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	norr
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	norr
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	norr
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	norr
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	norr
...	...	...	...	...	...	...	...	...	
631	632	2016	Raipur	2016-05-22	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore	field	norr
632	633	2016	Bangalore	2016-05-24	Gujarat Lions	Royal Challengers Bangalore	Royal Challengers Bangalore	field	norr
633	634	2016	Delhi	2016-05-25	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field	norr
634	635	2016	Delhi	2016-05-27	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad	field	norr
635	636	2016	Bangalore	2016-05-29	Sunrisers Hyderabad	Royal Challengers Bangalore	Sunrisers Hyderabad	bat	norr

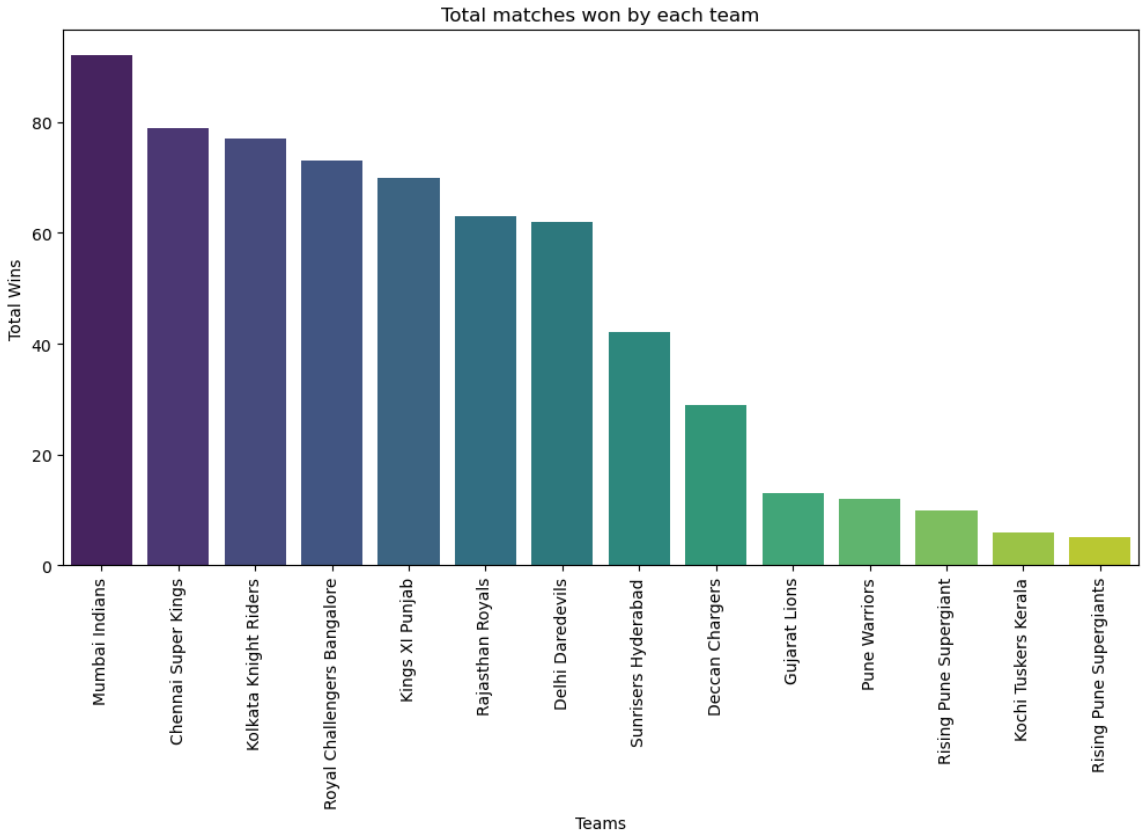
636 rows × 18 columns

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
```

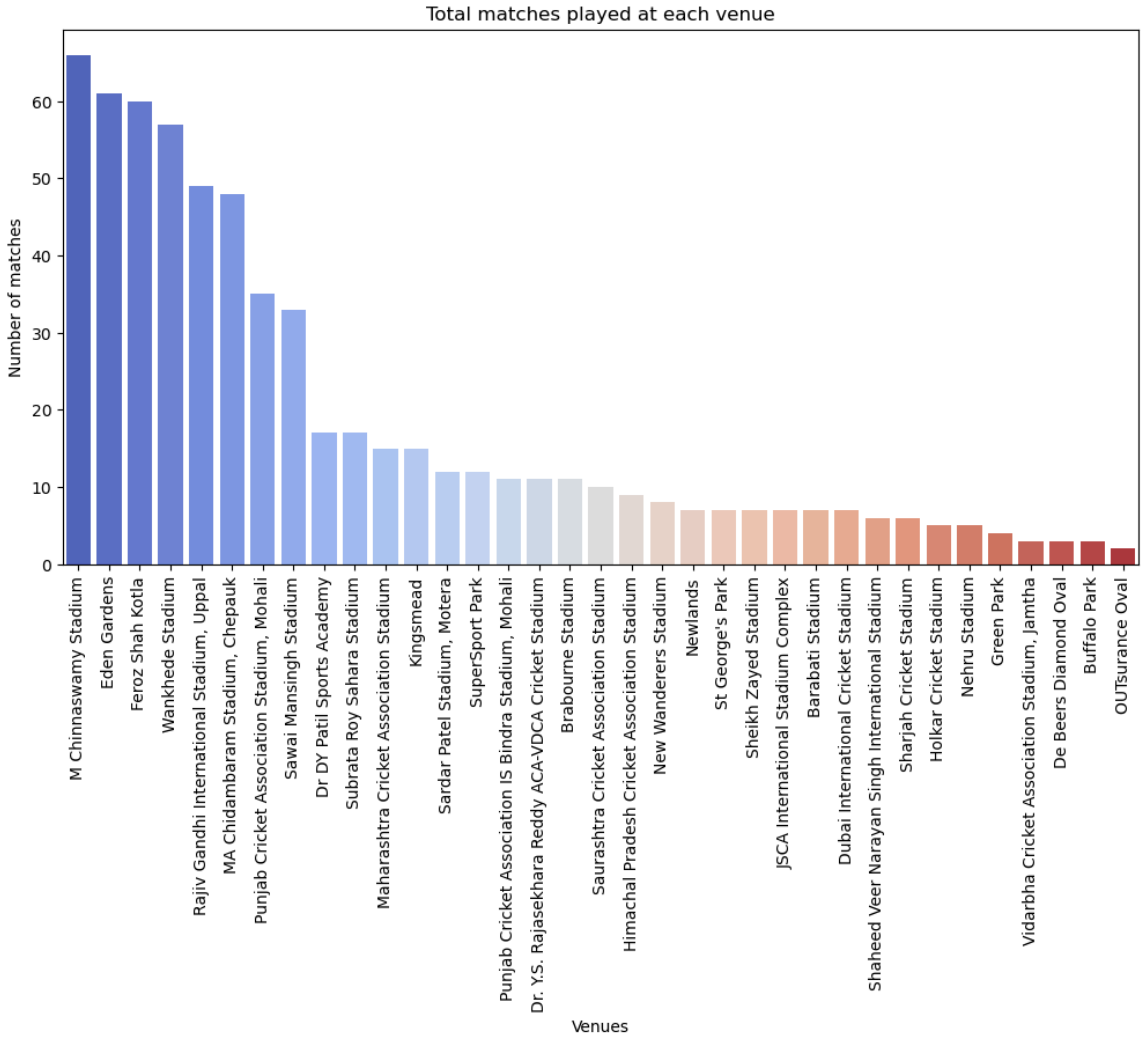
In [8]:
