In [41]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [42]:
```python
DF1 =pd.read_csv("deliveries.csv")
```

In [43]:
```python
DF1.head()
```

Out[43]:

| | match_id | inning | batting_team | bowling_team | over | ball | batsman | non_striker | bowler | is_super_over | ... | bye_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 1 | DA Warner | S Dhawan | TS Mills | 0 | ... | |
| **1** | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 2 | DA Warner | S Dhawan | TS Mills | 0 | ... | |
| **2** | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 3 | DA Warner | S Dhawan | TS Mills | 0 | ... | |
| **3** | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 4 | DA Warner | S Dhawan | TS Mills | 0 | ... | |
| **4** | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 5 | DA Warner | S Dhawan | TS Mills | 0 | ... | |

5 rows × 21 columns

In [44]:
```python
DF2  = pd.read_csv("matches.csv")
```

In [45]:
```python
DF2.head()
```

Out[45]:

| | id | season | city | date | team1 | team2 | toss_winner | toss_decision | result | dl_applied | winner |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2017 | Hyderabad | 05-04-2017 | Sunrisers Hyderabad | Royal Challengers Bangalore | Royal Challengers Bangalore | field | normal | 0 | Sunrisers Hyderabad |
| **1** | 2 | 2017 | Pune | 06-04-2017 | Mumbai Indians | Rising Pune Supergiant | Rising Pune Supergiant | field | normal | 0 | Rising Pune Supergiant |
| **2** | 3 | 2017 | Rajkot | 07-04-2017 | Gujarat Lions | Kolkata Knight Riders | Kolkata Knight Riders | field | normal | 0 | Kolkata Knight Riders |
| **3** | 4 | 2017 | Indore | 08-04-2017 | Rising Pune Supergiant | Kings XI Punjab | Kings XI Punjab | field | normal | 0 | Kings XI Punjab |

|   | id | season | city | date | team1 | team2 | toss_winner | toss_decision | result | dl_applied | winner |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **4** | 5 | 2017 | Bangalore | 08-04-2017 | Royal Challengers Bangalore | Delhi Daredevils | Royal Challengers Bangalore | bat | normal | 0 | Royal Challengers Bangalore |

In [46]:
```
DF1.shape
```

Out[46]:
```
(179078, 21)
```

In [47]:
```
DF2.shape
```

Out[47]:
```
(756, 18)
```

In [48]:
```
DF1.describe()
```

Out[48]:

|   | match_id | inning | over | ball | is_super_over | wide_runs | bye_runs |
|---|---|---|---|---|---|---|---|
| **count** | 179078.000000 | 179078.000000 | 179078.000000 | 179078.000000 | 179078.000000 | 179078.000000 | 179078.000000 |
| **mean** | 1802.252957 | 1.482952 | 10.162488 | 3.615587 | 0.000452 | 0.036721 | 0.004936 |
| **std** | 3472.322805 | 0.502074 | 5.677684 | 1.806966 | 0.021263 | 0.251161 | 0.116480 |
| **min** | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 190.000000 | 1.000000 | 5.000000 | 2.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 379.000000 | 1.000000 | 10.000000 | 4.000000 | 0.000000 | 0.000000 | 0.000000 |
| **75%** | 567.000000 | 2.000000 | 15.000000 | 5.000000 | 0.000000 | 0.000000 | 0.000000 |
| **max** | 11415.000000 | 5.000000 | 20.000000 | 9.000000 | 1.000000 | 5.000000 | 4.000000 |

In [49]:
```
DF2.describe()
```

Out[49]:

|   | id | season | dl_applied | win_by_runs | win_by_wickets |
|---|---|---|---|---|---|
| **count** | 756.000000 | 756.000000 | 756.000000 | 756.000000 | 756.000000 |
| **mean** | 1792.178571 | 2013.444444 | 0.025132 | 13.283069 | 3.350529 |
| **std** | 3464.478148 | 3.366895 | 0.156630 | 23.471144 | 3.387963 |
| **min** | 1.000000 | 2008.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 189.750000 | 2011.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 378.500000 | 2013.000000 | 0.000000 | 0.000000 | 4.000000 |
| **75%** | 567.250000 | 2016.000000 | 0.000000 | 19.000000 | 6.000000 |
| **max** | 11415.000000 | 2019.000000 | 1.000000 | 146.000000 | 10.000000 |

In [50]:
```
DF1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 179078 entries, 0 to 179077
Data columns (total 21 columns):
 #   Column            Non-Null Count   Dtype
 ---  ------            --------------   -----
 0    match_id          179078 non-null  int64
 1    inning            179078 non-null  int64
 2    batting_team      179078 non-null  object
 3    bowling_team      179078 non-null  object
 4    over              179078 non-null  int64
 5    ball              179078 non-null  int64
 6    batsman           179078 non-null  object
 7    non_striker       179078 non-null  object
 8    bowler            179078 non-null  object
 9    is_super_over     179078 non-null  int64
 10   wide_runs         179078 non-null  int64
 11   bye_runs          179078 non-null  int64
 12   legbye_runs       179078 non-null  int64
 13   noball_runs       179078 non-null  int64
 14   penalty_runs      179078 non-null  int64
 15   batsman_runs      179078 non-null  int64
 16   extra_runs        179078 non-null  int64
 17   total_runs        179078 non-null  int64
 18   player_dismissed  8834 non-null    object
 19   dismissal_kind    8834 non-null    object
 20   fielder           6448 non-null    object
dtypes: int64(13), object(8)
memory usage: 28.7+ MB
```

In [51]:
```
DF2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 18 columns):
 #    Column           Non-Null Count  Dtype
 ---  ------           --------------  -----
 0    id               756 non-null    int64
 1    season           756 non-null    int64
 2    city             749 non-null    object
 3    date             756 non-null    object
 4    team1            756 non-null    object
 5    team2            756 non-null    object
 6    toss_winner      756 non-null    object
 7    toss_decision    756 non-null    object
 8    result           756 non-null    object
 9    dl_applied       756 non-null    int64
 10   winner           752 non-null    object
 11   win_by_runs      756 non-null    int64
 12   win_by_wickets   756 non-null    int64
 13   player_of_match  752 non-null    object
 14   venue            756 non-null    object
 15   umpire1          754 non-null    object
 16   umpire2          754 non-null    object
 17   umpire3          119 non-null    object
dtypes: int64(5), object(13)
memory usage: 106.4+ KB
```

In [52]:
```
DF1.columns
```

Out[52]:
```
Index(['match_id', 'inning', 'batting_team', 'bowling_team', 'over', 'ball',
       'batsman', 'non_striker', 'bowler', 'is_super_over', 'wide_runs',
       'bye_runs', 'legbye_runs', 'noball_runs', 'penalty_runs',
       'batsman_runs', 'extra_runs', 'total_runs', 'player_dismissed',
       'dismissal_kind', 'fielder'],
      dtype='object')
```

In [53]:
```
DF2.columns
```

```
Out[53]: Index(['id', 'season', 'city', 'date', 'team1', 'team2', 'toss_winner',
               'toss_decision', 'result', 'dl_applied', 'winner', 'win_by_runs',
               'win_by_wickets', 'player_of_match', 'venue', 'umpire1', 'umpire2',
               'umpire3'],
              dtype='object')
```

```
In [54]: DF1.isnull().sum().sum()
```

```
Out[54]: 513118
```

```
In [55]: DF2.isnull().sum().sum()
```

```
Out[55]: 656
```

```
In [56]: # 1 Total number of innings of the matches played
         DF1['inning'].value_counts().sum()
```

```
Out[56]: 179078
```

```
In [57]: # 2 How many IPL seasons are we using to analyse
         DF2['season'].nunique()
```

```
Out[57]: 12
```

```
In [58]: DF2['season'].unique()
```

```
Out[58]: array([2017, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2018,
                2019], dtype=int64)
```

```
In [59]: # 3 Total number of matches played according to the dataset
         DF2 ['id'].count()
```

```
Out[59]: 756
```

```
In [60]: # 4 Which IPL team won by maximum runs

         DF2.loc[DF2['win_by_runs'].idxmax()]['winner']
```

```
Out[60]: 'Mumbai Indians'
```

```
In [61]: # 5 Which IPL team won by minimum runs

         DF2.loc[DF2['win_by_runs'].idxmin()]['winner']
```

```
Out[61]: 'Rising Pune Supergiant'
```

```
In [62]: # 6 Which IPL team won by consuming maximum wickets

         DF2.loc[DF2['win_by_wickets'].idxmax()]['winner']
```

```
Out[62]: 'Kolkata Knight Riders'
```

