

```
In [1]: '''
Question 1:
For the IPL dataset, answer the specified questions with summarization and e
libraries.
'''
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("matches1.csv")
df
```

Out[1]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	res
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	nor
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	nor
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	nor
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	nor
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	nor
...	...	...	...	...	...	...	...	...	...
631	632	2016	Raipur	2016-05-22	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore	field	nor
632	633	2016	Bangalore	2016-05-24	Gujarat Lions	Royal Challengers Bangalore	Royal Challengers Bangalore	field	nor
633	634	2016	Delhi	2016-05-25	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field	nor
634	635	2016	Delhi	2016-05-27	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad	field	nor
635	636	2016	Bangalore	2016-05-29	Sunrisers Hyderabad	Royal Challengers Bangalore	Sunrisers Hyderabad	bat	nor

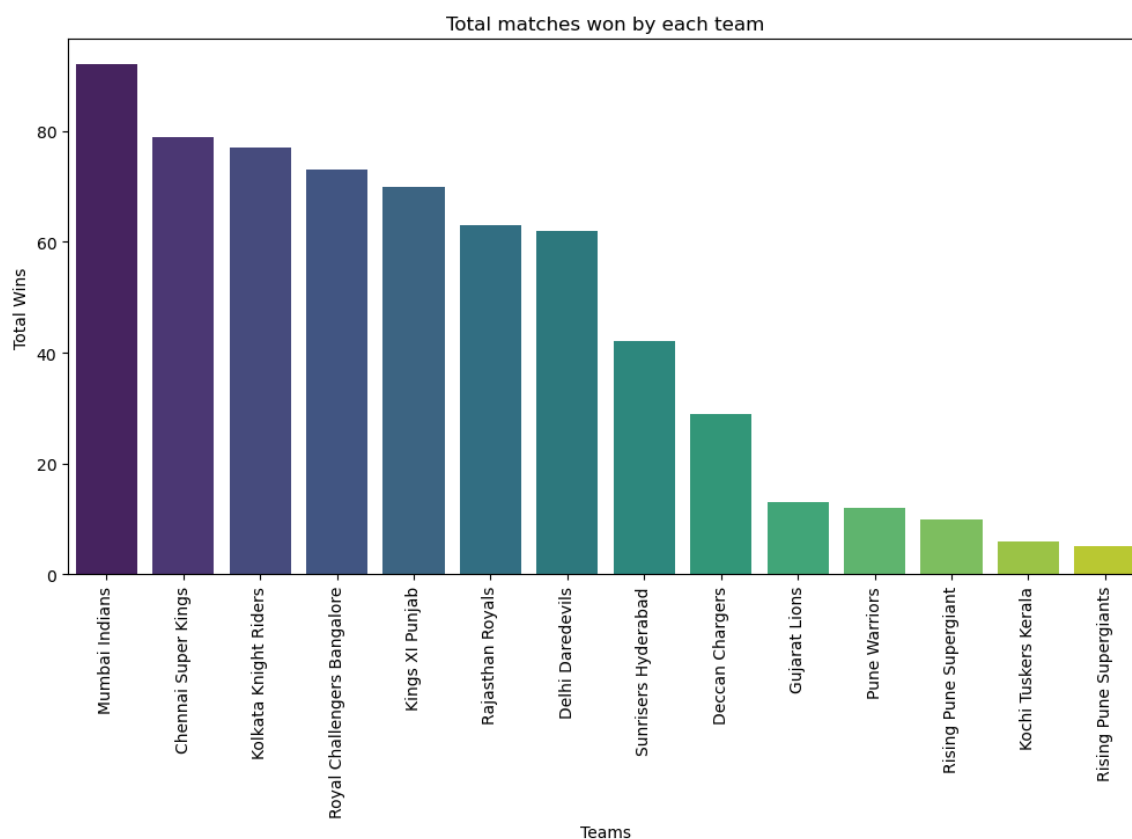
636 rows × 18 columns

```
In [7]: # 1. Find the names of teams that are part of IPL.
ipl_teams = set(df["team1"]).union(set(df["team2"]))
print("IPL Teams:\n")
for i in ipl_teams: print(i)
```

IPL Teams:

Deccan Chargers  
Rising Pune Supergiants  
Kochi Tuskers Kerala  
Rising Pune Supergiant  
Sunrisers Hyderabad  
Chennai Super Kings  
Mumbai Indians  
Rajasthan Royals  
Kings XI Punjab  
Kolkata Knight Riders  
Delhi Daredevils  
Gujarat Lions  
Royal Challengers Bangalore  
Pune Warriors

```
In [8]: # 2. Visualize the total number of matches won by each team.
team_wins = df["winner"].value_counts()
plt.figure(figsize=(12,6))
sns.barplot(x=team_wins.index, y=team_wins.values, palette="viridis")
plt.xticks(rotation=90)
plt.xlabel("Teams")
plt.ylabel("Total Wins")
plt.title("Total matches won by each team")
plt.show()
```



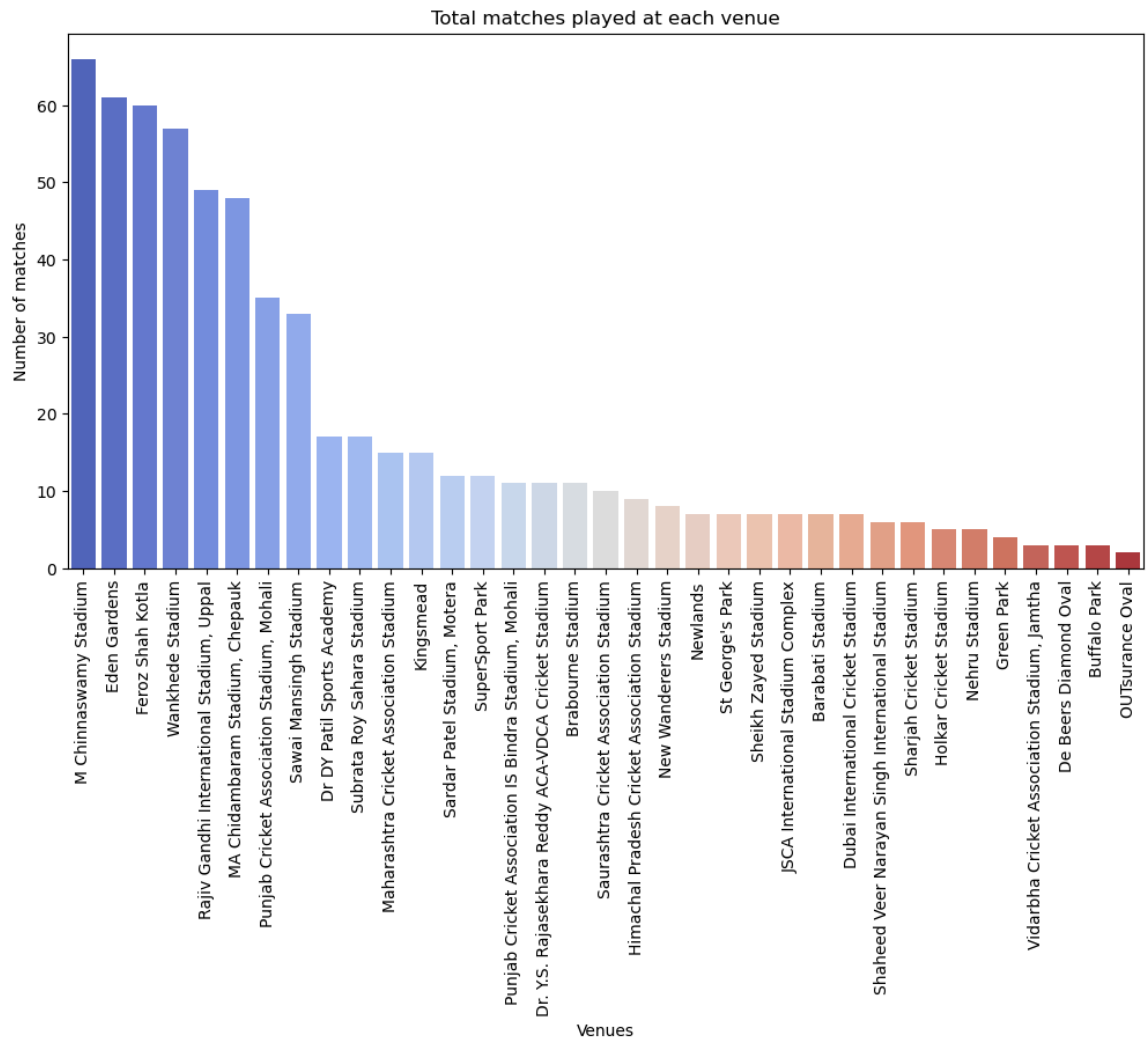
```
In [80]: # 3. List the names of umpires who have served as first umpire and the number of matches umpired
first_umpires = df["umpire1"].dropna().value_counts()
print(f"First umpires and the number of matches umpired:\n{first_umpires}")
```

