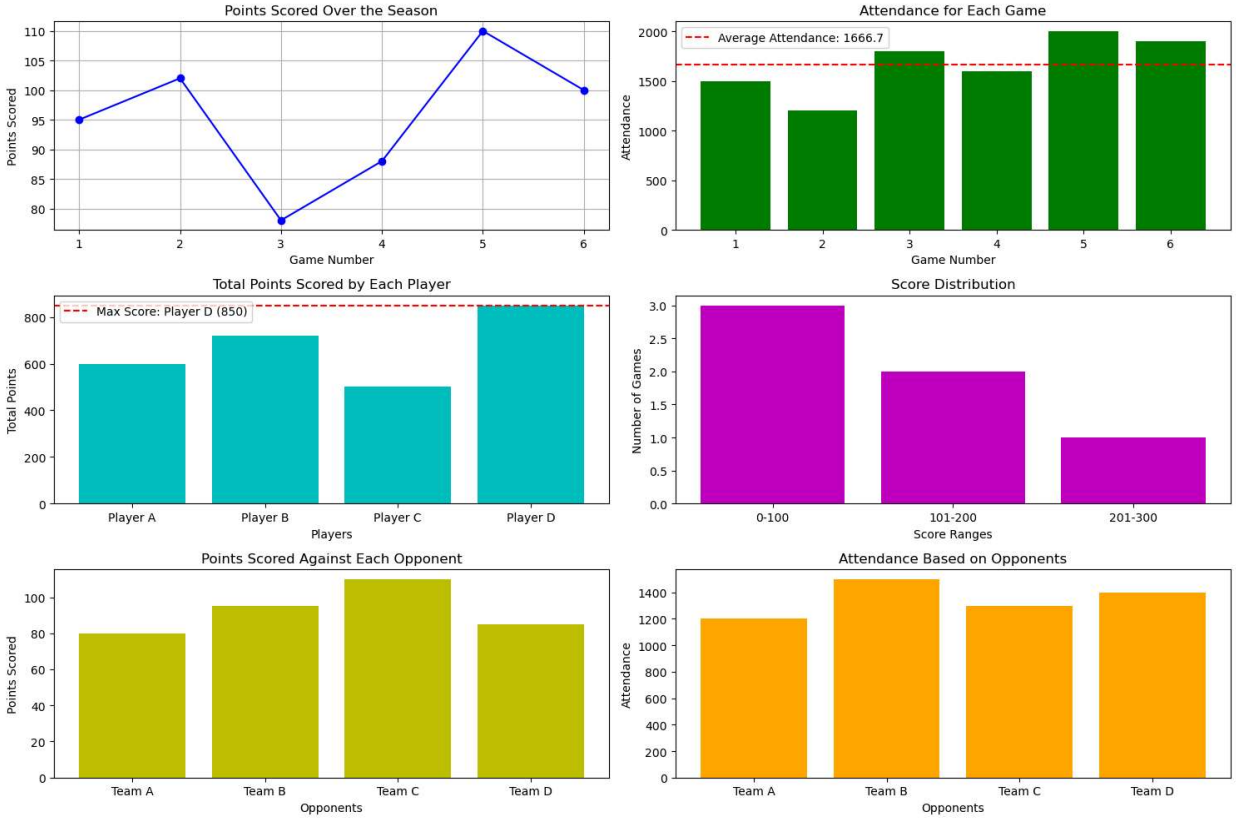
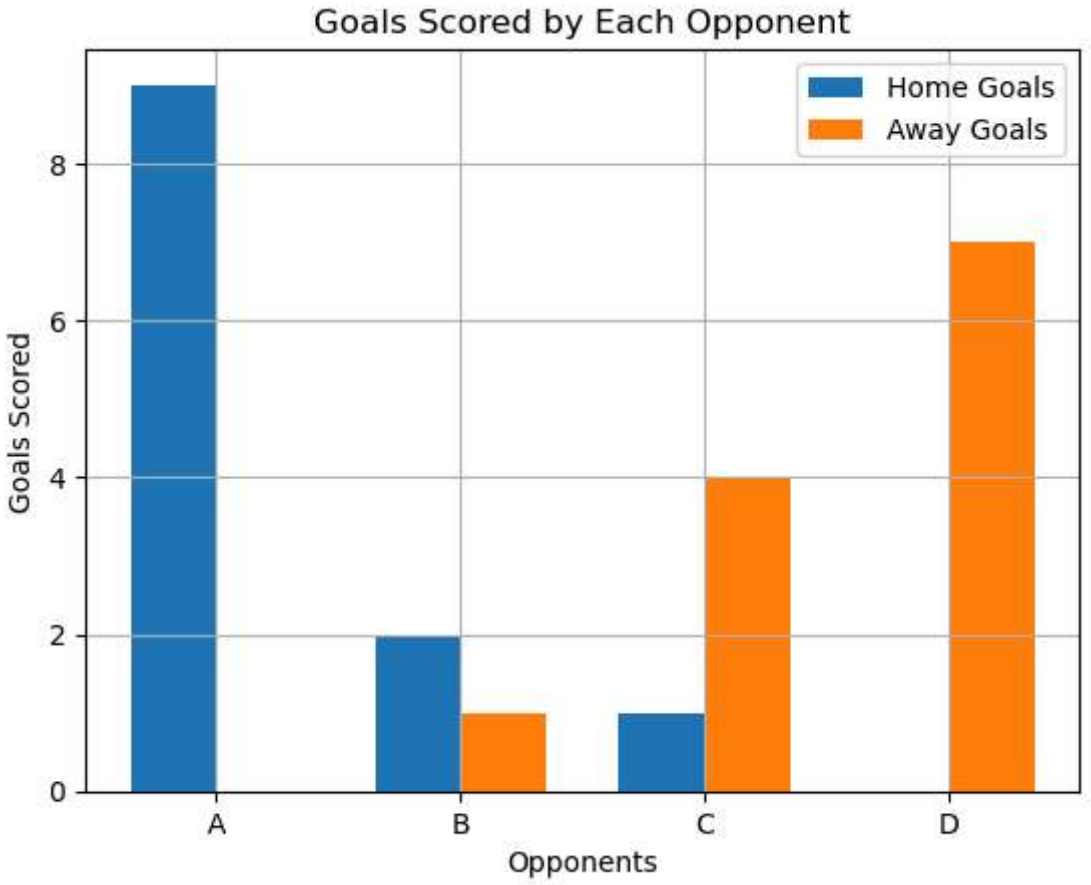
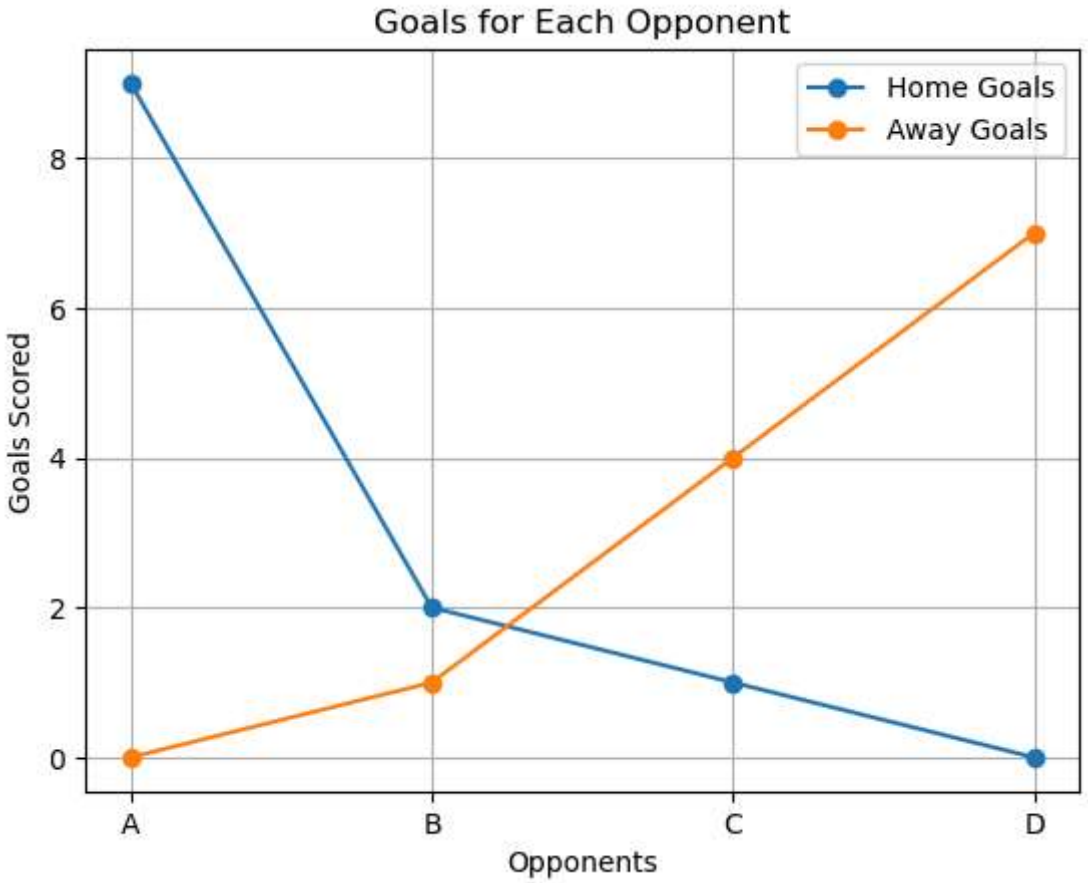
In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
g = np.array([1, 2, 3, 4, 5, 6])
gp = np.array([95, 102, 78, 88, 110, 100])
so = np.array([90, 88, 100, 85, 95, 100])
pa = np.array([1500, 1200, 1800, 1600, 2000, 1900])
plt.figure(figsize=(15, 10))
plt.subplot(3, 2, 1)
plt.plot(g, gp, marker='o', linestyle='-', color='b')
plt.title('Points Scored Over the Season')
plt.xlabel('Game Number')