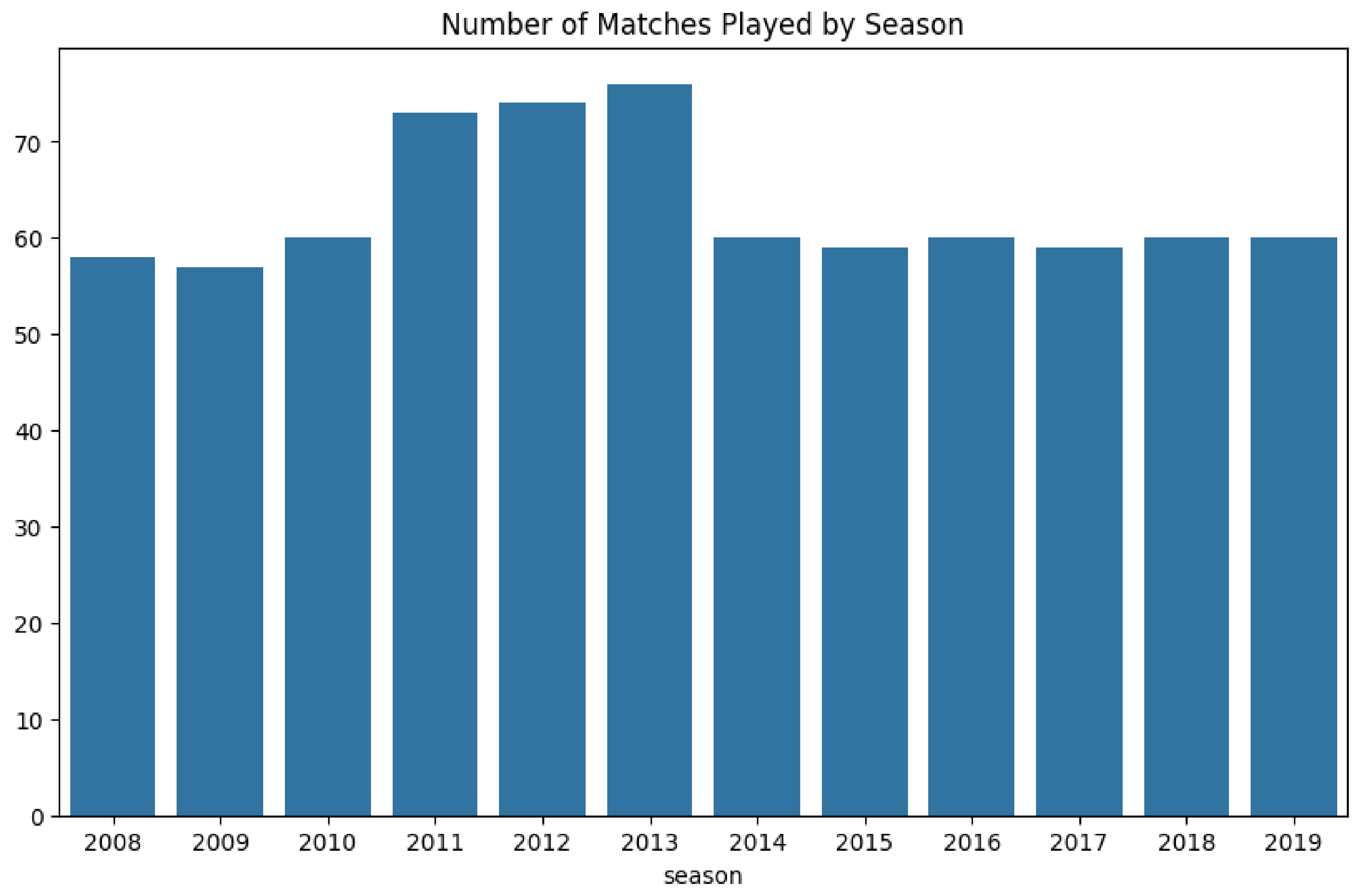
```
In [63]:
```

```
# 7 Which IPL team won by consuming minimum wickets

DF2.loc[DF2['win_by_wickets'].idxmin()]['winner']
```

Out[63]:
```
'Sunrisers Hyderabad'
```

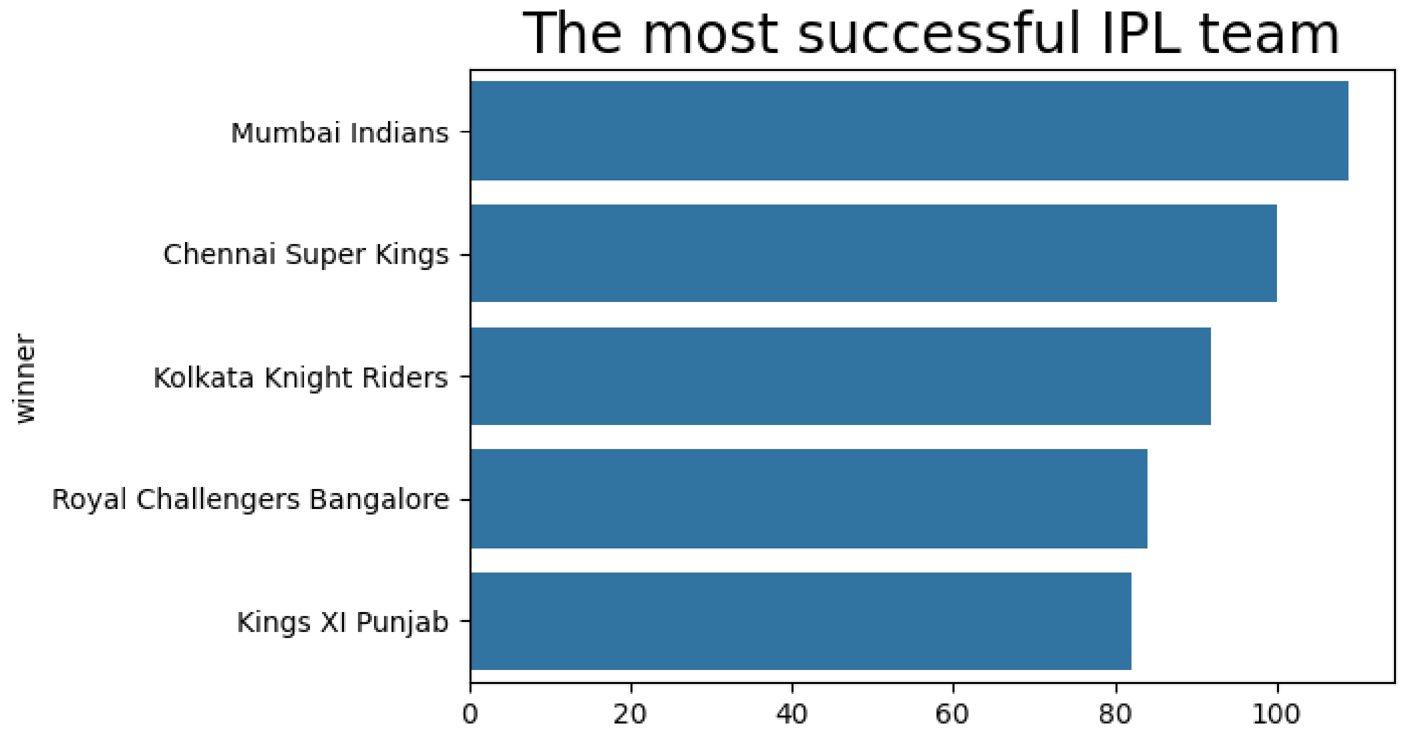
VISUALIZATION

In [64]:
```
# Which season consisted of the highest number of matches ever played

season_counts = DF2['season'].value_counts().head(12)

plt.figure(figsize=(10, 6))
sns.barplot(x=season_counts.index, y=season_counts.values)
plt.title('Number of Matches Played by Season',fontsize=12)
plt.show()
```



In [65]:
```
#Which is the most successful IPL team with all the data at hand

winner = DF2 ['winner'].value_counts().head()

plt.figure(figsize=(6,4))
sns.barplot(y=winner.index,x=winner.values)
plt.title("The most successful IPL team",fontsize=20)
```

Out[65]:
```
Text(0.5, 1.0, 'The most successful IPL team')
```



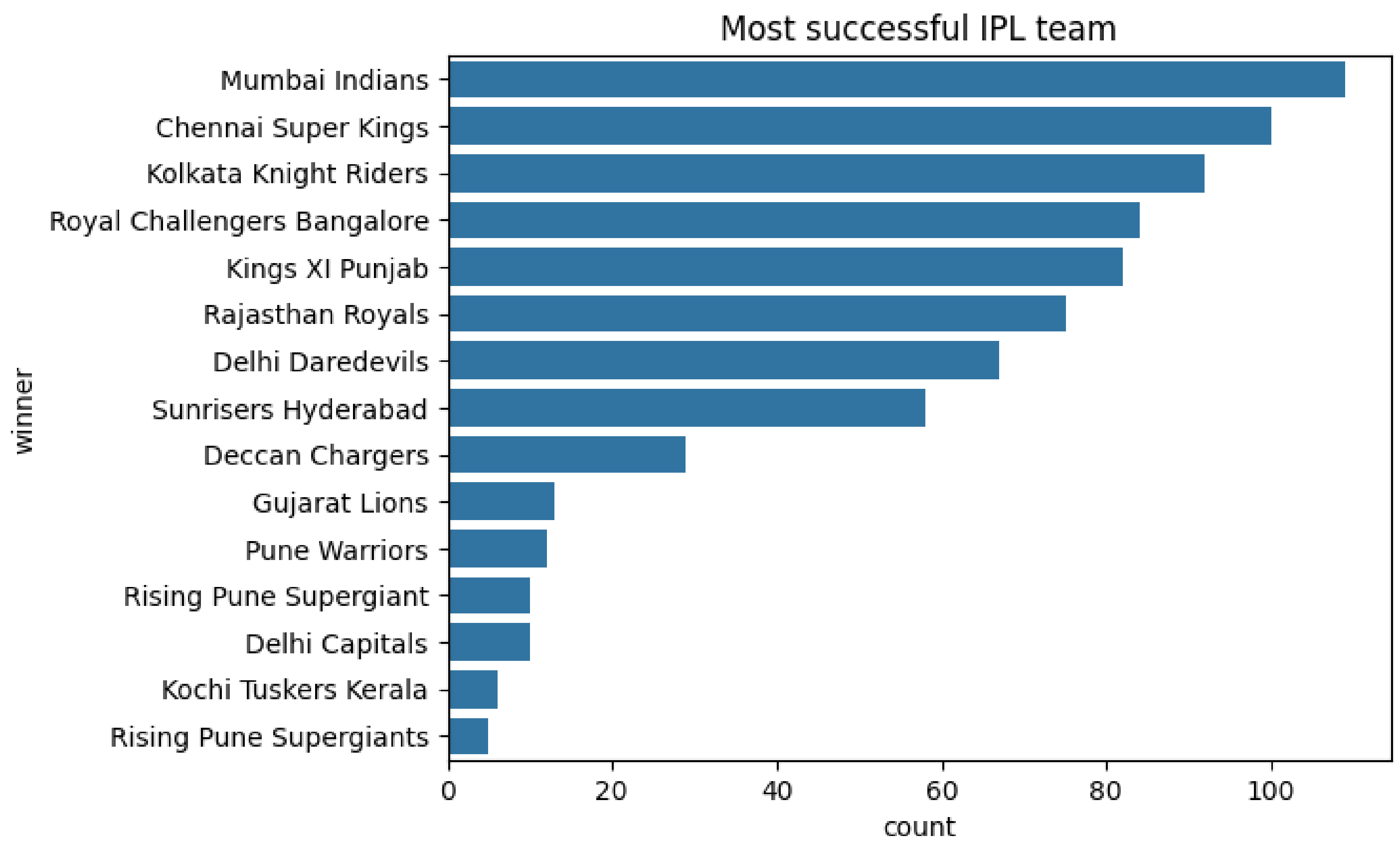
In [66]:
```
data = DF2["winner"].value_counts()
```