First umpires and the number of matches umpired:

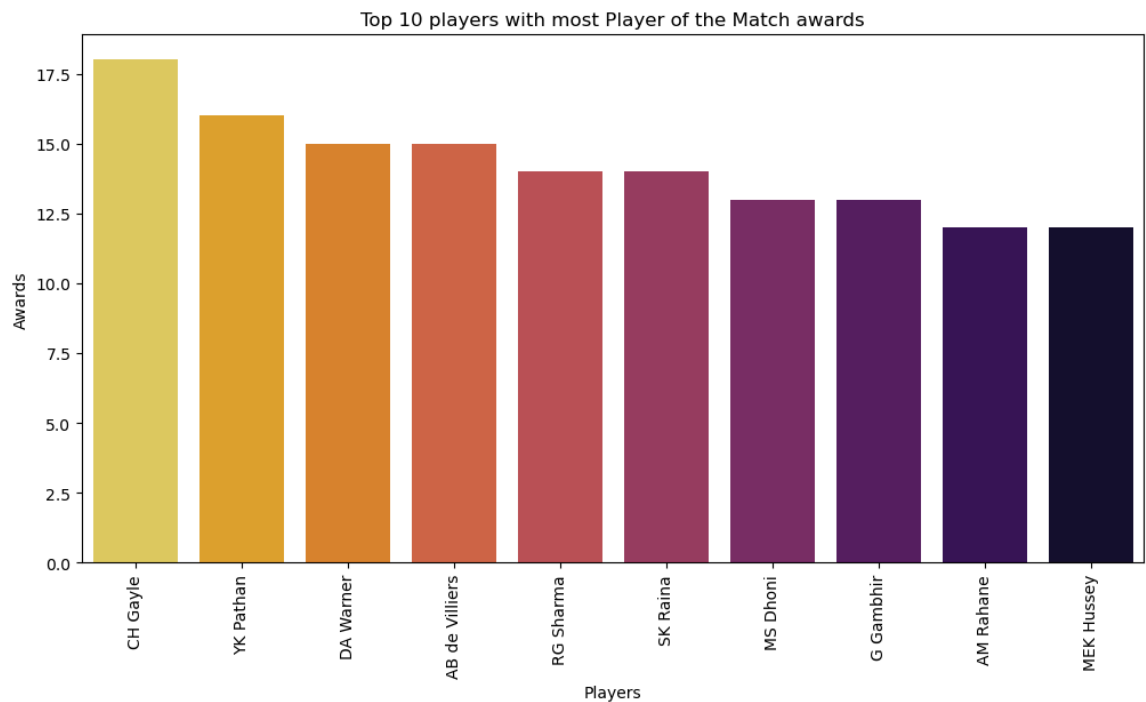
HDPK Dharmasena	73
Asad Rauf	51
AK Chaudhary	43
Aleem Dar	38
BF Bowden	37
S Ravi	36
BR Doctrove	34
M Erasmus	32
RE Koertzen	20
S Asnani	19
JD Cloete	16
CB Gaffaney	14
AY Dandekar	13
BG Jerling	13
NJ Llong	12
KN Ananthapadmanabhan	11
A Nand Kishore	11
VA Kulkarni	11
SS Hazare	11
Nitin Menon	11
RK Illingworth	11
K Hariharan	10
MR Benson	10
S Das	10
DJ Harper	9
RM Deshpande	7
GAV Baxter	7
IL Howell	7
BNJ Oxenford	7
C Shamsuddin	6
SJ Davis	6
SD Fry	5
SK Tarapore	5
CK Nandan	5
YC Barde	4
PR Reiffel	3
AM Saheba	3
A Deshmukh	3
K Bharatan	3
AV Jayaprakash	2
K Srinath	2
PG Pathak	2
SJA Taufel	1
SL Shastri	1

Name: umpire1, dtype: int64

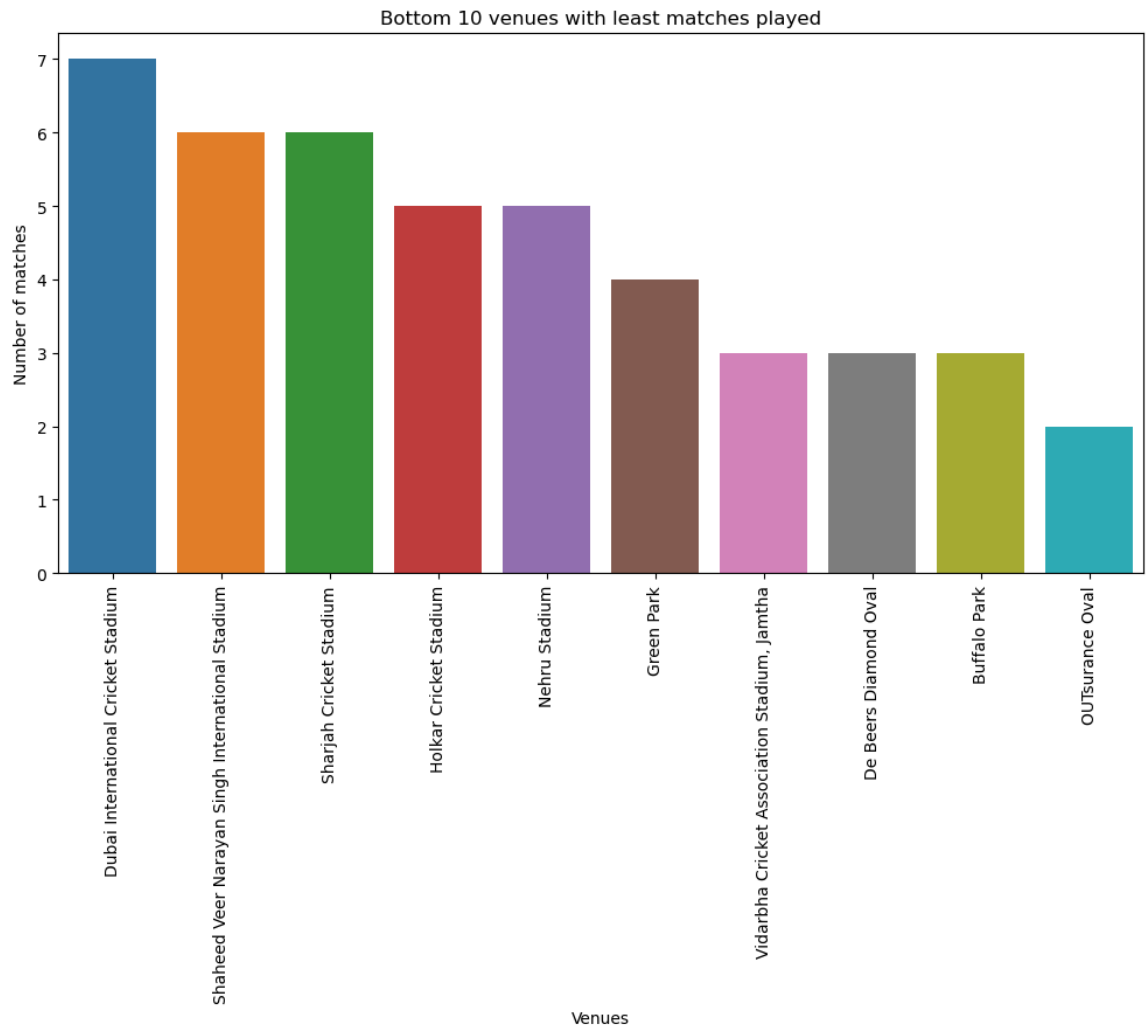
```
In [79]: # 4. Visualize how many matches were held in each venue.
venue_counts = df["venue"].value_counts()
plt.figure(figsize=(12,6))
sns.barplot(x=venue_counts.index, y=venue_counts.values, palette="coolwarm")
plt.xticks(rotation=90)
plt.xlabel("Venues")
plt.ylabel("Number of matches")
plt.title("Total matches played at each venue")
plt.show()
```



```
In [78]: import random
# 5. Visualize the top 10 players identified as the player of the match for
top_players = df["player_of_match"].value_counts().head(10)
plt.figure(figsize=(12,6))
palettes1=['Accent', 'Accent_r', 'Blues', 'Blues_r', 'BrBG', 'BrBG_r', 'BuGr
sns.barplot(x=top_players.index, y=top_players.values, palette=random.choice
plt.xticks(rotation=90)
plt.xlabel("Players")
plt.ylabel("Awards")
plt.title("Top 10 players with most Player of the Match awards")
plt.show()
```

