Team'].unique()
for attr in numeric_cols:
    all_median = df[attr].median()
    all_mean = df[attr].mean()
    all_std = df[attr].std()

    stats.append({'Team': 'all', 'Attribute': attr, 'Median': all_median, 'Mean': all_mean, 'Std': all_std})

    for team in teams:
        team_data = df[df['Team'] == team]
        team_median = team_data[attr].median()
        team_mean = team_data[attr].mean()
        team_std = team_data[attr].std()

        stats.append({'Team': team, 'Attribute': attr, 'Median': team_median, 'Mean': team_mean, 'Std': team_std})
```

6. Chuyển đổi danh sách stats thành DataFrame. Dùng pivot để tổ chức lại DataFrame theo định dạng yêu cầu. Lưu kết quả vào file CSV results2.csv.

```
# Chuyển đổi thành DataFrame và lưu vào file CSV
stats_df = pd.DataFrame(stats)
stats_df_pivot = stats_df.pivot(index='Team', columns='Attribute')
stats_df_pivot.columns = [f'{stat} of {attr}' for attr, stat in stats_df_pivot.columns]
stats_df_pivot.reset_index(inplace=True)
stats_df_pivot.to_csv('results2.csv', index=False, encoding='utf-8-sig')

print("Dữ liệu đã được lưu vào file results2.csv.")
```

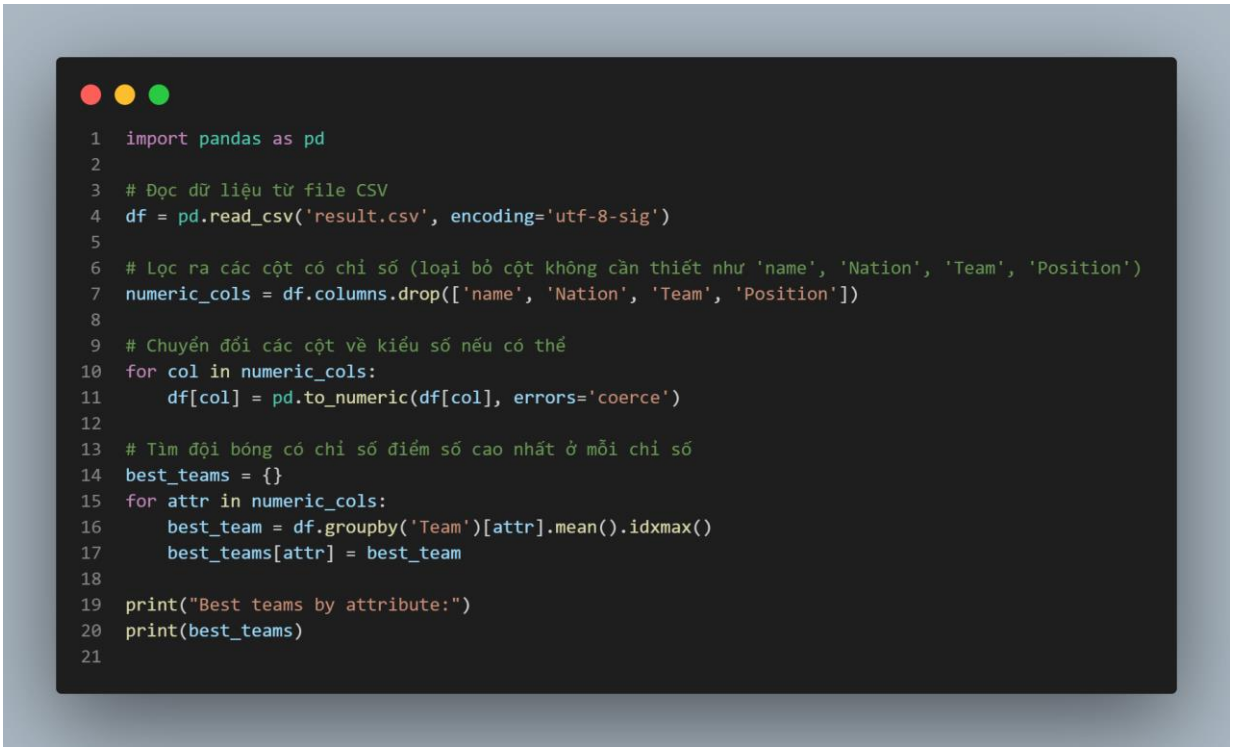
Kết quả đc lưu ở 3 file này:



- Vẽ historgram phân bố của mỗi chỉ số:

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

# Đọc dữ liệu từ file CSV
df = pd.read_csv('result.csv', encoding='utf-8-sig')

# Lọc ra các cột có chỉ số (loại bỏ cột không cần thiết như 'name', 'Nation', 'Team', 'Position')
numeric_cols = df.columns.drop(['name', 'Nation', 'Team', 'Position'])

# Chuyển đổi các cột về kiểu số nếu có thể
for col in numeric_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')

teams = df['Team'].unique()

# Vẽ histogram phân bố của mỗi chỉ số cho toàn giải và từng đội
for attr in numeric_cols:
    plt.figure(figsize=(10, 6))

    # Vẽ histogram cho toàn giải
    plt.hist(df[attr].dropna(), bins=15, alpha=0.5, label='All Teams')

    # Vẽ histogram cho từng đội
    for team in teams:
        team_data = df[df['Team'] == team]
        plt.hist(team_data[attr].dropna(), bins=15, alpha=0.5, label=team)

    plt.title(f'Histogram of {attr}')
    plt.xlabel(attr)
    plt.ylabel('Frequency')
    plt.legend()
    plt.show()
```

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



```python
import pandas as pd

# Đọc dữ liệu từ file CSV
df = pd.read_csv('result.csv', encoding='utf-8-sig')

# Lọc ra các cột có chỉ số (loại bỏ cột không cần thiết như 'name', 'Nation', 'Team', 'Position')
numeric_cols = df.columns.drop(['name', 'Nation', 'Team', 'Position'])

# Chuyển đổi các cột về kiểu số nếu có thể
for col in numeric_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Tìm đội bóng có chỉ số điểm số cao nhất ở mỗi chỉ số
best_teams = {}
for attr in numeric_cols:
    best_team = df.groupby('Team')[attr].mean().idxmax()
    best_teams[attr] = best_team

print("Best teams by attribute:")
print(best_teams)

```