In [67]:
```
data
```

Out[67]:
```
winner
Mumbai Indians                 109
Chennai Super Kings            100
Kolkata Knight Riders           92
Royal Challengers Bangalore     84
Kings XI Punjab                 82
Rajasthan Royals                75
Delhi Daredevils                67
Sunrisers Hyderabad             58
Deccan Chargers                 29
Gujarat Lions                   13
Pune Warriors                   12
Rising Pune Supergiant          10
Delhi Capitals                  10
Kochi Tuskers Kerala             6
Rising Pune Supergiants          5
Name: count, dtype: int64
```

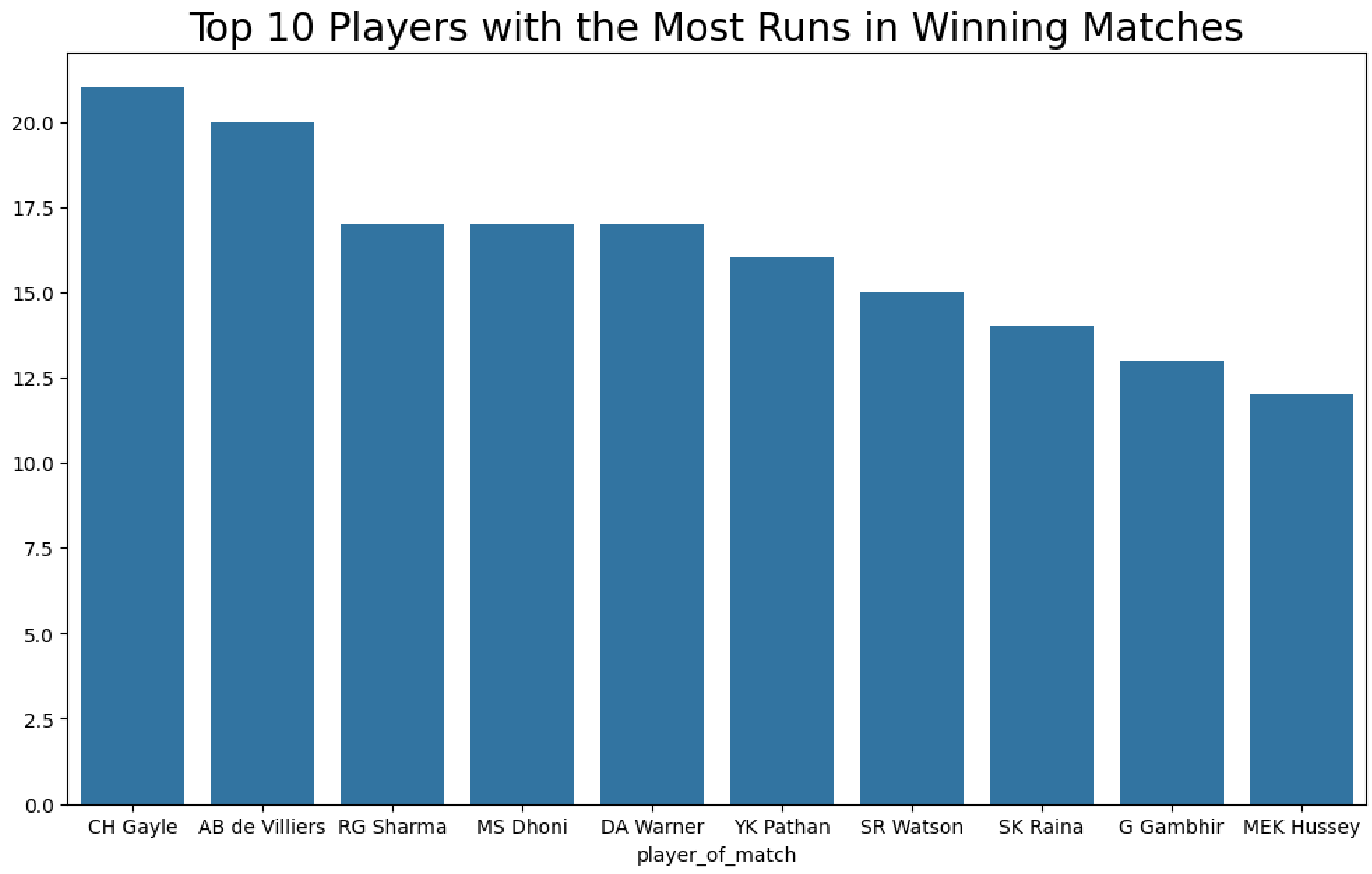
In [68]:
```
sns.barplot(y= data.index,x=data,orient='h')
plt.title("Most successful IPL team",fontsize=12)
```

Out[68]:
```
Text(0.5, 1.0, 'Most successful IPL team')
```



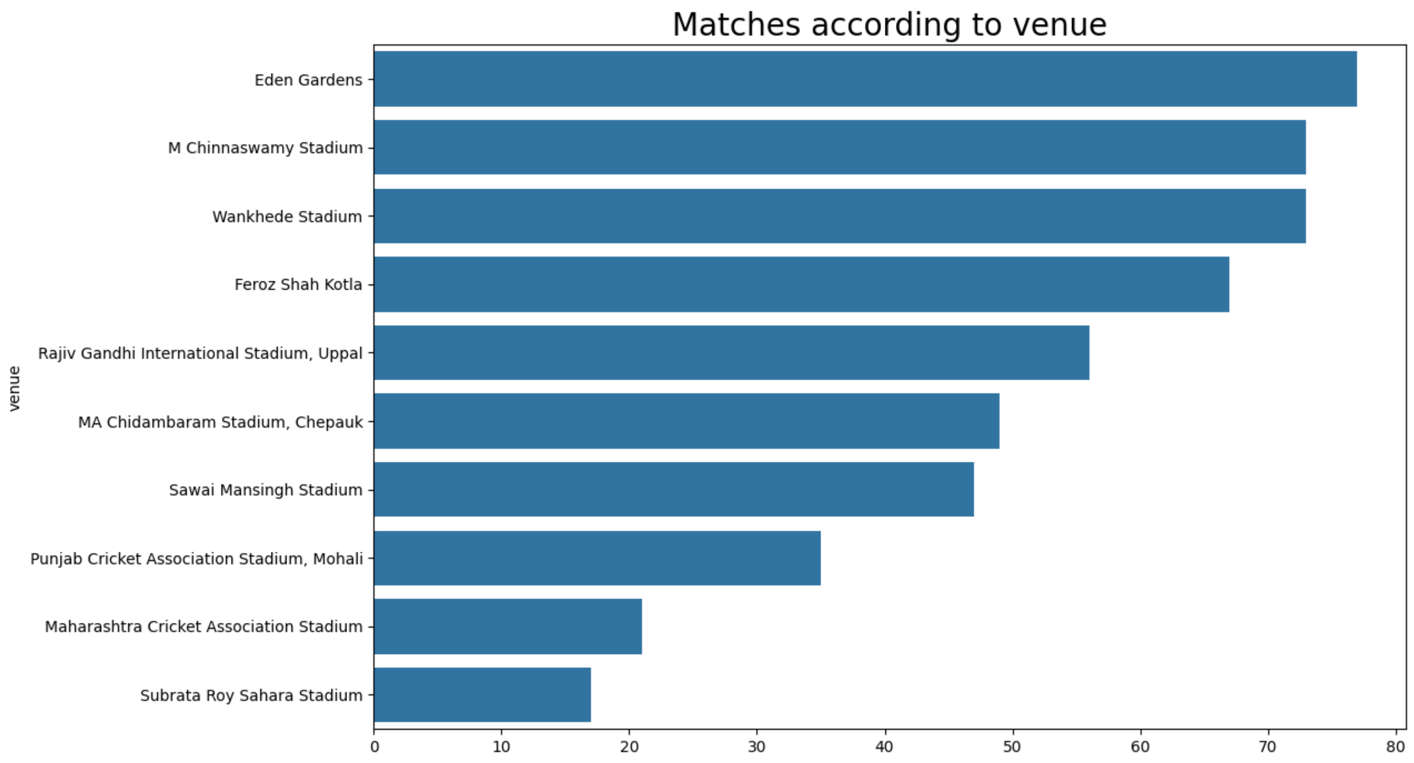
In [69]:
```
# Top players of winning matches

top_players = DF2['player_of_match'].value_counts().head(10)

plt.figure(figsize=(12, 7))
sns.barplot(x=top_players.index, y=top_players.values)
plt.title('Top 10 Players with the Most Runs in Winning Matches',fontsize =20)
```

Out[69]:
```
Text(0.5, 1.0, 'Top 10 Players with the Most Runs in Winning Matches')
```



In [70]:
```
#Matches according to venue

matches_venue = DF2['venue'].value_counts().head(10)

plt.figure(figsize=(12, 8))
sns.barplot(y=matches_venue.index, x=matches_venue.values)
plt.title('Matches according to venue',fontsize =20)
```