plt.ylabel('Points Scored')
plt.xticks(g)
plt.grid()
plt.subplot(3, 2, 2)
plt.bar(g, pa, color='g')
plt.title('Attendance for Each Game')
plt.xlabel('Game Number')
plt.ylabel('Attendance')
plt.xticks(g)
average_attendance = np.mean(pa)
plt.axhline(average_attendance, color='r', linestyle='--', label=f'Average Attendance:
plt.legend()
players = ['Player A', 'Player B', 'Player C', 'Player D']
points = [600, 720, 500, 850]
max_score = max(points)
max_player = players[points.index(max_score)]
plt.subplot(3, 2, 3)
plt.bar(players, points, color='c')
plt.title('Total Points Scored by Each Player')
plt.xlabel('Players')
plt.ylabel('Total Points')
plt.axhline(max_score, color='r', linestyle='--', label=f'Max Score: {max_player} ({ma
plt.legend()
score_ranges = ['0-100', '101-200', '201-300']
range_counts = [3, 2, 1]
plt.subplot(3, 2, 4)
plt.bar(score_ranges, range_counts, color='m')
plt.title('Score Distribution')
plt.xlabel('Score Ranges')
plt.ylabel('Number of Games')
opponents = ['Team A', 'Team B', 'Team C', 'Team D']
opponent_scores = [80, 95, 110, 85]