```
# 2.  Visualize the total number of matches won by each team.
team_wins = df["winner"].value_counts()
plt.figure(figsize=(12,6))
sns.barplot(x=team_wins.index, y=team_wins.values, palette="viridis")
plt.xticks(rotation=90)
plt.xlabel("Teams")
plt.ylabel("Total Wins")
plt.title("Total matches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



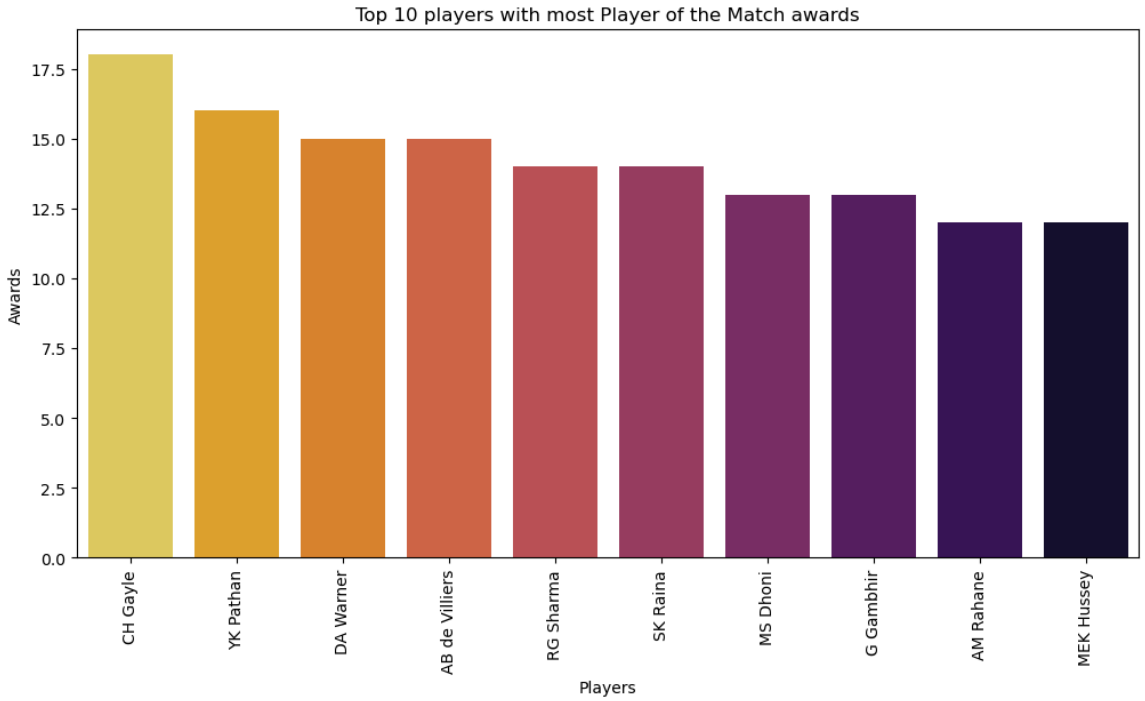
In [80]:
```
# 3. List the names of umpires who have served as first umpire and the numbe
first_umpires = df["umpire1"].dropna().value_counts()
print(f"First umpires and the number of matches umpired:\n{first_umpires}")
```

```
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 Shamshuddin             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
```

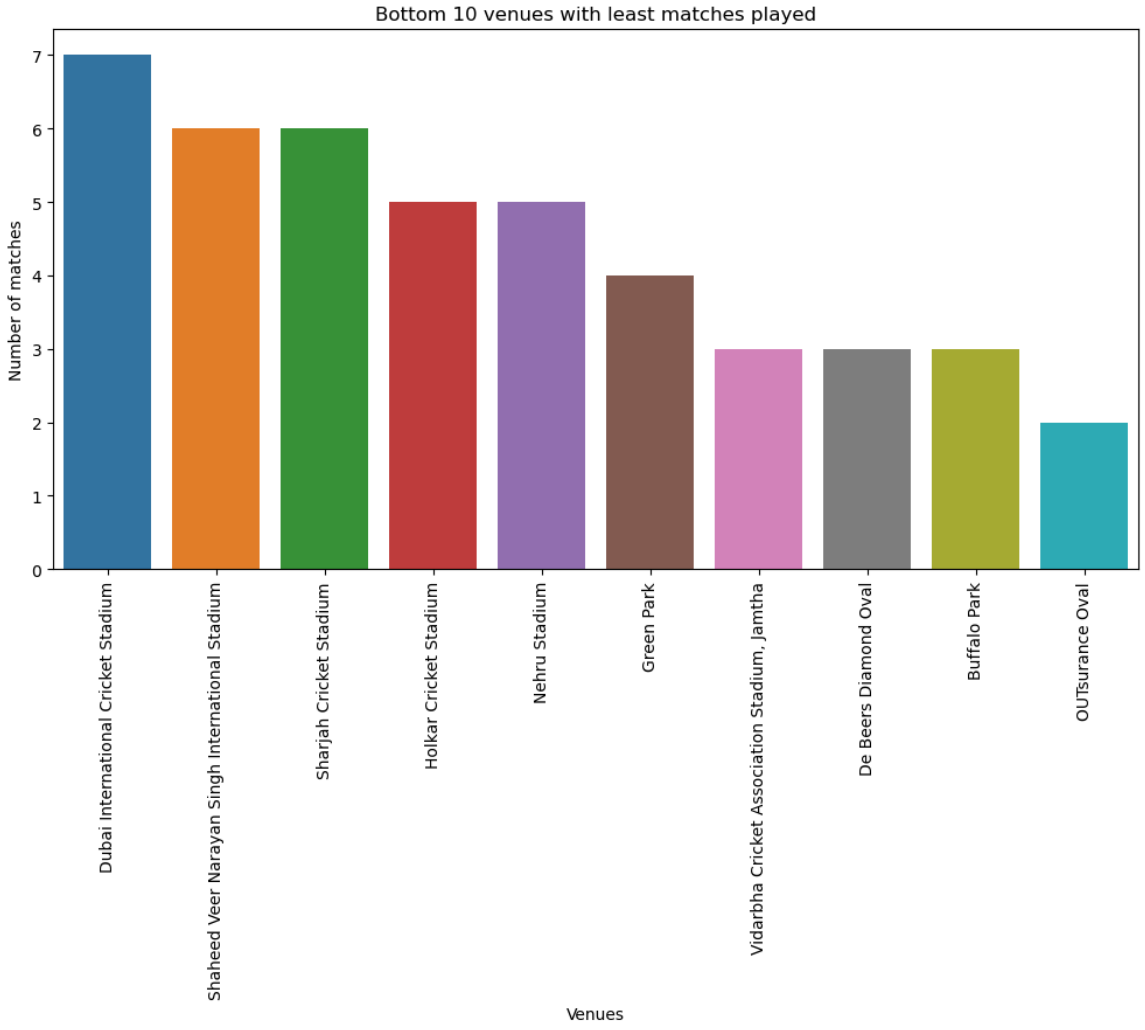
In [79]:
```
# 4. Visualize how many matches were held in each venue.
venue_counts = df["venue"].value_counts()