Out[70]:
```
Text(0.5, 1.0, 'Matches according to venue')
```

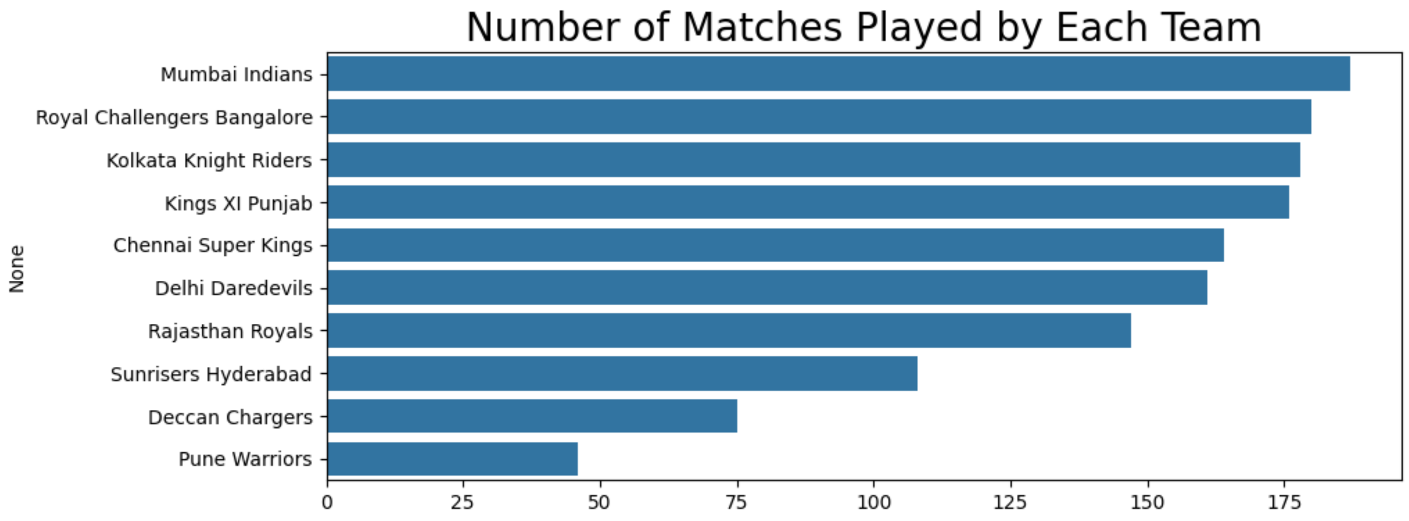


In [71]:
```python
# Matches according to venue
matches = DF2.venue.value_counts().head(10)
matches
```

Out[71]:
```
venue
Eden Gardens                                    77
M Chinnaswamy Stadium                           73
Wankhede Stadium                                73
Feroz Shah Kotla                                67
Rajiv Gandhi International Stadium, Uppal       56
MA Chidambaram Stadium, Chepauk                 49
Sawai Mansingh Stadium                          47
Punjab Cricket Association Stadium, Mohali      35
Maharashtra Cricket Association Stadium         21
Subrata Roy Sahara Stadium                      17
Name: count, dtype: int64
```

In [72]:
```python
#The number of matches played by each team

teams = pd.concat([DF2['team1'],DF2['team2']])

all_teams = teams.value_counts().head(10)

plt.figure(figsize=(10, 4))
sns.barplot(y=all_teams.index, x=all_teams.values)
plt.title('Number of Matches Played by Each Team',fontsize=20)
```

Out[72]:
```
Text(0.5, 1.0, 'Number of Matches Played by Each Team')
```



In [73]:
```python
#The winners in each Season

winners = DF2.groupby(["season","winner"])['id'].count().reset_index()
winners
```

Out[73]:

| | season | winner | id |
|---|---|---|---|
| **0** | 2008 | Chennai Super Kings | 9 |
| **1** | 2008 | Deccan Chargers | 2 |
| **2** | 2008 | Delhi Daredevils | 7 |
| **3** | 2008 | Kings XI Punjab | 10 |
| **4** | 2008 | Kolkata Knight Riders | 6 |
| **...** | ... | ... | ... |

|  | season | winner | id |
|---|---|---|---|
| **95** | 2019 | Kolkata Knight Riders | 6 |
| **96** | 2019 | Mumbai Indians | 11 |
| **97** | 2019 | Rajasthan Royals | 5 |
| **98** | 2019 | Royal Challengers Bangalore | 5 |
| **99** | 2019 | Sunrisers Hyderabad | 6 |

100 rows × 3 columns

In [74]:

```
winners.pivot_table(values='winner',index='season',columns='winner').fillna(0)
```

Out[74]:

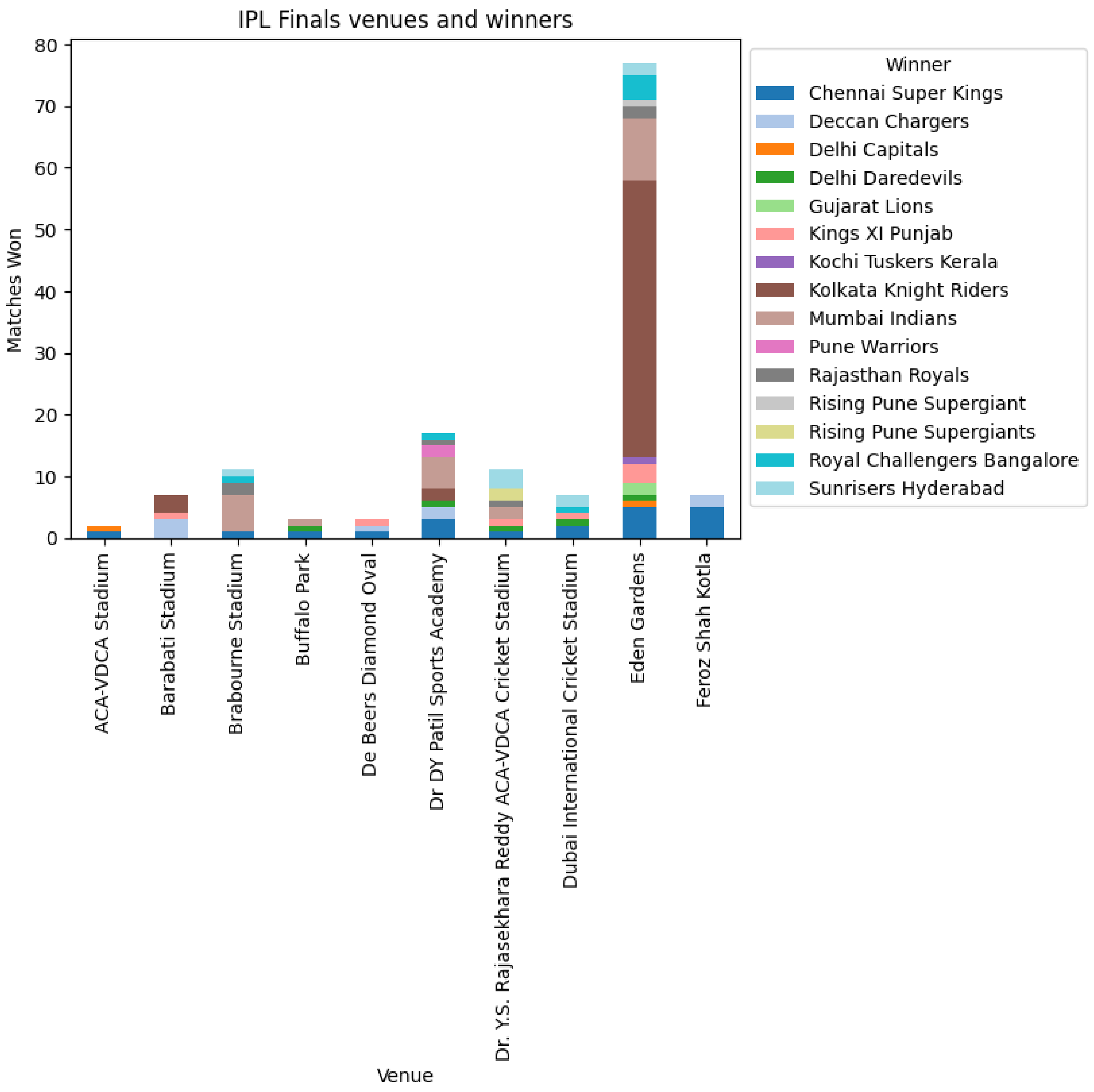
| winner | Chennai Super Kings | Deccan Chargers | Delhi Capitals | Delhi Daredevils | Gujarat Lions | Kings XI Punjab | Kochi Tuskers Kerala | Kolkata Knight Riders | Mumbai Indians | Pune Warriors | Rajasthan Royals |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **season** | | | | | | | | | | | |
| **2008** | 9.0 | 2.0 | 0.0 | 7.0 | 0.0 | 10.0 | 0.0 | 6.0 | 7.0 | 0.0 | 13.0 |
| **2009** | 8.0 | 9.0 | 0.0 | 10.0 | 0.0 | 7.0 | 0.0 | 3.0 | 5.0 | 0.0 | 6.0 |
| **2010** | 9.0 | 8.0 | 0.0 | 7.0 | 0.0 | 4.0 | 0.0 | 7.0 | 11.0 | 0.0 | 6.0 |
| **2011** | 11.0 | 6.0 | 0.0 | 4.0 | 0.0 | 7.0 | 6.0 | 8.0 | 10.0 | 4.0 | 6.0 |
| **2012** | 10.0 | 4.0 | 0.0 | 11.0 | 0.0 | 8.0 | 0.0 | 12.0 | 10.0 | 4.0 | 7.0 |
| **2013** | 12.0 | 0.0 | 0.0 | 3.0 | 0.0 | 8.0 | 0.0 | 6.0 | 13.0 | 4.0 | 11.0 |
| **2014** | 10.0 | 0.0 | 0.0 | 2.0 | 0.0 | 12.0 | 0.0 | 11.0 | 7.0 | 0.0 | 7.0 |
| **2015** | 10.0 | 0.0 | 0.0 | 5.0 | 0.0 | 3.0 | 0.0 | 7.0 | 10.0 | 0.0 | 7.0 |
| **2016** | 0.0 | 0.0 | 0.0 | 7.0 | 9.0 | 4.0 | 0.0 | 8.0 | 7.0 | 0.0 | 0.0 |
| **2017** | 0.0 | 0.0 | 0.0 | 6.0 | 4.0 | 7.0 | 0.0 | 9.0 | 12.0 | 0.0 | 0.0 |
| **2018** | 11.0 | 0.0 | 0.0 | 5.0 | 0.0 | 6.0 | 0.0 | 9.0 | 6.0 | 0.0 | 7.0 |
| **2019** | 10.0 | 0.0 | 10.0 | 0.0 | 0.0 | 6.0 | 0.0 | 6.0 | 11.0 | 0.0 | 5.0 |

In [75]:

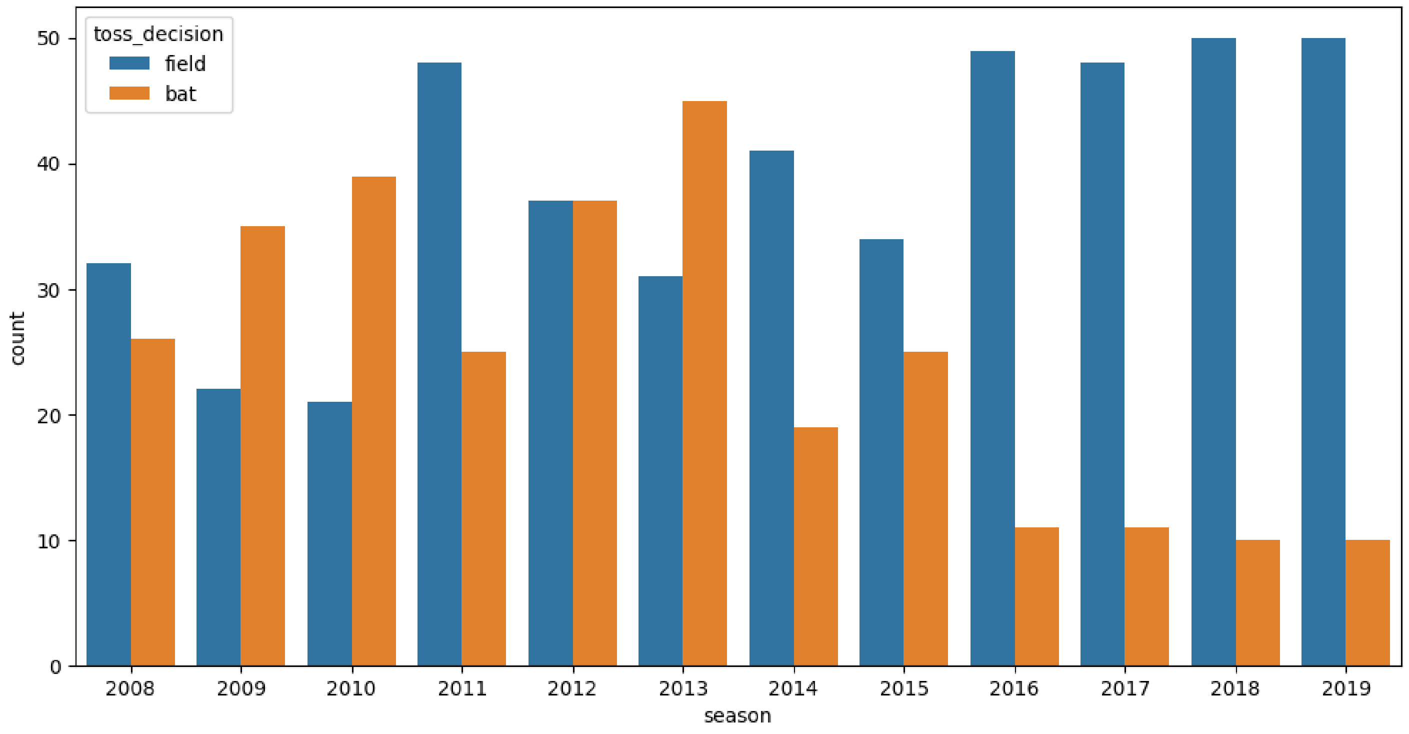
```
#IPL Finals venues and winners along with the number of wins.

venue_winners =DF2.groupby(['venue','winner'])['id'].count().reset_index().head(50)

venue_winners_matrix = venue_winners.pivot(index='venue',columns='winner',values='id').fi

plt.figure(figsize=(12, 8))
venue_winners_matrix.plot(kind='bar', stacked=True, colormap='tab20')
plt.title('IPL Finals venues and winners')
plt.xlabel('Venue')
plt.ylabel('Matches Won')
plt.legend(title='Winner', bbox_to_anchor=(1, 1))
```

Out[75]:

```
<matplotlib.legend.Legend at 0x21e12b37650>

<Figure size 1200x800 with 0 Axes>
```



In [76]:
```
plt.figure(figsize=(12,6))
sns.countplot(x='season', hue="toss_decision", data=DF2)
plt.show()
```



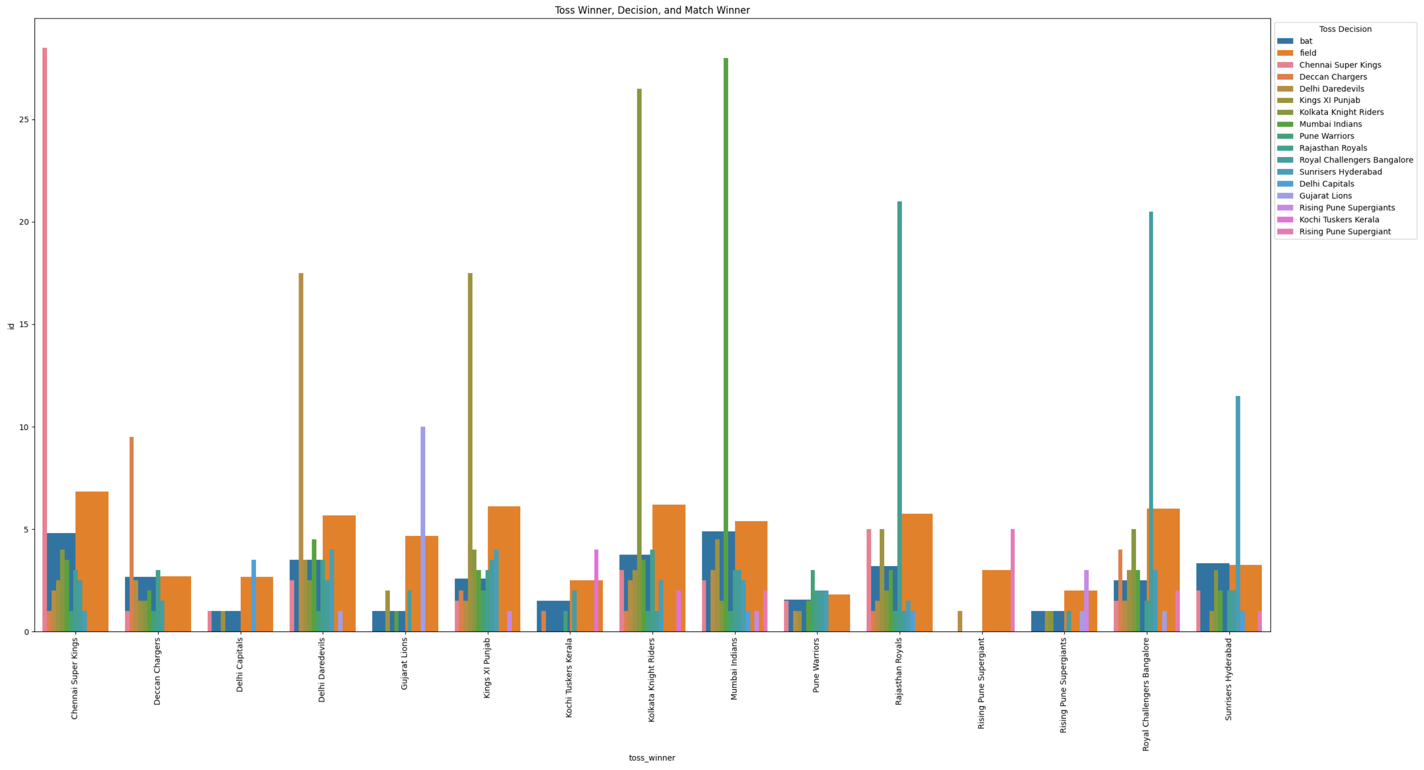
In [77]:

```
#The toss winner, toss decision, winner in final matches.

toss_info = DF2.groupby(['toss_winner','toss_decision','winner'])['id'].count().reset_inde

plt.figure(figsize=(28, 14))
sns.barplot(x='toss_winner', y='id', hue='toss_decision', data=toss_info, ci=None)
sns.barplot(x='toss_winner', y='id', hue='winner', data=toss_info, ci=None)
plt.title('Toss Winner, Decision, and Match Winner')

plt.xticks(rotation=90)
plt.legend(title='Toss Decision', bbox_to_anchor=(1, 1))
```

```
C:\Users\Adesh\AppData\Local\Temp\ipykernel_11092\3136541692.py:6: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='toss_winner', y='id', hue='toss_decision', data=toss_info, ci=None)
C:\Users\Adesh\AppData\Local\Temp\ipykernel_11092\3136541692.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='toss_winner', y='id', hue='winner', data=toss_info, ci=None)
```

Out[77]:

```
<matplotlib.legend.Legend at 0x21e12a65040>
```



In [78]:
```
#The man of the match for each winning team


mm_winners = DF2.groupby(['player_of_match', 'winner'])['id'].count().reset_index()

mm_winner_matrix = mm_winners.pivot(index='player_of_match', columns='winner', values='id'
print(mm_winner_matrix)

mm_winner_matrix
```

```
winner              Chennai Super Kings  Deccan Chargers  Delhi Capitals  \
player_of_match
A Chandila                          0.0              0.0             0.0
A Joseph                            0.0              0.0             0.0
A Kumble                            0.0              1.0             0.0
A Mishra                            0.0              2.0             1.0
A Nehra                             3.0              0.0             0.0
...                                 ...              ...             ...
Washington Sundar                   0.0              0.0             0.0
YK Pathan                           0.0              0.0             0.0
YS Chahal                           0.0              0.0             0.0
Yuvraj Singh                        0.0              0.0             0.0
Z Khan                              0.0              0.0             0.0

winner              Delhi Daredevils  Gujarat Lions  Kings XI Punjab  \
player_of_match
A Chandila                       0.0            0.0              0.0
A Joseph                         0.0            0.0              0.0
A Kumble                         0.0            0.0              0.0
A Mishra                         4.0            0.0              0.0
A Nehra                          1.0            0.0              0.0
...                              ...            ...              ...
Washington Sundar                0.0            0.0              0.0
YK Pathan                        0.0            0.0              0.0
YS Chahal                        0.0            0.0              0.0
Yuvraj Singh                     1.0            0.0              1.0
Z Khan                           1.0            0.0              0.0

winner              Kochi Tuskers Kerala  Kolkata Knight Riders  \
player_of_match
```

```
A Chandila                    0.0                    0.0
A Joseph                      0.0                    0.0
A Kumble                      0.0                    0.0
A Mishra                      0.0                    0.0
A Nehra                       0.0                    0.0
...                           ...                    ...
Washington Sundar             0.0                    0.0
YK Pathan                     0.0                    7.0
YS Chahal                     0.0                    0.0
Yuvraj Singh                  0.0                    0.0
Z Khan                        0.0                    0.0

winner             Mumbai Indians  Pune Warriors  Rajasthan Royals  \
player_of_match
A Chandila                    0.0            0.0               1.0
A Joseph                      1.0            0.0               0.0
A Kumble                      0.0            0.0               0.0
A Mishra                      0.0            0.0               0.0
A Nehra                       1.0            0.0               0.0
...                           ...            ...               ...
Washington Sundar             0.0            0.0               0.0
YK Pathan                     1.0            0.0               8.0
YS Chahal                     0.0            0.0               0.0
Yuvraj Singh                  0.0            0.0               0.0
Z Khan                        0.0            0.0               0.0

winner             Rising Pune Supergiant  Rising Pune Supergiants  \
player_of_match
A Chandila                            0.0                      0.0
A Joseph                              0.0                      0.0
A Kumble                              0.0                      0.0
A Mishra                              0.0                      0.0
A Nehra                               0.0                      0.0
...                                   ...                      ...
Washington Sundar                     1.0                      0.0
YK Pathan                             0.0                      0.0
YS Chahal                             0.0                      0.0
Yuvraj Singh                          0.0                      0.0
Z Khan                                0.0                      0.0

winner             Royal Challengers Bangalore  Sunrisers Hyderabad
player_of_match
A Chandila                                 0.0                  0.0
A Joseph                                   0.0                  0.0
A Kumble                                   2.0                  0.0
A Mishra                                   0.0                  4.0
A Nehra                                    0.0                  1.0
...                                        ...                  ...
Washington Sundar                          0.0                  0.0
YK Pathan                                  0.0                  0.0
YS Chahal                                  1.0                  0.0
Yuvraj Singh                               2.0                  1.0
Z Khan                                     0.0                  0.0

[226 rows x 15 columns]
```

Out[78]:

| winner | Chennai Super Kings | Deccan Chargers | Delhi Capitals | Delhi Daredevils | Gujarat Lions | Kings XI Punjab | Kochi Tuskers Kerala | Kolkata Knight Riders | Mumbai Indians | Pune Warriors | Ra |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **player_of_match** | | | | | | | | | | | |
| **A Chandila** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **A Joseph** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | |
| **A Kumble** | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |

| winner | Chennai Super Kings | Deccan Chargers | Delhi Capitals | Delhi Daredevils | Gujarat Lions | Kings XI Punjab | Kochi Tuskers Kerala | Kolkata Knight Riders | Mumbai Indians | Pune Warriors | Ra |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **player_of_match** | | | | | | | | | | | |
| **A Mishra** | 0.0 | 2.0 | 1.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **A Nehra** | 3.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **Washington Sundar** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **YK Pathan** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 7.0 | 1.0 | 0.0 | |
| **YS Chahal** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **Yuvraj Singh** | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **Z Khan** | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |

226 rows × 15 columns

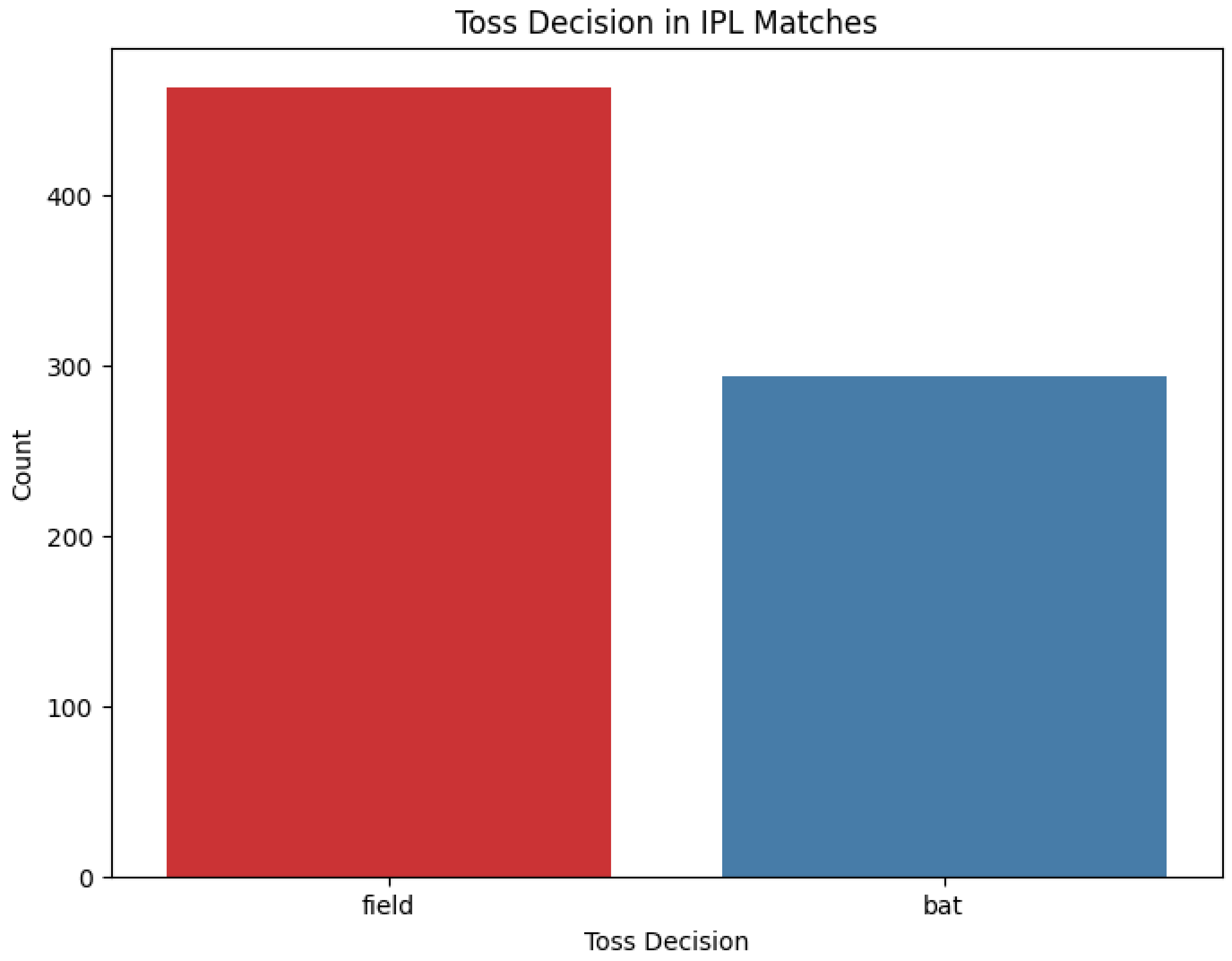
In [79]:

```python
#Decision in every toss either fielding or batting

decision_for_toss = DF2['toss_decision'].value_counts()

plt.figure(figsize=(8, 6))
sns.barplot(x=decision_for_toss.index, y=decision_for_toss.values, palette='Set1')
plt.title('Toss Decision in IPL Matches')
plt.xlabel('Toss Decision')
plt.ylabel('Count')
```

```
C:\Users\Adesh\AppData\Local\Temp\ipykernel_11092\4000637388.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=decision_for_toss.index, y=decision_for_toss.values, palette='Set1')
```

Out[79]:

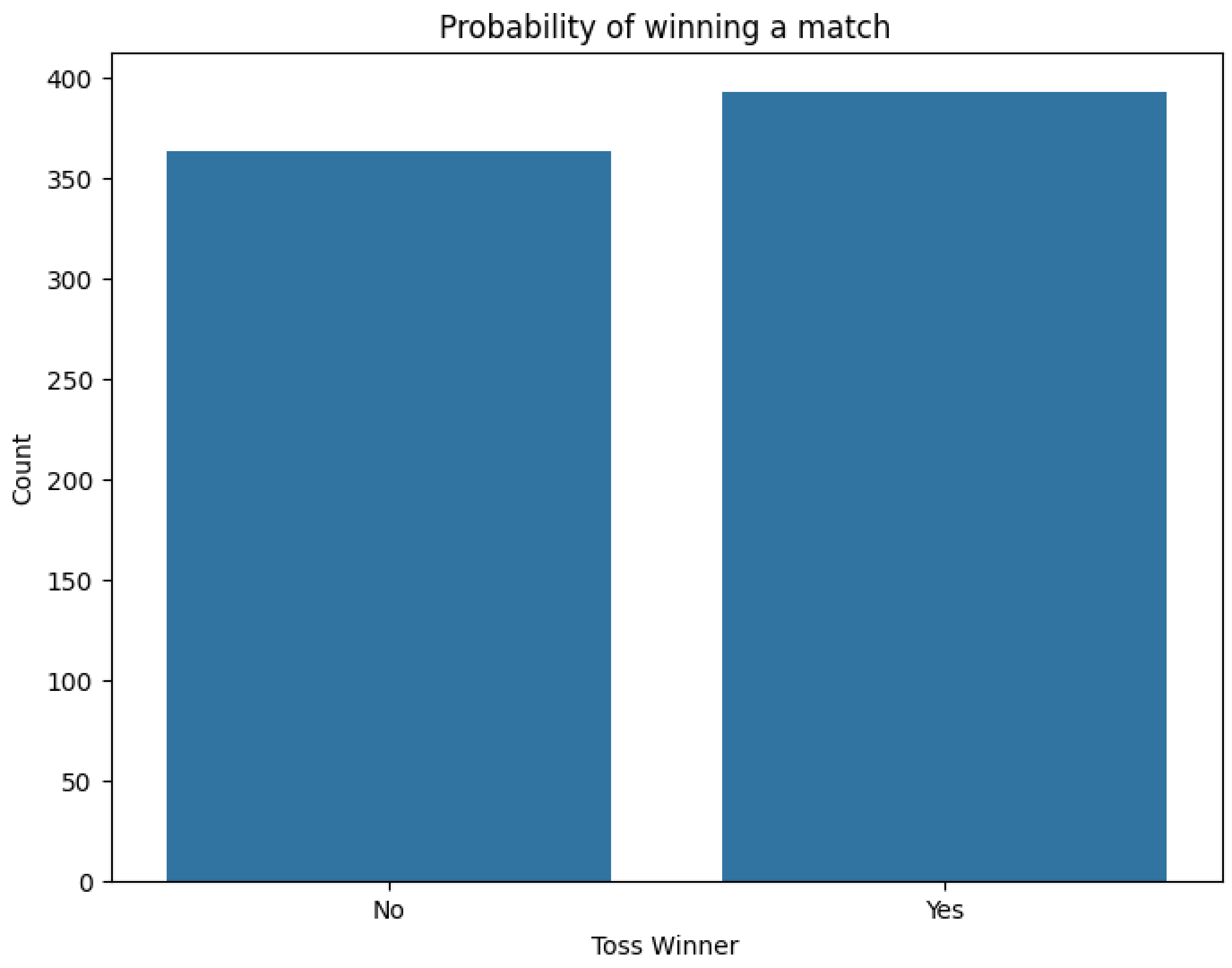
```
Text(0, 0.5, 'Count')
```



```
In [80]: #Decision in every toss either fielding or batting
         Toss_winners = DF2.toss_decision.value_counts()
         Toss_winners
```

```
Out[80]: toss_decision
         field    463
         bat      293
         Name: count, dtype: int64
```

```
In [81]: #What is the probability of winning a match if the toss was won

         DF2['toss_match_winner'] = DF2['toss_winner'] == DF2['winner']

         plt.figure(figsize=(8, 6))
         sns.countplot(x='toss_match_winner', data=DF2)
         plt.title('Probability of winning a match')
         plt.xlabel('Toss Winner')
         plt.ylabel('Count')
         plt.xticks(ticks=[0, 1], labels=['No', 'Yes'])
         plt.show()
```



```
In [82]: #-20.The total runs by fours hit and the total number of fours hit by each team

         four_data=DF1[DF1['batsman_runs']==4]
         four_data.groupby('batting_team')['batsman_runs'].agg([('runs by fours','sum'),('fours','c
```

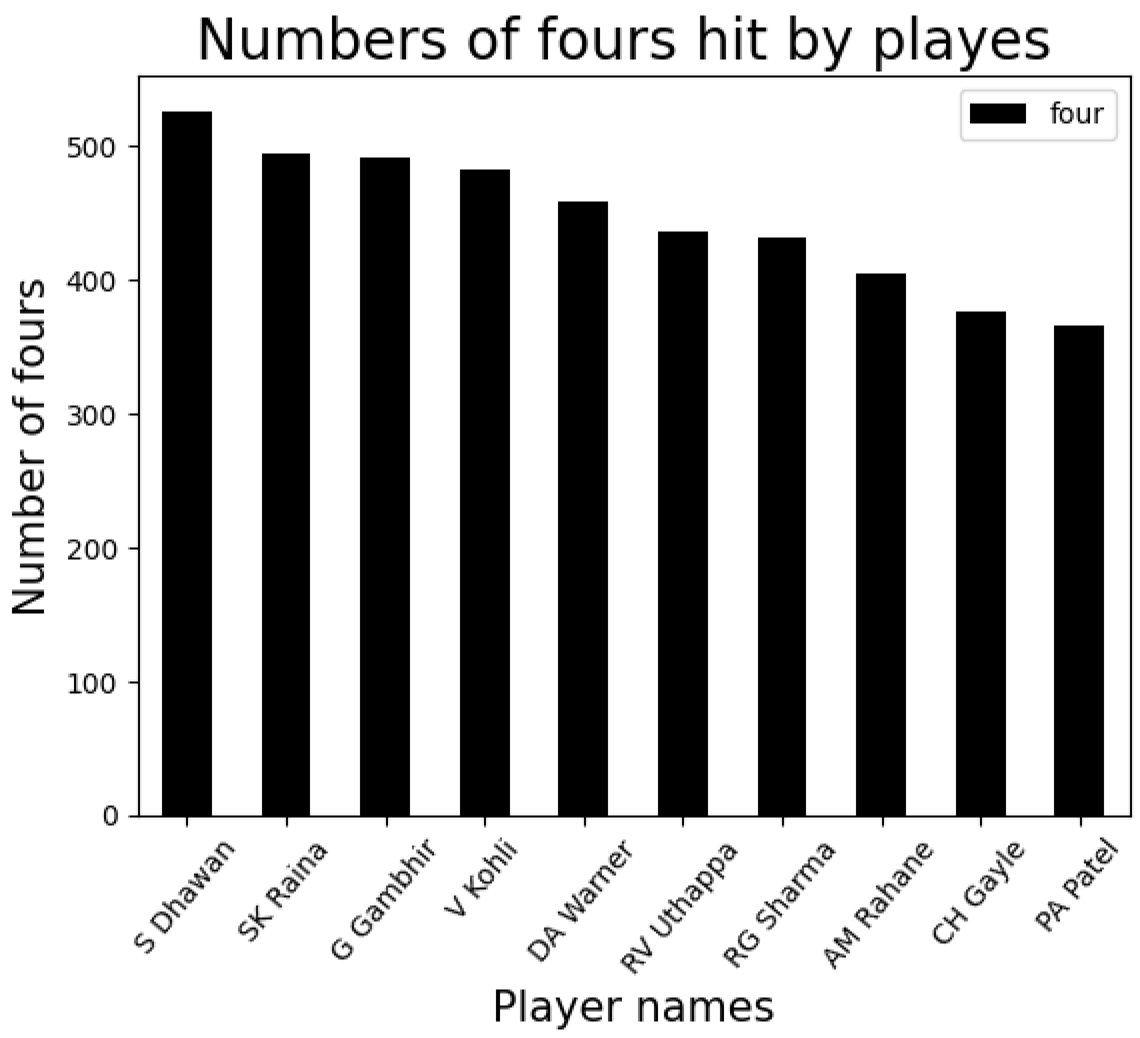
Out[82]:

| batting_team | runs by fours | fours |
|---|---|---|
| Chennai Super Kings | 8772 | 2193 |
| Deccan Chargers | 3828 | 957 |
| Delhi Capitals | 968 | 242 |
| Delhi Daredevils | 8632 | 2158 |
| Gujarat Lions | 1840 | 460 |
| Kings XI Punjab | 9832 | 2458 |
| Kochi Tuskers Kerala | 680 | 170 |
| Kolkata Knight Riders | 9736 | 2434 |
| Mumbai Indians | 10352 | 2588 |
| Pune Warriors | 2100 | 525 |
| Rajasthan Royals | 8140 | 2035 |
| Rising Pune Supergiant | 788 | 197 |

| batting_team | runs by fours | fours |
|---|---|---|
| Rising Pune Supergiants | 684 | 171 |
| Royal Challengers Bangalore | 9440 | 2360 |
| Sunrisers Hyderabad | 5776 | 1444 |

In [83]:
```
# The fours hit by players

batsman_four = four_data.groupby('batsman')['batsman_runs'].agg([('four','count')]).reset_

a = batsman_four.iloc[:10,:].plot('batsman','four',kind='bar',color='black')
plt.title("Numbers of fours hit by playes ",fontsize=20)
plt.xticks(rotation=50)
plt.xlabel("Player names",fontsize=15)
plt.ylabel("Number of fours",fontsize=15)
```

Out[83]:
```
Text(0, 0.5, 'Number of fours')
```



In [84]:
```
#The total runs by the sixes hit and the number of sixes hit by each team

six_data = DF1[DF1['batsman_runs']==6]
a = six_data.groupby('batting_team')['batsman_runs'].agg([('Runs by six','sum'),('sixes','
a
```