plt.subplot(3, 2, 5)
plt.bar(opponents, opponent_scores, color='y')
plt.title('Points Scored Against Each Opponent')
plt.xlabel('Opponents')
plt.ylabel('Points Scored')
attendance_per_opponent = [1200, 1500, 1300, 1400]
plt.subplot(3, 2, 6)
plt.bar(opponents, attendance_per_opponent, color='orange')
plt.title('Attendance Based on Opponents')
plt.xlabel('Opponents')
plt.ylabel('Attendance')
plt.tight_layout()
plt.show()
```

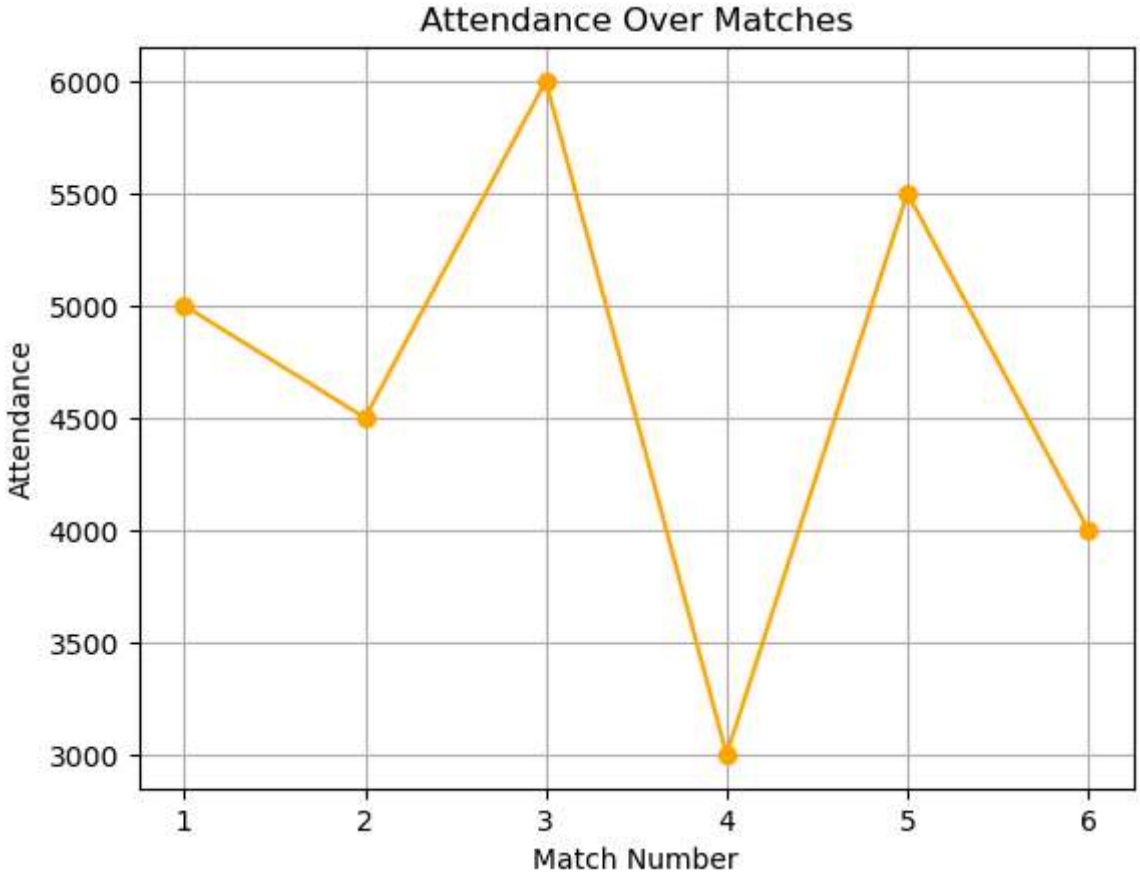


In [2]:
```
import matplotlib.pyplot as plt
import numpy as np

