Title: Football League Standings Analysis

Introduction: This project analyzes the standings of the Spanish La Liga by either fetching live data using an API or using a provided Excel file. It determines team qualifications for European competitions and identifies relegated teams based on the final standings.

Option 1: Steps and Approach

1) **API Key Authentication**

   a. Authenticated using RapidAPI headers. I was not able to use the AllSportsAPI (https://allsportsapi.com/) to fetch football data due to the associated costs.

   b. Headers used in requests:

```
headers = {
    "X-RapidAPI-Key": "c98[redacted]ffece66e0db567[redacted]",
    "X-RapidAPI-Host": "fre[redacted]-football-data.p.[redacted]
}
```

2) **Fetch Country Key for Spain**

   a. Endpoint: **football-get-all-countries**

   b. Extracted "ccode" for Spain ("ESP").

3) **Fetch League ID for Spanish La Liga**

   a. Endpoint: **football-get-all-leagues-with-countries**

   b. Filtered leagues in Spain (ccode="ESP") using "laliga" keyword.

```
countries_url = "https://free-api-live-football-data.p.rapidapi.com/football-get-all-countries"
countries_response = requests.get(countries_url, headers=headers).json()

# country code or country_key of Spain or, here we can use different country name which we want
spain_ccode = None
for country in countries_response["response"]["countries"]:
    if country["name"].lower() == "spain":
        spain_ccode = country["ccode"]
        break

print("Country key for Spain:", spain_ccode)

#-----------------

# api end point "football-get-all-leagues-with-countries" form Rapidapi
leagues_url = "https://free-api-live-football-data.p.rapidapi.com/football-get-all-leagues-with-countries"
leagues_response = requests.get(leagues_url, headers=headers).json()


def normalize(text):
    return text.lower().strip()

# Keywords to match LaLiga, LaLiga2...and also we can use Liga F
search_keywords = ["laliga"]

league_ids = []

for country in leagues_response["response"]["leagues"]:
    if country["ccode"] == "ESP":   # or we can use globally, if country["ccode"] == anycountry_ccode:
        for league in country["leagues"]:
            name = normalize(league.get("name", ""))
            localized_name = normalize(league.get("localizedName", ""))
            if any(keyword in name or keyword in localized_name for keyword in search_keywords):
                league_ids.append((league["id"], league["name"]))
                print(league["id"], league["name"])
        break

#----------------
```

4) **Fetch Standings Data**

a. For each league ID, fetched standings with endpoint **football-get-standing-all?leagueid=<id>**

b. Retrieved data for each team, including:

i. Matches played, wins, draws, losses.

ii. Goals scored, goal difference, points.

iii. League and team identifiers.