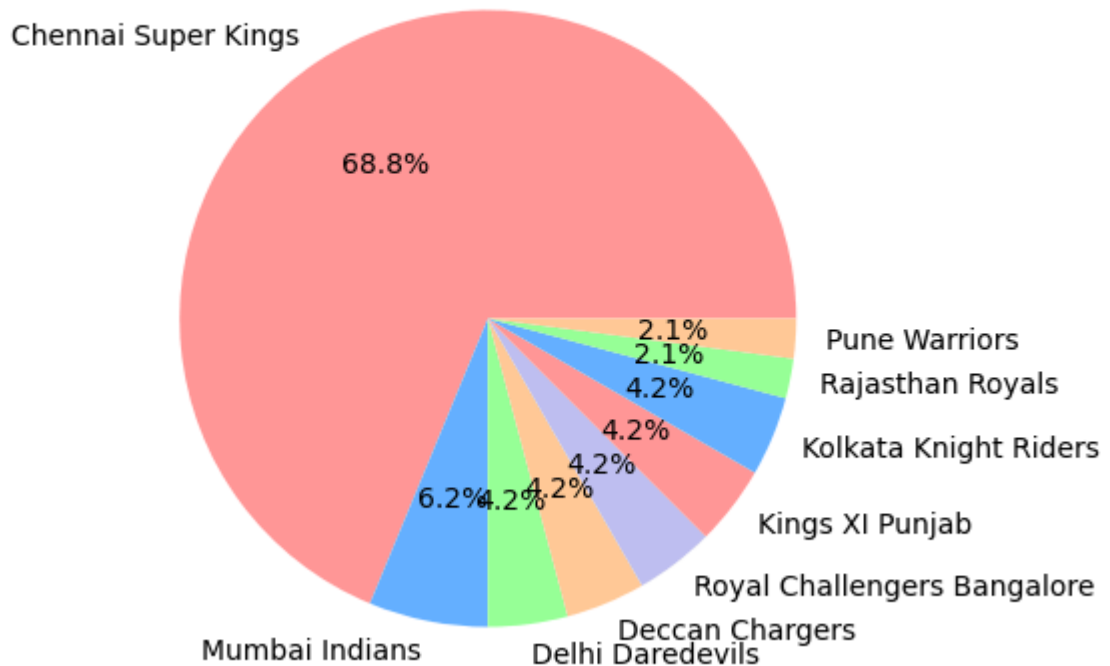


```
In [17]: # 6. Visualize the bottom 10 venues where the least number of matches were p
bottom_venues = venue_counts.tail(10)
plt.figure(figsize=(12,6))
sns.barplot(x=bottom_venues.index, y=bottom_venues.values)
plt.xticks(rotation=90)
plt.xlabel("Venues")
plt.ylabel("Number of matches")
plt.title("Bottom 10 venues with least matches played")
plt.show()
```



```
In [22]: # 7. Visualize the number of wins of each team for the venue of 'MA Chidambaram Stadium, Chepauk'
chidambaram_wins = df[df["venue"] == "MA Chidambaram Stadium, Chepauk"]["wins"]
plt.figure(figsize=(10, 5))
colors = ['#ff9999', '#66b3ff', '#99ff99', '#ffcc99', '#c2c2f0']
plt.pie(chidambaram_wins, labels=chidambaram_wins.index, autopct="%1.1f%%", colors=colors)
plt.title("Matches won at MA Chidambaram Stadium, Chepauk")
plt.show()
```

Matches won at MA Chidambaram Stadium, Chepauk



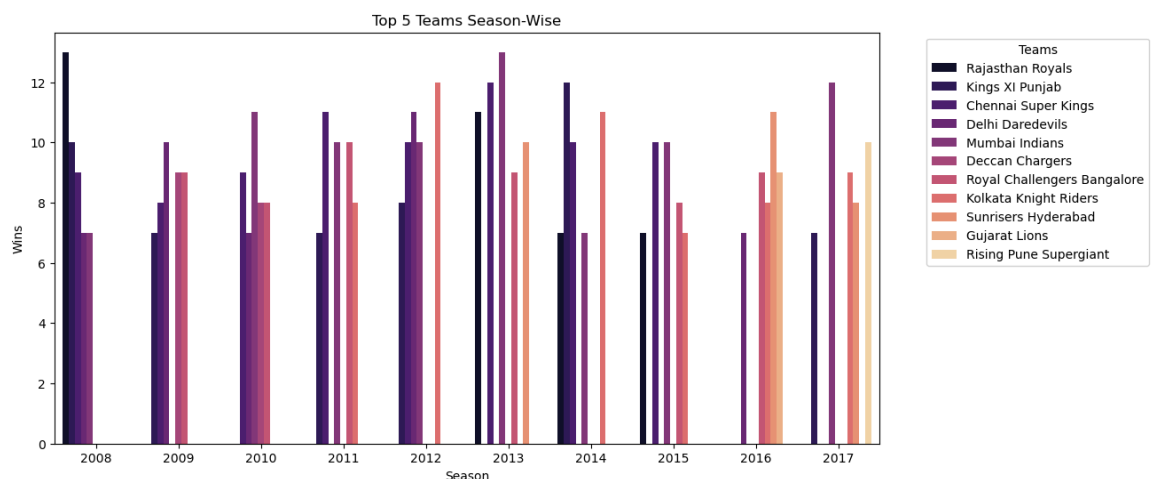
```
In [25]: # 8. Find and List the venues where the match resulted in a tie.
tie_venues = df[df["result"] == "tie"]["venue"].unique()
print("Venues where matches ended in a tie:\n")
for i in tie_venues: print(i)
```

Venues where matches ended in a tie:

