In [ ]:

```
import requests
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
from matplotlib.gridspec import GridSpec
```

In [ ]:

```
# API endpoint and data fetch
url = "https://www.thesportsdb.com/api/v1/json/3/lookuptable.php?l=4328&s=2023-2024"
response = requests.get(url)
data = response.json()
```

In [ ]:

```
# Data processing with type conversion
teams = []
for team in data['table']:
    teams.append({
        "Team": team["strTeam"],
        "Position": int(team["intRank"]),
        "Wins": int(team["intWin"]),
        "Draws": int(team["intDraw"]),
        "Losses": int(team["intLoss"]),
        "GoalsFor": int(team["intGoalsFor"]),
        "GoalsAgainst": int(team["intGoalsAgainst"]),
        "GoalDifference": int(team["intGoalDifference"]),
        "Points": int(team["intPoints"]),
    })
```

In [ ]:

```
df = pd.DataFrame(teams).sort_values(by="Points", ascending=False)
```

In [ ]:

```
df.to_csv("Premier_league.csv", index=False)
```

In [ ]:

```
def save_to_txt(filename, data_frame):
    with open(filename, 'w', encoding='utf-8') as f:
        f.write("PREMIER LEAGUE TEAM STATS 2023-2024\n")
        f.write("="*50 + "\n\n")
        for _, row in data_frame.iterrows():
            f.write(f"{row['Position']}. {row['Team']}\n")
            f.write(f"   Points: {row['Points']} | Record: {row['Wins']}W-{row['Draws']}D-{row['Losses']}L\n")
            f.write(f"   Goals: {row['GoalsFor']}F {row['GoalsAgainst']}A (Diff: {row['GoalDifference']})\n")
            f.write("-"*50 + "\n")
```

In [ ]:

```
save_to_txt("Premier_League2023-2024.txt", df)
```

In [ ]:

```
#dashboard creation
plt.figure(figsize=(18, 13))
plt.suptitle("Premier League 2023-2024 Season Analysis", fontsize=14, y=1.02)
```

In [ ]:

```
gs = GridSpec(3, 3, figure=plt.gcf())
```

In [ ]:

```
# Performance Scatter Plot
ax1 = plt.subplot(gs[0, 0])
sns.scatterplot(data=df, x='Wins', y='Losses', size='Points',
                hue='Position', sizes=(50, 300), palette='viridis_r', ax=ax1)
ax1.set_title("Team Performance: Wins vs Losses (Size=Points)")
ax1.legend(bbox_to_anchor=(1, 1))
```

In [ ]:

```
# 2. Goals Difference Bar Chart
ax2 = plt.subplot(gs[0, 1:])
sns.barplot(data=df.sort_values('GoalDifference', ascending=False),
            x='Team', y='GoalDifference', palette='coolwarm', ax=ax2)
ax2.set_title("Goal Difference by Team")
ax2.tick_params(axis='x', rotation=45)
ax2.axhline(0, color='black', linewidth=0.5)
```

In [ ]:

```
# 3. Wins-Draws-Losses Stacked Bar
ax3 = plt.subplot(gs[1, :])
df[['Team', 'Wins', 'Draws', 'Losses']].set_index('Team').plot(
    kind='bar', stacked=True, color=['#2ecc71', '#f39c12', '#e74c3c'], ax=ax3)
ax3.set_title("Wins-Draws-Losses Distribution")
ax3.tick_params(axis='x', rotation=45)
ax3.legend(title='Result')
```

In [ ]:

```
# 4. Points Distribution Pie Chart for the top 5 teams
ax4 = plt.subplot(gs[2, 0])
top_teams = df.head(5)
top_teams.set_index('Team')['Points'].plot.pie(
    autopct='%1.1f%%', startangle=90,
    wedgeprops={'linewidth': 1, 'edgecolor': 'white'},
    colors=sns.color_palette('pastel'), ax=ax4)
ax4.set_title("Points Distribution (Top 5 Teams)")
ax4.set_ylabel('')
```

In [ ]:

```
# 5. Position vs Goals Radar Chart
ax5 = plt.subplot(gs[2, 1:], polar=True)
categories = ['Wins', 'Draws', 'Losses', 'GoalsFor', 'GoalsAgainst']
N = len(categories)
angles = [n / float(N) * 2 * 3.14159 for n in range(N)]
angles += angles[:1]
```

In [ ]:

```
for i, row in df.head(3).iterrows():  # Plot top 3 teams
    values = row[categories].tolist()
    values += values[:1]
    ax5.plot(angles, values, linewidth=1, linestyle='solid',
             label=row['Team'])
    ax5.fill(angles, values, alpha=0.1)
```

In [ ]:

```
ax5.set_xticks(angles[:-1])
ax5.set_xticklabels(categories)
ax5.set_title("Top 3 Teams Performance Radar Chart")
ax5.legend(loc='upper right')
```

In [ ]:

```
plt.tight_layout()
plt.savefig('premier_league_visualization.png', dpi=300, bbox_inches='tight')
plt.show()
```

In [ ]:

```
# Open results
os.system('start Premier_League2023-2024.txt')
os.system('start premier_league_visualization.png')
input("Press Enter to close...")
```