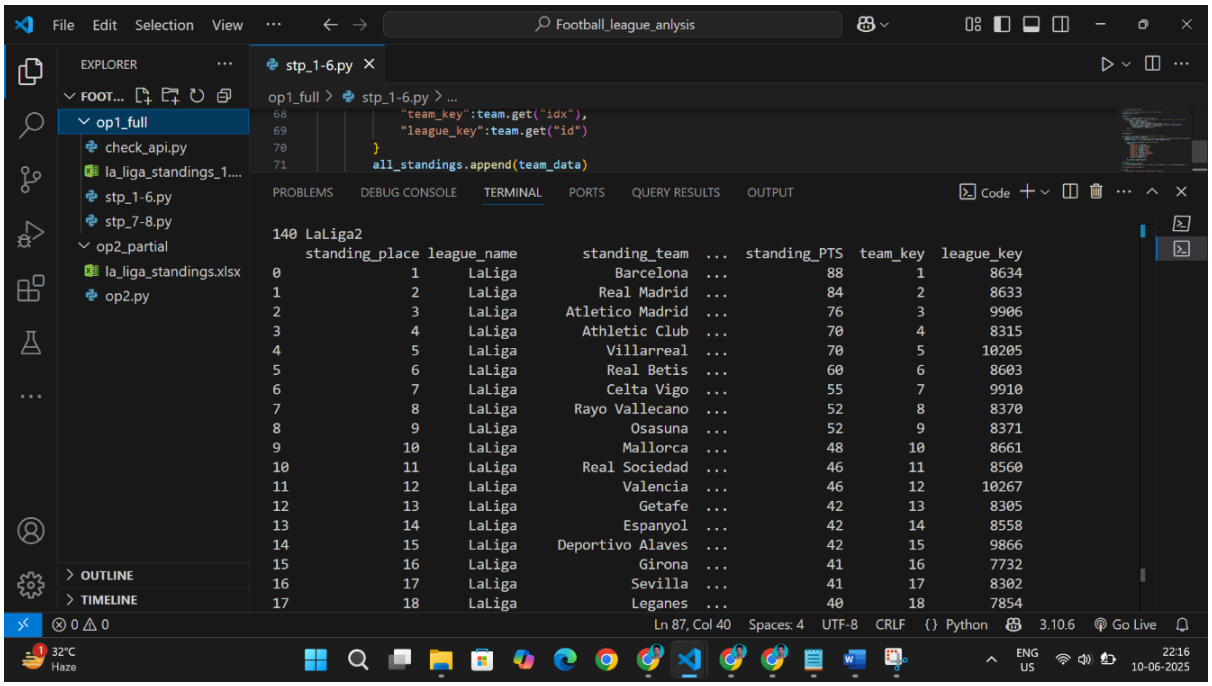
```
all_standings = []

# api end point "football-get-standing-all?leagueid=id" form Rapidapi
for league_id, league_name in league_ids:
    standings_url = f"https://free-api-live-football-data.p.rapidapi.com/football-get-standing-all?leagueid={league
    standings_response = requests.get(standings_url, headers=headers).json()

    for team in standings_response["response"]["standing"]:
        team_data = {
            "league_name": league_name,
            "standing_team": team.get("name"),
            "standing_P": team.get("played"),
            "standing_W": team.get("wins"),
            "standing_D": team.get("draws"),
            "standing_L": team.get("losses"),
            "standing_F": team.get("scoresStr"),
            "standing_GD": team.get("goalConDiff"),
            "standing_PTS": team.get("pts"),
            "team_key":team.get("idx"),
            "league_key":team.get("id")
        }
        all_standings.append(team_data)
```

API Data Limitations:

For the first option, I was not able to fetch data from the AllSportsAPI due to payment requirements. Instead, I used multiple endpoints from the free live football data API on RapidAPI. As a result, I could not fetch four data fields (standing_A, league_season, league_round, standing_updated) as required by the assessment task because these fields were not present in the JSON response.

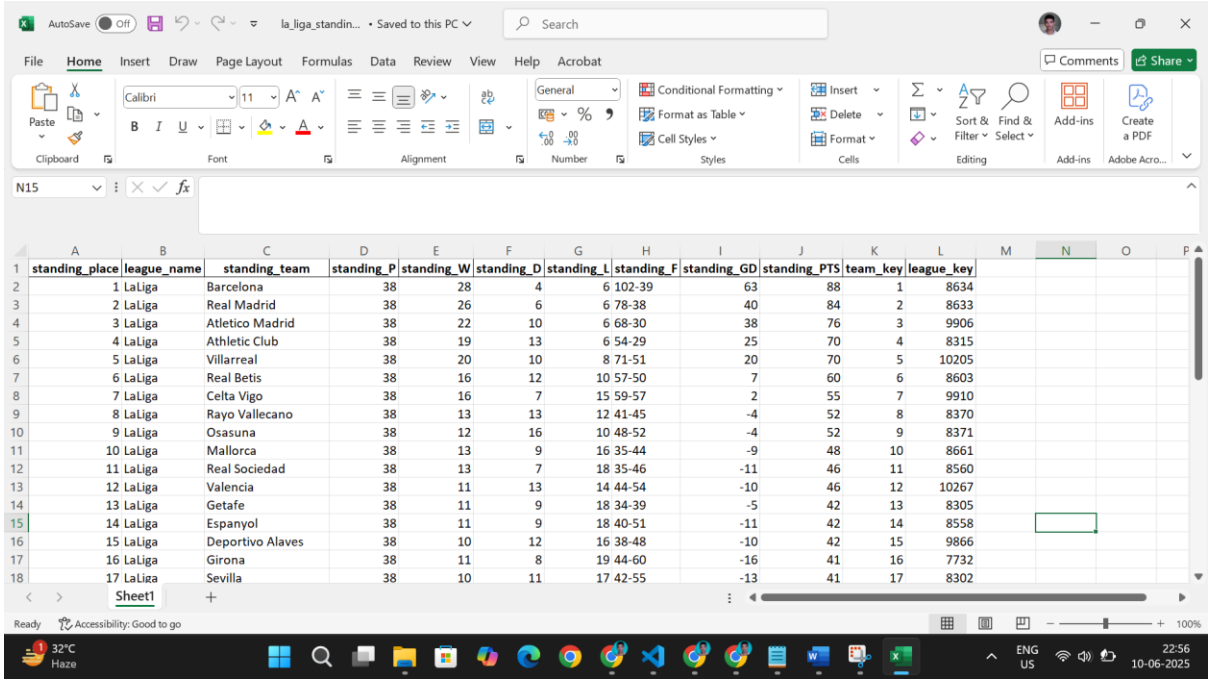
5) **Create DataFrame**

a. Created Pandas DataFrame with selected columns



6) **Save DataFrame to Excel**

a. File: la_liga_standings_1.xlsx.



standing_place	league_name	standing_team	standing_P	standing_W	standing_D	standing_L	standing_F	standing_GD	standing_PTS	team_key	league_key
1	LaLiga	Barcelona	38	28	4	6	102-39	63	88	1	8634
2	LaLiga	Real Madrid	38	26	6	6	78-38	40	84	2	8633
3	LaLiga	Atletico Madrid	38	22	10	6	68-30	38	76	3	9906
4	LaLiga	Athletic Club	38	19	13	6	54-29	25	70	4	8315
5	LaLiga	Villarreal	38	20	10	8	71-51	20	70	5	10205
6	LaLiga	Real Betis	38	16	12	10	57-50	7	60	6	8603
7	LaLiga	Celta Vigo	38	16	7	15	59-57	2	55	7	9910
8	LaLiga	Rayo Vallecano	38	13	13	12	41-45	-4	52	8	8370
9	LaLiga	Osasuna	38	12	16	10	48-52	-4	52	9	8371
10	LaLiga	Mallorca	38	13	9	16	35-44	-9	48	10	8661
11	LaLiga	Real Sociedad	38	13	7	18	35-46	-11	46	11	8560
12	LaLiga	Valencia	38	11	13	14	44-54	-10	46	12	10267
13	LaLiga	Getafe	38	11	9	18	34-39	-5	42	13	8305
14	LaLiga	Espanyol	38	11	9	18	40-51	-11	42	14	8558
15	LaLiga	Deportivo Alaves	38	10	12	16	38-48	-10	42	15	9866
16	LaLiga	Girona	38	11	8	19	44-60	-16	41	16	7732
17	LaLiga	Sevilla	38	10	11	17	42-55	-13	41	17	8302

7) **Analyze Standings**

a. Loaded the Excel file.

b. Created a new DataFrame identifying:

i. Top 4 teams → Champions League.

ii. 5th team → Europa League Group Stage.

iii. 6th team → Europa Conference League Qualifiers.

iv. Last 3 teams → Relegation.

8) **Return the Analysis**

a. Final DataFrame:

	Team	Points	Qualification
0	Barcelona	88	Champions League
1	Real Madrid	84	Champions League
2	Atletico Madrid	76	Champions League
3	Athletic Club	70	Champions League
4	Villarreal	70	Europa League Group Stage
5	Real Betis	60	Europa Conference League Qualifiers
6	Tenerife	36	Relegation
7	Racing de Ferrol	30	Relegation
8	Cartagena	23	Relegation

Option 2: Steps and Approach

Steps and Approach

1. **Use Provided Excel File**

   a) Loaded data from: **D:\project\Football_league_anlysis\la_liga_standings.xlsx**

   This File contains full standings for Spanish La Liga.

2. **Analyze Standings**

   a) Sorted teams by "standing_place".

   Top 4 teams → Champions League

   5th team → Europa League Group Stage

   6th team → Europa Conference League Qualifiers

   Last 3 teams → Relegation

3. **Return Analysis as DataFrame**

   a) Created final DataFrame

	Team	Points	Qualification
0	Barcelona	51	Champions League
1	Real Madrid	51	Champions League
2	Atl. Madrid	50	Champions League
3	Ath Bilbao	45	Champions League
4	Villarreal	41	Europa League Group Stage
5	Rayo Vallecano	35	Europa Conference League Qualifiers
6	Valencia	23	Relegation
7	Alaves	22	Relegation
8	Valladolid	15	Relegation