plt.figure(figsize=(12,6))
sns.barplot(x=venue_counts.index, y=venue_counts.values, palette="coolwarm")
plt.xticks(rotation=90)
plt.xlabel("Venues")
plt.ylabel("Number of matches")
plt.title("Total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



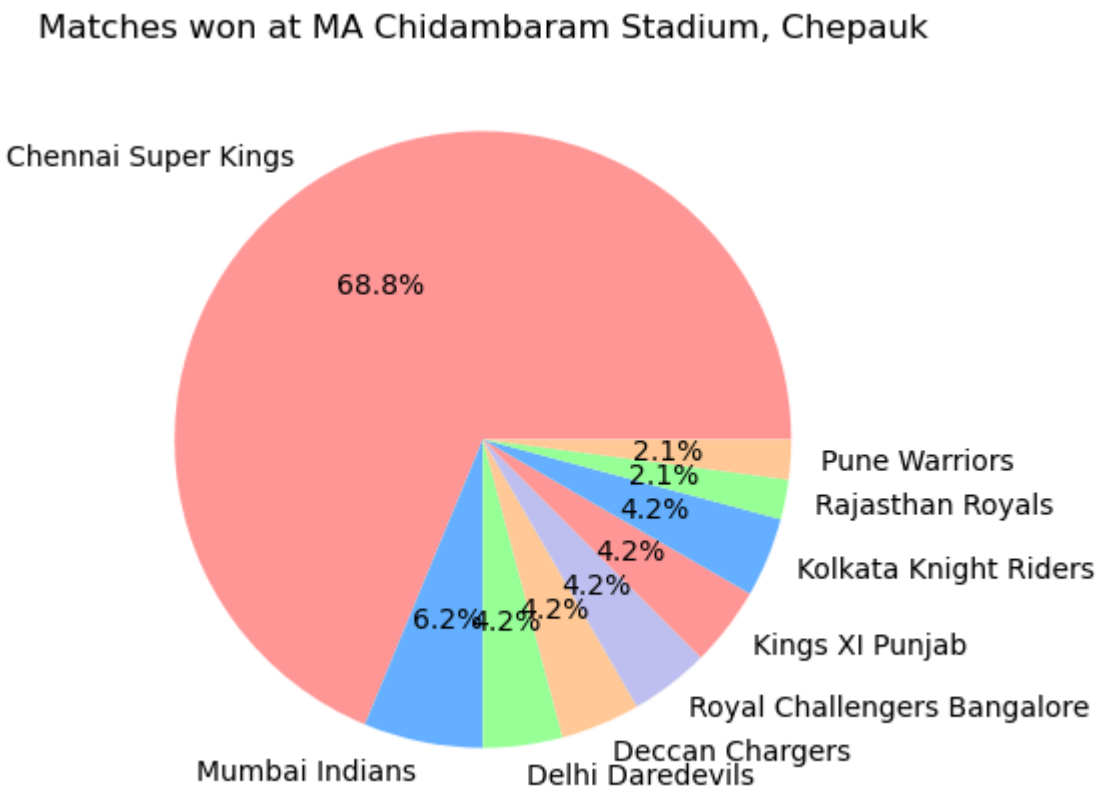
In [78]:
```
import random
# 5. Visualize the top 10 players identified as the player of the match for
top_players = df["player_of_match"].value_counts().head(10)
plt.figure(figsize=(12,6))
palettes1=['Accent', 'Accent_r', 'Blues', 'Blues_r', 'BrBG', 'BrBG_r', 'BuGr
sns.barplot(x=top_players.index, y=top_players.values, 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 Chidamba
chidambaram_wins = df[df["venue"] == "MA Chidambaram Stadium, Chepauk"]["wir
plt.figure(figsize=(10, 5))
colors = ['#ff9999', '#66b3ff', '#99ff99', '#ffcc99', '#c2c2f0']