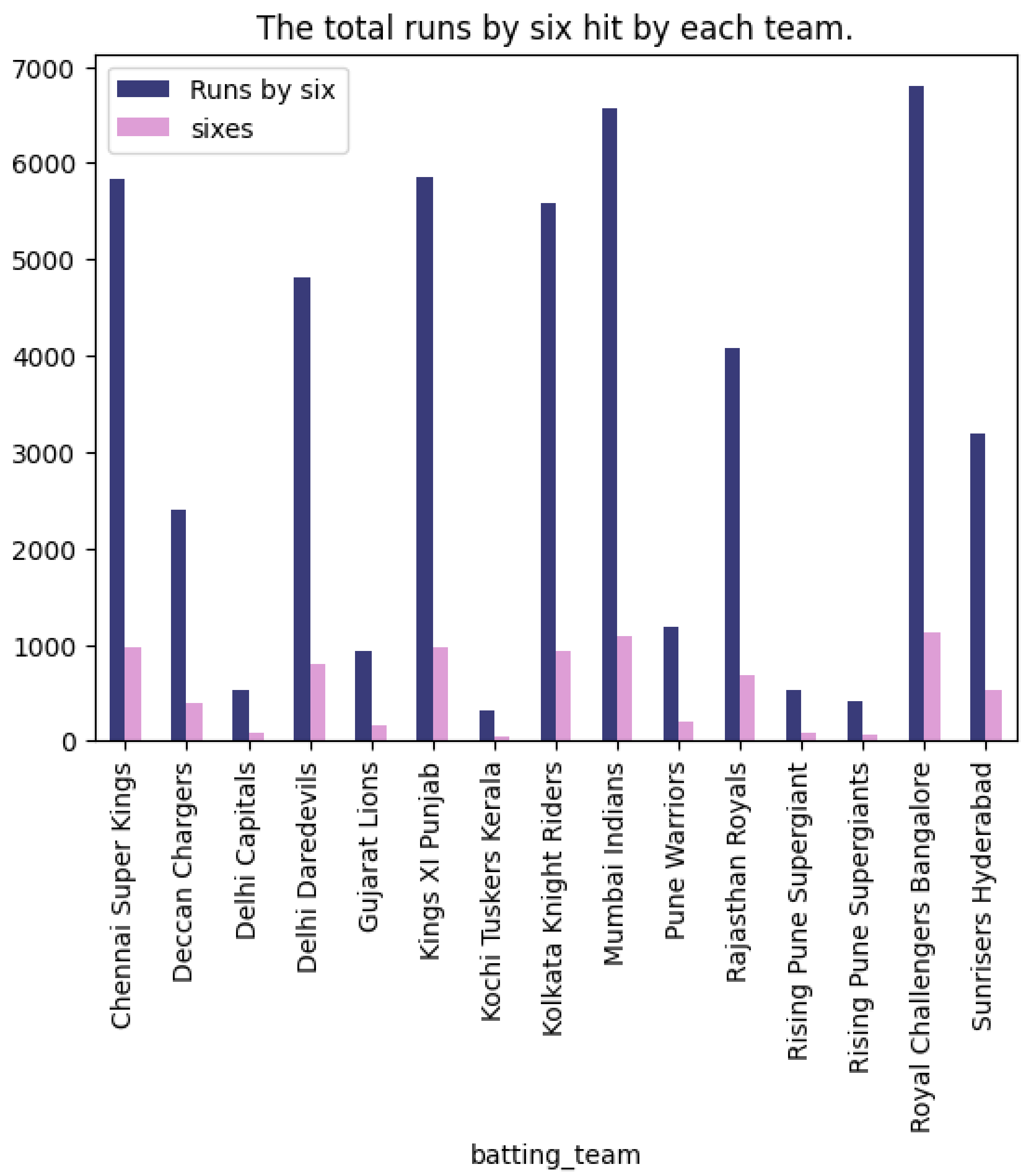
Out[84]:

| | Runs by six | sixes |
|---|---|---|

| batting_team | Runs by six | sixes |
|---|---|---|
| **batting_team** | | |
| **Chennai Super Kings** | 5838 | 973 |
| **Deccan Chargers** | 2400 | 400 |
| **Delhi Capitals** | 522 | 87 |
| **Delhi Daredevils** | 4806 | 801 |
| **Gujarat Lions** | 930 | 155 |
| **Kings XI Punjab** | 5856 | 976 |
| **Kochi Tuskers Kerala** | 318 | 53 |
| **Kolkata Knight Riders** | 5580 | 930 |
| **Mumbai Indians** | 6576 | 1096 |
| **Pune Warriors** | 1176 | 196 |
| **Rajasthan Royals** | 4086 | 681 |
| **Rising Pune Supergiant** | 534 | 89 |
| **Rising Pune Supergiants** | 408 | 68 |
| **Royal Challengers Bangalore** | 6792 | 1132 |
| **Sunrisers Hyderabad** | 3198 | 533 |

In [85]:
```
plt.figure(figsize =(12,8))
a.plot(kind="bar",colormap ='tab20b')
plt.xlabel=('batting team')
plt.ylabel=('Number of six')
plt.title("The total runs by six hit by each team.")
```

Out[85]:
```
Text(0.5, 1.0, 'The total runs by six hit by each team.')

<Figure size 1200x800 with 0 Axes>
```



In [86]:

```
# The number of sixes hit in each season
batsman_six = DF1[DF1["batsman_runs"] == 6]
a=batsman_six.groupby("batsman")["batsman_runs"].agg([("six","count")]).reset_index()
a
```

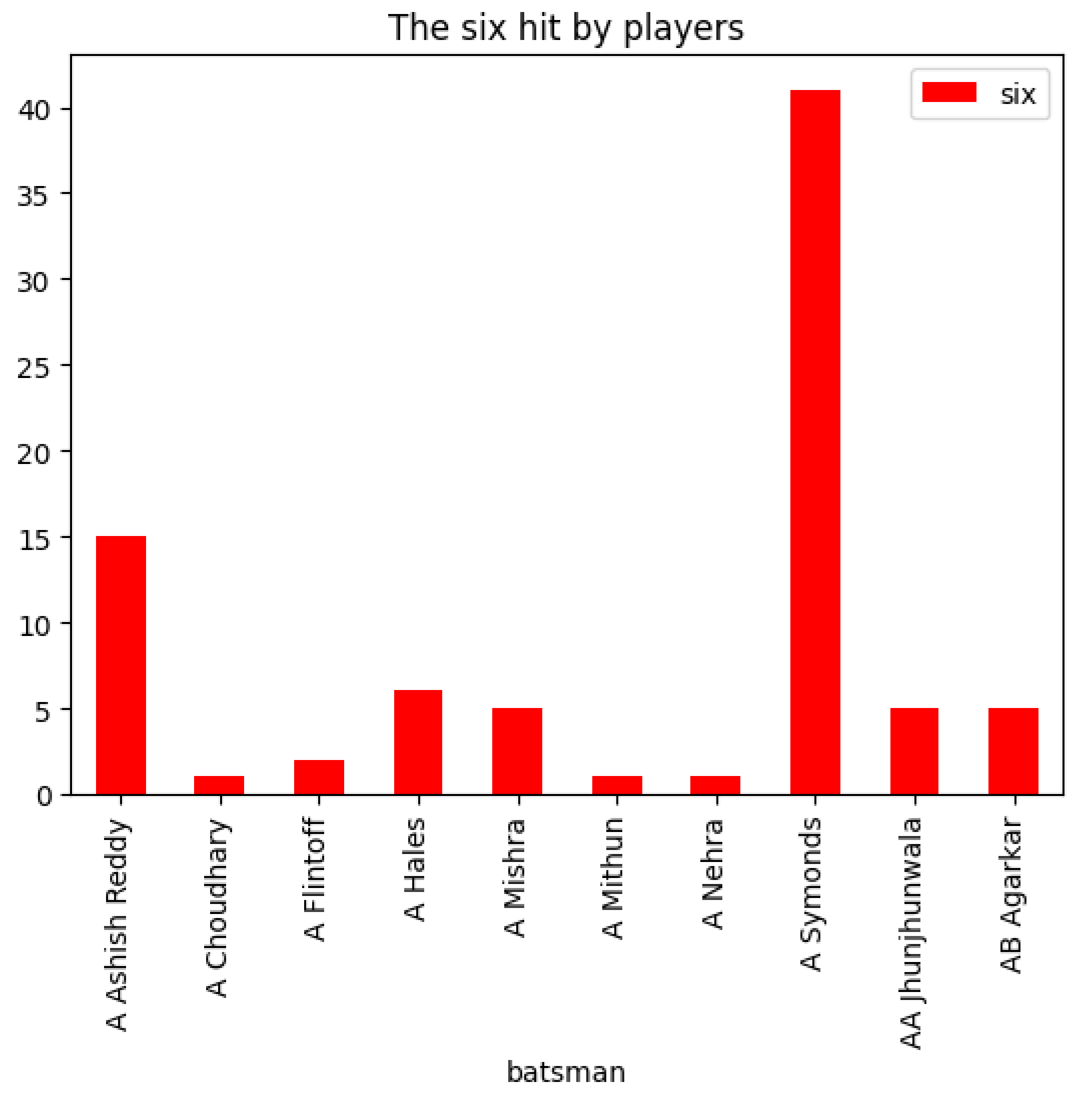
Out[86]:

| | batsman | six |
|---|---|---|
| **0** | A Ashish Reddy | 15 |
| **1** | A Choudhary | 1 |
| **2** | A Flintoff | 2 |
| **3** | A Hales | 6 |
| **4** | A Mishra | 5 |
| **...** | ... | ... |
| **331** | Y Venugopal Rao | 37 |
| **332** | YK Pathan | 161 |
| **333** | YV Takawale | 3 |

|  | batsman | six |
|---|---|---|
| **334** | Yuvraj Singh | 149 |
| **335** | Z Khan | 2 |

336 rows × 2 columns

In [87]:
```
b = a.iloc[:10,:].plot('batsman','six',kind ='bar',color='Red')
plt.xlabel = ('Player name')
plt.ylabel= ('Number of six')
plt.title("The six hit by players")
plt.show()
```



CONCLUSION In the final analysis of my Python project on the Indian Premier League, the integration of NumPy, Pandas, and Seaborn facilitated a detailed exploration of cricketing data. Through compelling visualizations, the project unveiled trends, player contributions, and team dynamics, offering a sophisticated perspective on the IPL. The utilization of data science tools not only enhanced the project's depth but also underscored the role of statistical insights in shaping strategies for teams and players alike. This project stands as a testament to the impactful intersection of technology and sports analytics in decoding the thrilling narrative of the IPL.