Saurashtra Cricket Association Stadium  
 Newlands  
 MA Chidambaram Stadium, Chepauk  
 Rajiv Gandhi International Stadium, Uppal  
 M Chinnaswamy Stadium  
 Sheikh Zayed Stadium  
 Sardar Patel Stadium, Motera

```
In [74]: # 9. List and visualize the best teams(top 5), season-wise.
import matplotlib.pyplot as plt
import seaborn as sns
import random

# Group by season and winner, then count the number of wins per team in each
season_winners = df.groupby(["season", "winner"]).size().reset_index(name="wins")
# Sort values by season and wins in descending order to get the top teams per
season_winners_sorted = season_winners.sort_values(["season", "wins"], ascending=True)
# Select the top 5 teams per season
top_teams_season = season_winners_sorted.groupby("season").head(5)
# Visualize using seaborn barplot
plt.figure(figsize=(12, 6))
palettes=['Accent', 'Accent_r', 'Blues', 'Blues_r', 'BrBG', 'BrBG_r', 'BuGn', 'BuGn_r',
           'CM', 'CM_r', 'Dark2', 'Dark2_r', 'GnBu', 'GnBu_r', 'OrRd', 'OrRd_r', 'PRGn',
           'PRGn_r', 'Paired', 'Paired_r', 'PiYG', 'PiYG_r', 'PuBu', 'PuBu_r', 'PuBuGn',
           'PuBuGn_r', 'PuRd', 'PuRd_r', 'RdBu', 'RdBu_r', 'RdYl', 'RdYl_r', 'RdYlBu',
           'RdYlBu_r', 'Set1', 'Set1_r', 'Set2', 'Set2_r', 'Set3', 'Set3_r', 'Spectral',
           'Spectral_r', 'WPa1', 'WPa1_r', 'WPa2', 'WPa2_r', 'WPa3', 'WPa3_r', 'WPa4',
           'WPa4_r', 'WPa5', 'WPa5_r', 'WPa6', 'WPa6_r', 'WPa7', 'WPa7_r', 'WPa8',
           'WPa8_r', 'WPa9', 'WPa9_r', 'WPa10', 'WPa10_r', 'WPa11', 'WPa11_r', 'WPa12',
           'WPa12_r', 'WPa13', 'WPa13_r', 'WPa14', 'WPa14_r', 'WPa15', 'WPa15_r', 'WPa16',
           'WPa16_r', 'WPa17', 'WPa17_r', 'WPa18', 'WPa18_r', 'WPa19', 'WPa19_r', 'WPa20',
           'WPa20_r', 'WPa21', 'WPa21_r', 'WPa22', 'WPa22_r', 'WPa23', 'WPa23_r', 'WPa24',
           'WPa24_r', 'WPa25', 'WPa25_r', 'WPa26', 'WPa26_r', 'WPa27', 'WPa27_r', 'WPa28',
           'WPa28_r', 'WPa29', 'WPa29_r', 'WPa30', 'WPa30_r', 'WPa31', 'WPa31_r', 'WPa32',
           'WPa32_r', 'WPa33', 'WPa33_r', 'WPa34', 'WPa34_r', 'WPa35', 'WPa35_r', 'WPa36',
           'WPa36_r', 'WPa37', 'WPa37_r', 'WPa38', 'WPa38_r', 'WPa39', 'WPa39_r', 'WPa40',
           'WPa40_r', 'WPa41', 'WPa41_r', 'WPa42', 'WPa42_r', 'WPa43', 'WPa43_r', 'WPa44',
           'WPa44_r', 'WPa45', 'WPa45_r', 'WPa46', 'WPa46_r', 'WPa47', 'WPa47_r', 'WPa48',
           'WPa48_r', 'WPa49', 'WPa49_r', 'WPa50', 'WPa50_r', 'WPa51', 'WPa51_r', 'WPa52',
           'WPa52_r', 'WPa53', 'WPa53_r', 'WPa54', 'WPa54_r', 'WPa55', 'WPa55_r', 'WPa56',
           'WPa56_r', 'WPa57', 'WPa57_r', 'WPa58', 'WPa58_r', 'WPa59', 'WPa59_r', 'WPa60',
           'WPa60_r', 'WPa61', 'WPa61_r', 'WPa62', 'WPa62_r', 'WPa63', 'WPa63_r', 'WPa64',
           'WPa64_r', 'WPa65', 'WPa65_r', 'WPa66', 'WPa66_r', 'WPa67', 'WPa67_r', 'WPa68',
           'WPa68_r', 'WPa69', 'WPa69_r', 'WPa70', 'WPa70_r', 'WPa71', 'WPa71_r', 'WPa72',
           'WPa72_r', 'WPa73', 'WPa73_r', 'WPa74', 'WPa74_r', 'WPa75', 'WPa75_r', 'WPa76',
           'WPa76_r', 'WPa77', 'WPa77_r', 'WPa78', 'WPa78_r', 'WPa79', 'WPa79_r', 'WPa80',
           'WPa80_r', 'WPa81', 'WPa81_r', 'WPa82', 'WPa82_r', 'WPa83', 'WPa83_r', 'WPa84',
           'WPa84_r', 'WPa85', 'WPa85_r', 'WPa86', 'WPa86_r', 'WPa87', 'WPa87_r', 'WPa88',
           'WPa88_r', 'WPa89', 'WPa89_r', 'WPa90', 'WPa90_r', 'WPa91', 'WPa91_r', 'WPa92',
           'WPa92_r', 'WPa93', 'WPa93_r', 'WPa94', 'WPa94_r', 'WPa95', 'WPa95_r', 'WPa96',
           'WPa96_r', 'WPa97', 'WPa97_r', 'WPa98', 'WPa98_r', 'WPa99', 'WPa99_r', 'WPa100',
           'WPa100_r']
sns.barplot(x="season", y="wins", hue="winner", data=top_teams_season, palette=palettes)
plt.xlabel("Season")
plt.ylabel("Wins")
plt.title("Top 5 Teams Season-Wise")
plt.legend(title="Teams", bbox_to_anchor=(1.05, 1), loc="upper left")
plt.show()
```

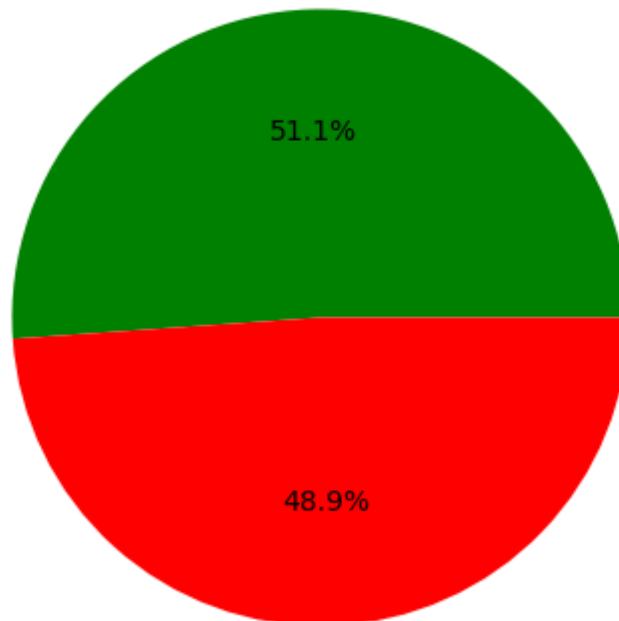




```
In [60]: # 10. Analyse and visualize the relationship between winning the toss and wi
toss_wins = df[df["toss_winner"] == df["winner"]].shape[0]
total_matches = df.shape[0]
toss_match_win_ratio = toss_wins / total_matches
plt.figure(figsize=(5, 5))
plt.pie([toss_match_win_ratio, 1 - toss_match_win_ratio], labels=["Toss Winr
plt.title("Correlation Between Toss Winning and Match Winning")
plt.show()
```

### Correlation Between Toss Winning and Match Winning

Toss Winner Won Match



Toss Winner Lost Match