plt.pie(chidambaram_wins, labels=chidambaram_wins.index, autopct="%1.1f%%",c
plt.title("Matches won at MA Chidambaram Stadium, Chepauk")
plt.show()
```



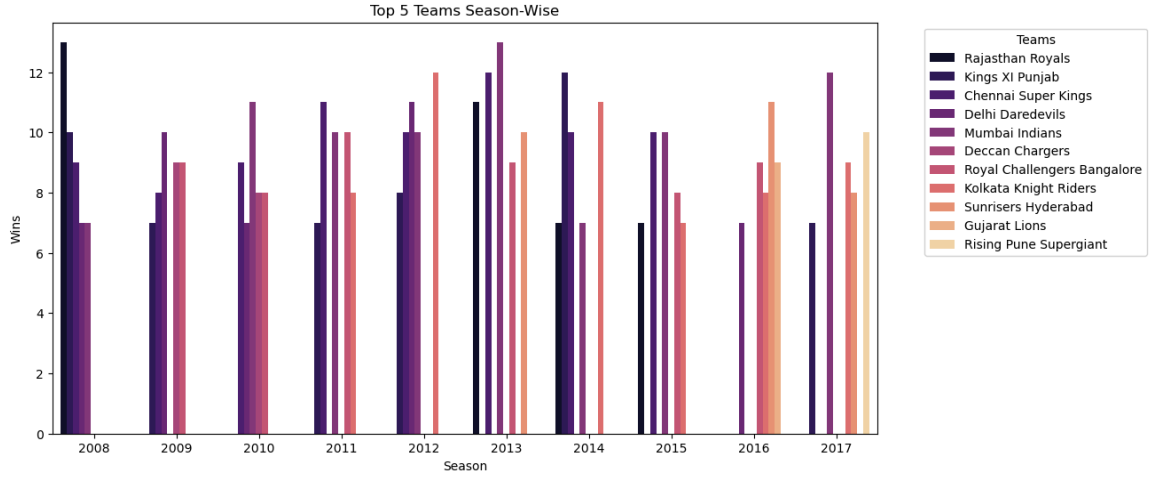
In [25]:
```
# 8. Find and list the venues where the match resulted in a tie.
tie_venues = df[df["result"] == "tie"]["venue"].unique()
print("Venues where matches ended in a tie:\n")
for i in tie_venues: print(i)
```

```
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="w
# Sort values by season and wins in descending order to get the top teams pe
season_winners_sorted = season_winners.sort_values(["season", "wins"], ascer
# 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'
sns.barplot(x="season", y="wins", hue="winner", data=top_teams_season, palet
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 Winn
plt.title("Correlation Between Toss Winning and Match Winning")
plt.show()
```