match = np.array((1, 2, 3, 4, 5, 6))
home_team = np.array(('A', 'C', 'A', 'B', 'A', 'B'))
away_team = np.array(('B', 'D', 'C', 'D', 'D', 'C'))
home_goals = np.array((2, 1, 4, 0, 3, 2))
away_goals = np.array((1, 3, 2, 1, 3, 2))
att = np.array((5000, 4500, 6000, 3000, 5500, 4000))
season = np.array((2023, 2023, 2023, 2023, 2023, 2023))

opponents = np.unique(np.concatenate((away_team, home_team)))
home_goals_by_opponent = [home_goals[home_team == team].sum() for team in opponents]
away_goals_by_opponent = [away_goals[away_team == team].sum() for team in opponents]

x = np.arange(len(opponents))
width = 0.35

plt.bar(x - width/2, home_goals_by_opponent, width, label='Home Goals')
plt.bar(x + width/2, away_goals_by_opponent, width, label='Away Goals')
plt.xlabel('Opponents')
plt.ylabel('Goals Scored')
plt.title('Goals Scored by Each Opponent')
plt.xticks(x, opponents)
plt.legend()
plt.grid()
plt.show()

plt.figure()
plt.plot(opponents, home_goals_by_opponent, marker='o', label='Home Goals')
plt.plot(opponents, away_goals_by_opponent, marker='o', label='Away Goals')
plt.xlabel('Opponents')
plt.ylabel('Goals Scored')
```

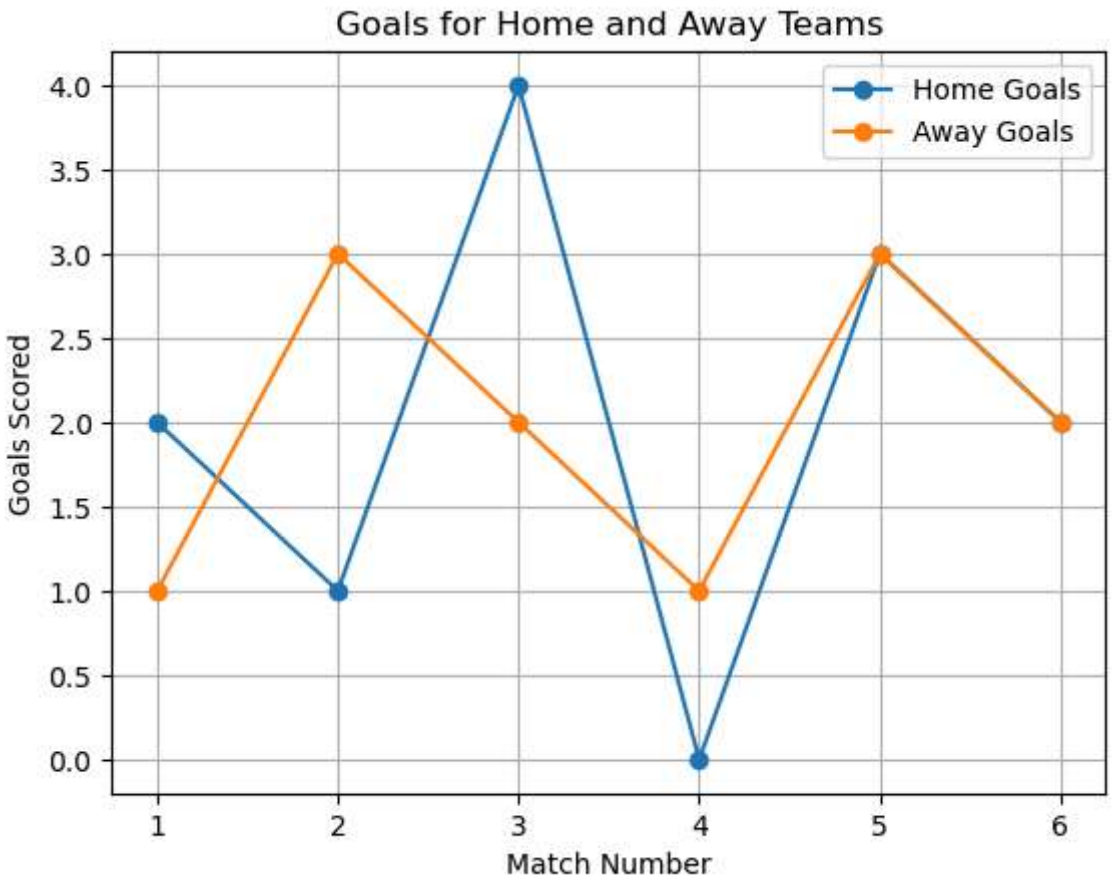
```
plt.title('Goals for Each Opponent')
plt.legend()
plt.grid()
plt.show()

plt.figure()
plt.plot(match, att, marker='o', color='orange')
plt.xlabel('Match Number')
plt.ylabel('Attendance')
plt.title('Attendance Over Matches')
plt.xticks(match)
plt.grid()
plt.show()

plt.figure()
plt.plot(match, home_goals, marker='o', label='Home Goals')
plt.plot(match, away_goals, marker='o', label='Away Goals')
plt.xlabel('Match Number')
plt.ylabel('Goals Scored')
plt.title('Goals for Home and Away Teams')
plt.xticks(match)
plt.legend()
plt.grid()

highest_home_goals = home_goals.max()
highest_away_goals = away_goals.max()
print(f"Highest Home Goals: {highest_home_goals}")
print(f"Highest Away Goals: {highest_away_goals}")

plt.show()
```








```
Highest Home Goals: 4
Highest Away Goals: 3
```


Goals for Home and Away Teams
Home Goals
Away Goals
Goals Scored
Match Number
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6

In [ ]: