In [1]:
```
import pandas as pd
ipl_matches_df = pd.read_csv('matches.csv')
```

In [2]:
```
ipl_matches_df
```

Out[2]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0	Royal Challengers Bangalore	15	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
751	11347	2019	Mumbai	05/05/19	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	normal	0	Mumbai Indians	0	
752	11412	2019	Chennai	07/05/19	Chennai Super Kings	Mumbai Indians	Chennai Super Kings	bat	normal	0	Mumbai Indians	0	
753	11413	2019	Visakhapatnam	08/05/19	Sunrisers Hyderabad	Delhi Capitals	Delhi Capitals	field	normal	0	Delhi Capitals	0	
754	11414	2019	Visakhapatnam	10/05/19	Delhi Capitals	Chennai Super Kings	Chennai Super Kings	field	normal	0	Chennai Super Kings	0	
755	11415	2019	Hyderabad	12/05/19	Mumbai Indians	Chennai Super Kings	Mumbai Indians	bat	normal	0	Mumbai Indians	1	

756 rows × 18 columns

In [3]:
```
ipl_matches_df.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 [4]:
```
ipl_matches_df.describe()
```

Out[4]:

	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 [5]:
```
ipl_matches_df.columns
```

Out[5]:
```
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 [6]:
```
#the first index that doesn't contain a NaN value
ipl_matches_df.umpire3.first_valid_index()
```

Out[6]:
```
636
```

In [7]:
```
#Confirming the first valid index
ipl_matches_df.loc[633:640]
```

Out[7]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_w
633	634	2016	Delhi	2016-05-25	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Sunrisers Hyderabad	22	
634	635	2016	Delhi	2016-05-27	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal	0	Sunrisers Hyderabad	0	
635	636	2016	Bangalore	2016-05-29	Sunrisers Hyderabad	Royal Challengers Bangalore	Sunrisers Hyderabad	bat	normal	0	Sunrisers Hyderabad	8	
636	7894	2018	Mumbai	07/04/18	Mumbai Indians	Chennai Super Kings	Chennai Super Kings	field	normal	0	Chennai Super Kings	0	
637	7895	2018	Mohali	08/04/18	Delhi Daredevils	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	
638	7896	2018	Kolkata	08/04/18	Royal Challengers Bangalore	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	
639	7897	2018	Hyderabad	09/04/18	Rajasthan Royals	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal	0	Sunrisers Hyderabad	0	
640	7898	2018	Chennai	10/04/18	Kolkata Knight Riders	Chennai Super Kings	Chennai Super Kings	field	normal	0	Chennai Super Kings	0	

In [8]:
```
ipl_matches_df.isnull().sum()
```

Out[8]:
```
id                  0
season              0
city                7
date                0
team1               0
team2               0
toss_winner         0
toss_decision       0
result              0
dl_applied          0
winner              4
win_by_runs         0
win_by_wickets      0
player_of_match     4
venue               0
umpire1             2
umpire2             2
umpire3           637
dtype: int64
```

In [9]:
```
ipl_matches_df = ipl_matches_df.drop(columns=['umpire3'], axis=1)
```

In [10]:
```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
sns.color_palette("Paired")
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (12, 8)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

In [11]:
```
teams_per_season = ipl_matches_df.groupby('season')['winner'].value_counts()
teams_per_season
```

Out[11]:
```
season  winner
2008    Rajasthan Royals               13
        Kings XI Punjab                10
        Chennai Super Kings             9
        Delhi Daredevils                7
        Mumbai Indians                  7
                                       ..
2019    Kings XI Punjab                 6
        Kolkata Knight Riders           6
        Sunrisers Hyderabad             6
        Rajasthan Royals                5
        Royal Challengers Bangalore     5
Name: winner, Length: 100, dtype: int64
```

In [12]:
```
"""
for i, v in win_per_season.iteritems():
    print(i, v)

for items in win_per_season.iteritems():
    print(items)
"""
year = 2008
win_per_season_df = pd.DataFrame(columns=['year', 'team', 'wins'])
for items in teams_per_season.iteritems():
    if items[0][0]==year:
        print(items)
        win_series = pd.DataFrame({
            'year': [items[0][0]],
            'team': [items[0][1]],
            'wins': [items[1]]
        })
        win_per_season_df = win_per_season_df.append(win_series)
        year += 1
```

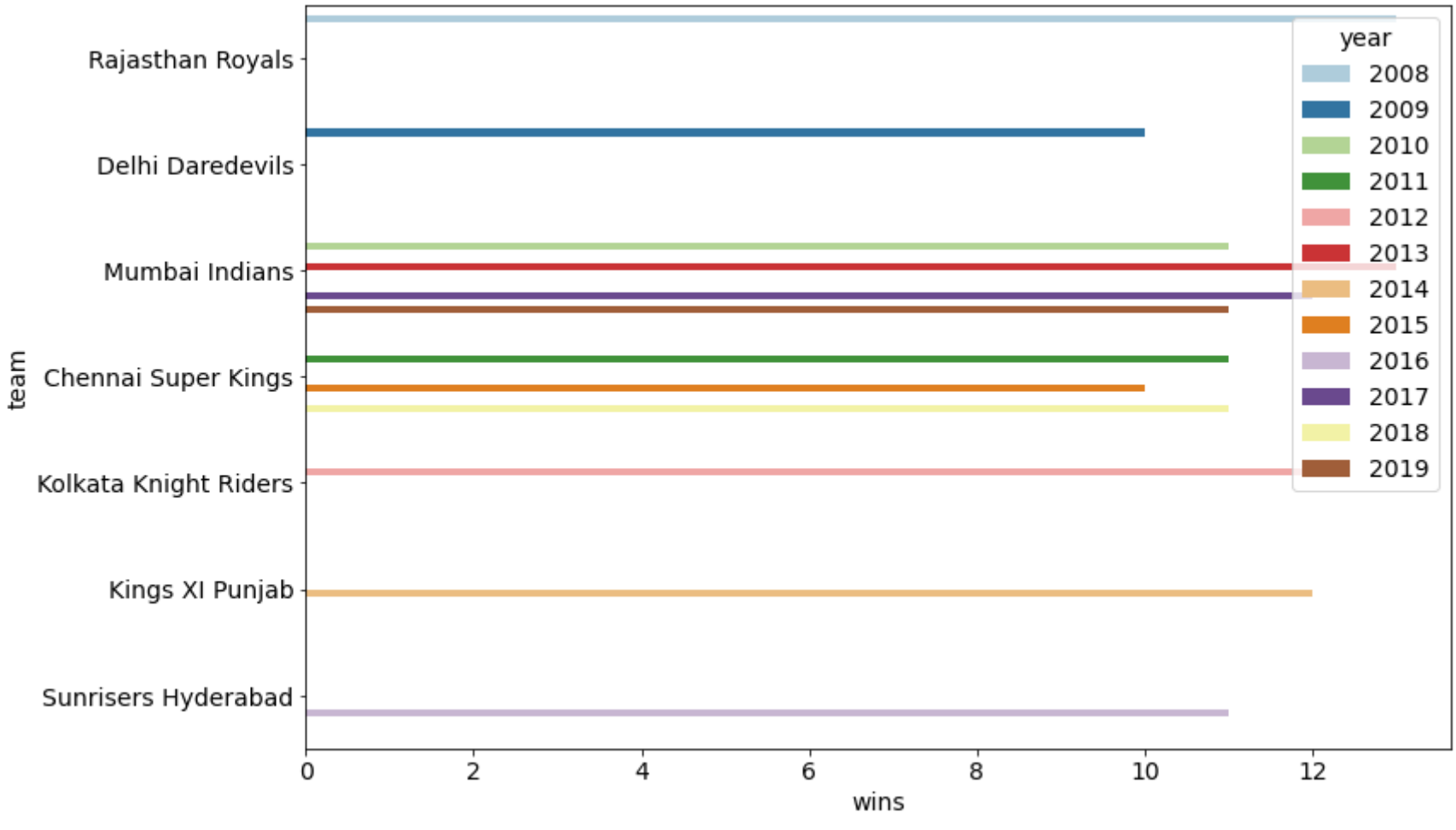
```
((2008, 'Rajasthan Royals'), 13)
((2009, 'Delhi Daredevils'), 10)
((2010, 'Mumbai Indians'), 11)
((2011, 'Chennai Super Kings'), 11)
((2012, 'Kolkata Knight Riders'), 12)
((2013, 'Mumbai Indians'), 13)
((2014, 'Kings XI Punjab'), 12)
((2015, 'Chennai Super Kings'), 10)
((2016, 'Sunrisers Hyderabad'), 11)
((2017, 'Mumbai Indians'), 12)
((2018, 'Chennai Super Kings'), 11)
((2019, 'Mumbai Indians'), 11)
```

In [13]:
```
win_per_season_df
```

Out[13]:

	year	team	wins
0	2008	Rajasthan Royals	13
0	2009	Delhi Daredevils	10
0	2010	Mumbai Indians	11
0	2011	Chennai Super Kings	11
0	2012	Kolkata Knight Riders	12
0	2013	Mumbai Indians	13
0	2014	Kings XI Punjab	12
0	2015	Chennai Super Kings	10
0	2016	Sunrisers Hyderabad	11
0	2017	Mumbai Indians	12
0	2018	Chennai Super Kings	11
0	2019	Mumbai Indians	11

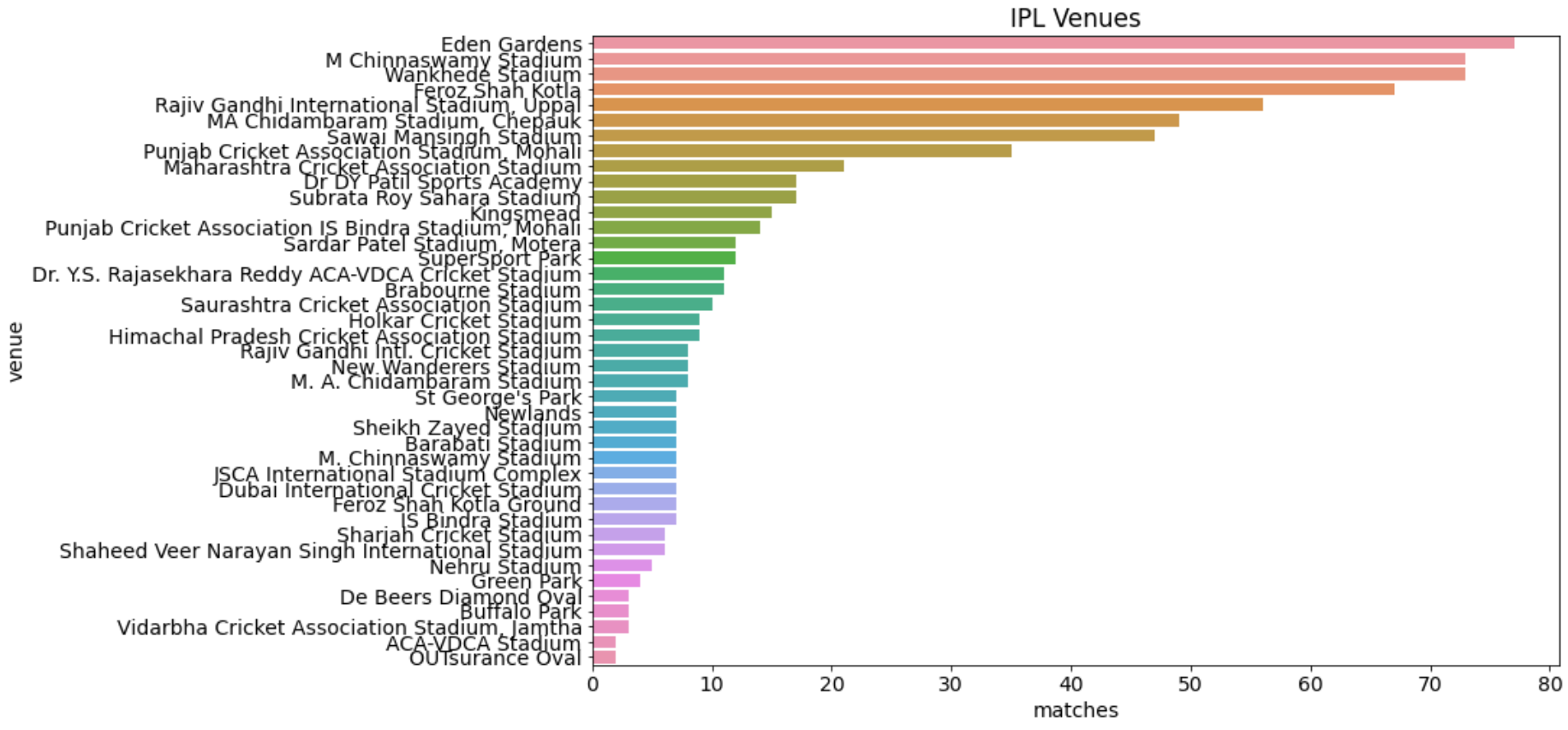
In [14]:
```
sns.barplot('wins', 'team', hue='year', data=win_per_season_df, palette='Paired');
```



In [15]:
```
venue_ser = ipl_matches_df['venue'].value_counts()
```

In [16]:
```
venue_df = pd.DataFrame(columns=['venue', 'matches'])
for items in venue_ser.iteritems():
    temp_df = pd.DataFrame({
        'venue':[items[0]],
        'matches':[items[1]]
    })
    venue_df = venue_df.append(temp_df, ignore_index=True)
```

In [17]:
```
plt.title("IPL Venues")
sns.barplot(x='matches', y='venue', data=venue_df);
```



In [18]:
```
venue_df
```

Out[18]:

	venue	matches
0	Eden Gardens	77
1	M Chinnaswamy Stadium	73
2	Wankhede Stadium	73
3	Feroz Shah Kotla	67
4	Rajiv Gandhi International Stadium, Uppal	56
5	MA Chidambaram Stadium, Chepauk	49
6	Sawai Mansingh Stadium	47
7	Punjab Cricket Association Stadium, Mohali	35
8	Maharashtra Cricket Association Stadium	21
9	Dr DY Patil Sports Academy	17
10	Subrata Roy Sahara Stadium	17
11	Kingsmead	15
12	Punjab Cricket Association IS Bindra Stadium, ...	14
13	Sardar Patel Stadium, Motera	12
14	SuperSport Park	12
15	Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket St...	11
16	Brabourne Stadium	11
17	Saurashtra Cricket Association Stadium	10
18	Holkar Cricket Stadium	9
19	Himachal Pradesh Cricket Association Stadium	9
20	Rajiv Gandhi Intl. Cricket Stadium	8
21	New Wanderers Stadium	8
22	M. A. Chidambaram Stadium	8
23	St George's Park	7
24	Newlands	7
25	Sheikh Zayed Stadium	7
26	Barabati Stadium	7
27	M. Chinnaswamy Stadium	7
28	JSCA International Stadium Complex	7
29	Dubai International Cricket Stadium	7
30	Feroz Shah Kotla Ground	7
31	IS Bindra Stadium	7
32	Sharjah Cricket Stadium	6
33	Shaheed Veer Narayan Singh International Stadium	6
34	Nehru Stadium	5
35	Green Park	4
36	De Beers Diamond Oval	3
37	Buffalo Park	3
38	Vidarbha Cricket Association Stadium, Jamtha	3
39	ACA-VDCA Stadium	2
40	OUTsurance Oval	2

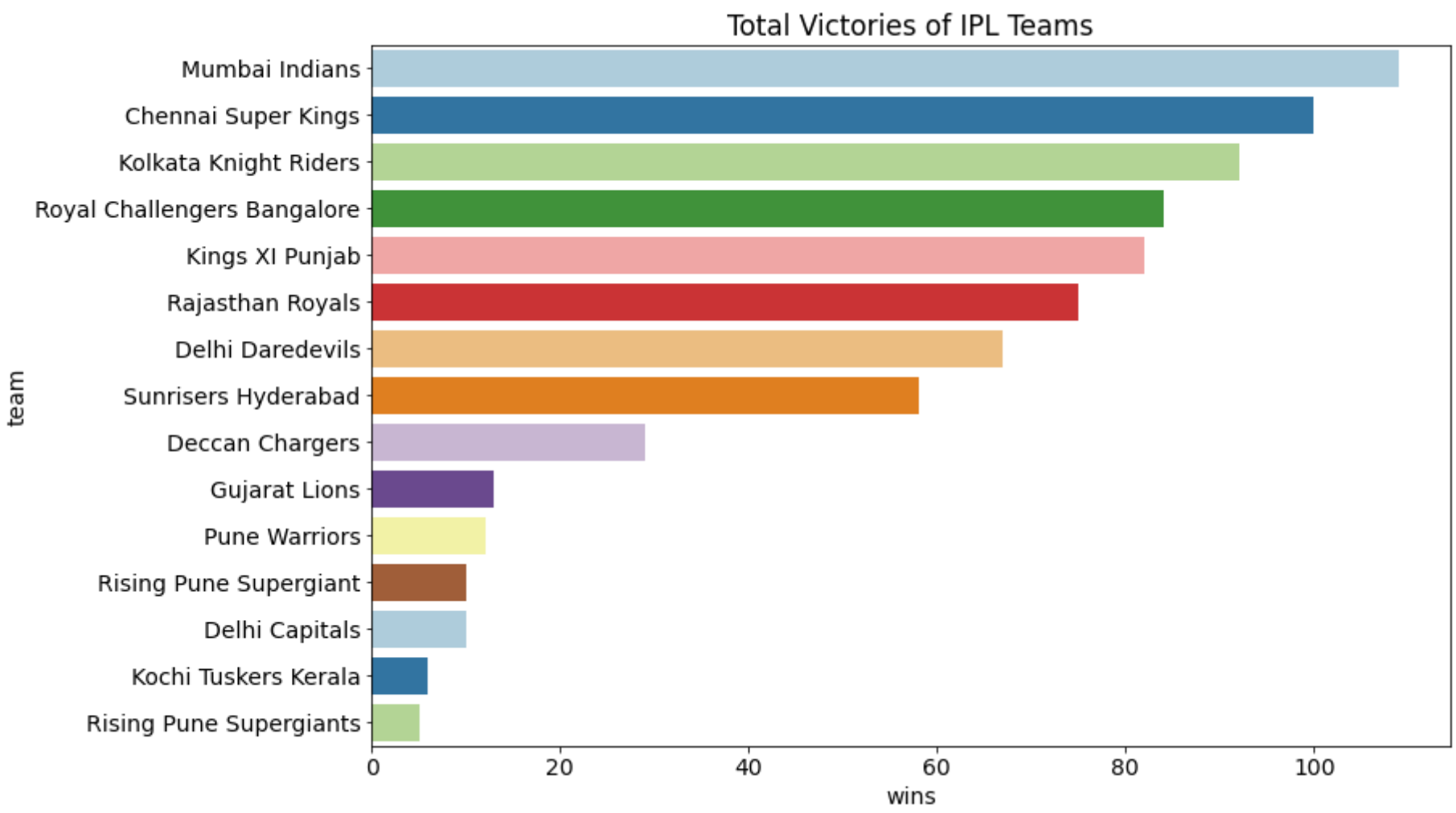
In [19]:
```
team_wins_ser = ipl_matches_df['winner'].value_counts()

team_wins_df = pd.DataFrame(columns=["team", "wins"])
for items in team_wins_ser.iteritems():
    temp_df1 = pd.DataFrame({
        'team':[items[0]],
        'wins':[items[1]]
    })
    team_wins_df = team_wins_df.append(temp_df1, ignore_index=True)
```

In [20]:
```
team_wins_df
```

Out[20]:

	team	wins
0	Mumbai Indians	109
1	Chennai Super Kings	100
2	Kolkata Knight Riders	92
3	Royal Challengers Bangalore	84
4	Kings XI Punjab	82
5	Rajasthan Royals	75
6	Delhi Daredevils	67
7	Sunrisers Hyderabad	58
8	Deccan Chargers	29
9	Gujarat Lions	13
10	Pune Warriors	12
11	Rising Pune Supergiant	10
12	Delhi Capitals	10
13	Kochi Tuskers Kerala	6
14	Rising Pune Supergiants	5

In [21]:
```
plt.title("Total Victories of IPL Teams")
sns.barplot(x='wins', y='team', data=team_wins_df, palette='Paired');
```



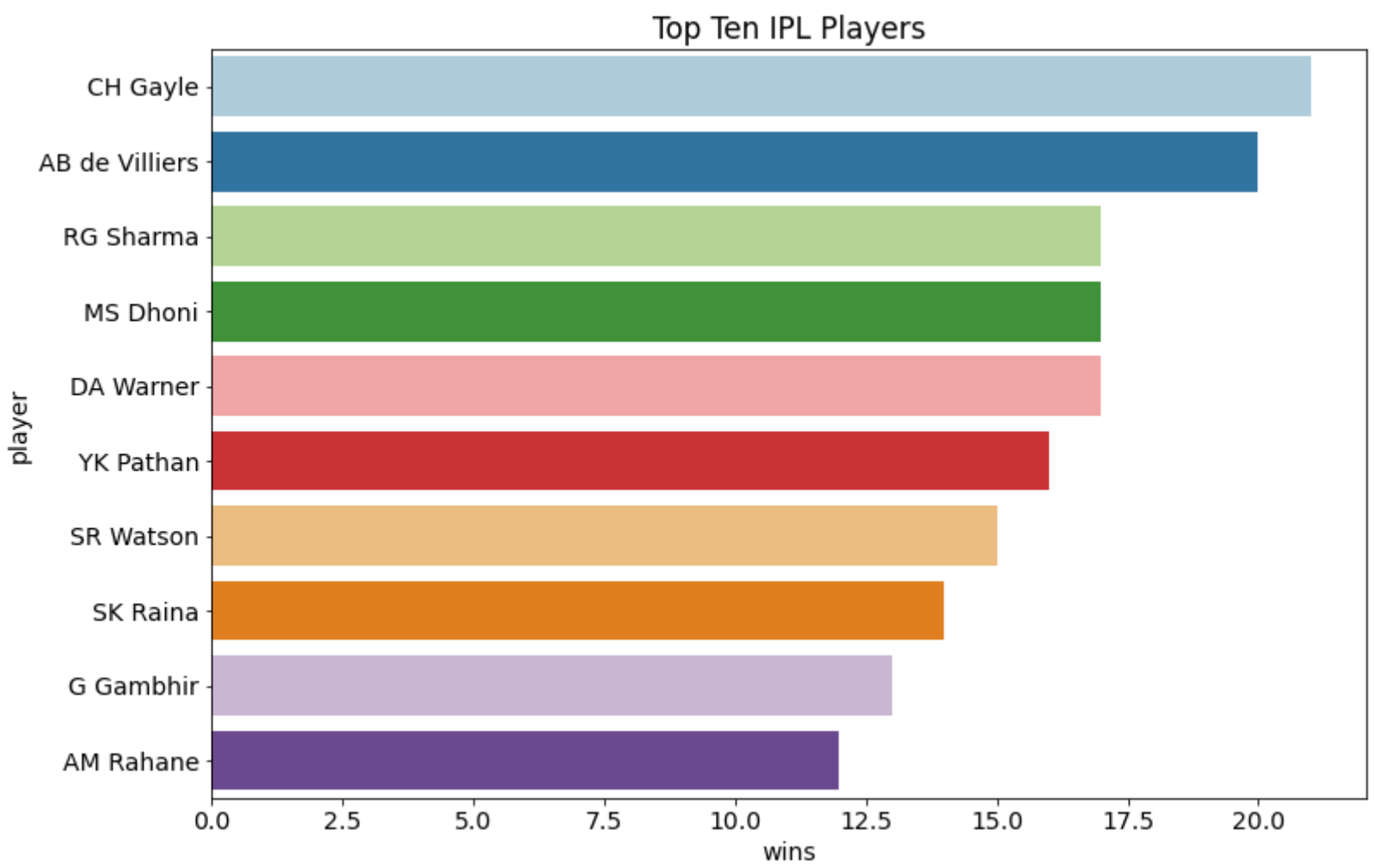
In [22]:
```
mvp_ser = ipl_matches_df['player_of_match'].value_counts()

mvp_ten_df = pd.DataFrame(columns=["player", "wins"])
count = 0
for items in mvp_ser.iteritems():
    if count>9:
        break
    else:
        temp_df2 = pd.DataFrame({
            'player':[items[0]],
            'wins':[items[1]]
        })
        mvp_ten_df = mvp_ten_df.append(temp_df2, ignore_index=True)
        count += 1
```

In [23]:
```
mvp_ten_df
```

Out[23]:

	player	wins
0	CH Gayle	21
1	AB de Villiers	20
2	RG Sharma	17
3	MS Dhoni	17
4	DA Warner	17
5	YK Pathan	16
6	SR Watson	15
7	SK Raina	14
8	G Gambhir	13
9	AM Rahane	12

In [24]:
```
plt.title("Top Ten IPL Players")
sns.barplot(x='wins', y='player', data=mvp_ten_df, palette='Paired');
```



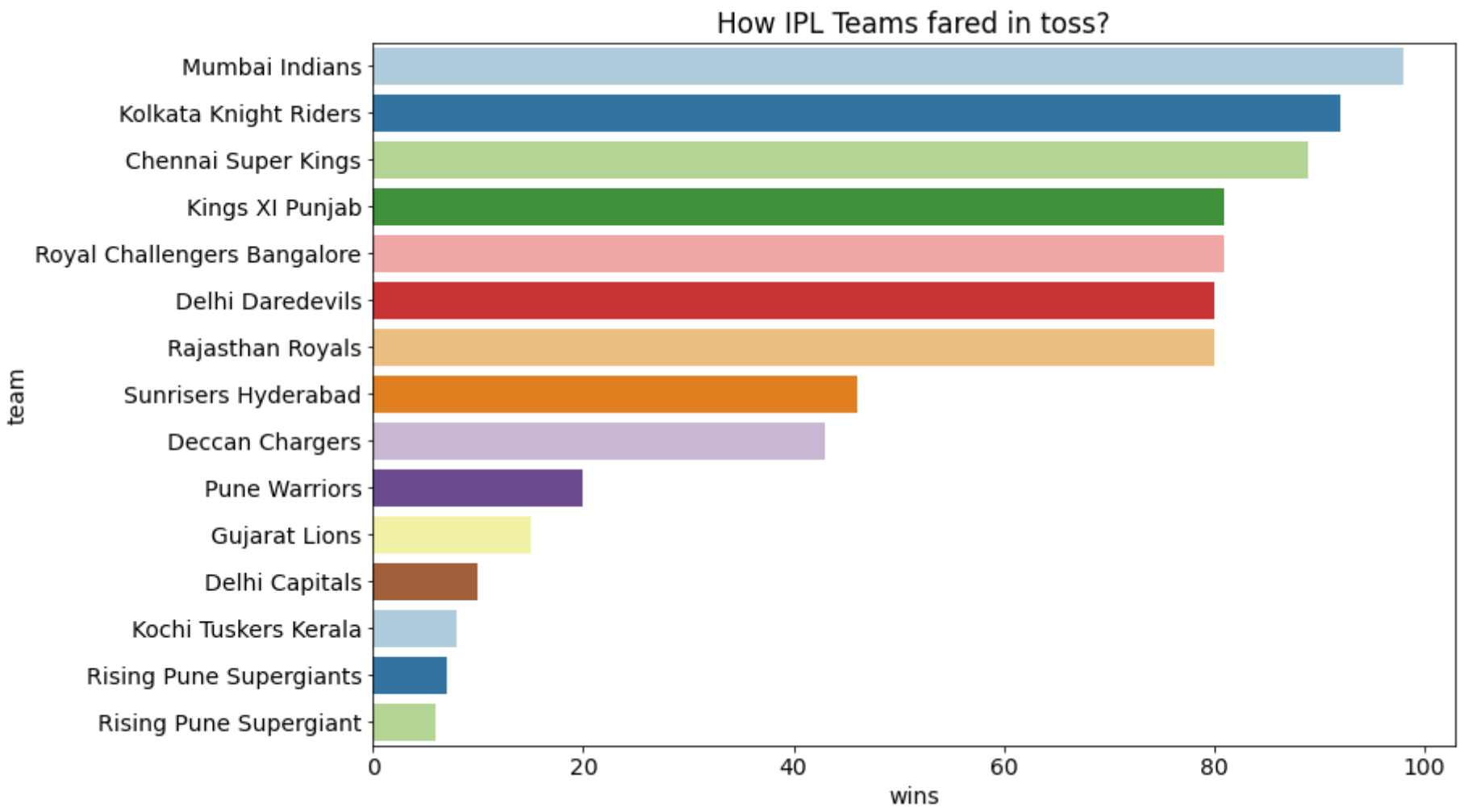
In [25]:
```
toss_ser = ipl_matches_df['toss_winner'].value_counts()

toss_df = pd.DataFrame(columns=["team", "wins"])

for items in toss_ser.iteritems():
    temp_df3 = pd.DataFrame({
        'team':[items[0]],
        'wins':[items[1]]
    })
    toss_df = toss_df.append(temp_df3, ignore_index=True)
```

In [26]:
```
toss_df
```

Out[26]:

	team	wins
0	Mumbai Indians	98
1	Kolkata Knight Riders	92
2	Chennai Super Kings	89
3	Kings XI Punjab	81
4	Royal Challengers Bangalore	81
5	Delhi Daredevils	80
6	Rajasthan Royals	80
7	Sunrisers Hyderabad	46
8	Deccan Chargers	43
9	Pune Warriors	20
10	Gujarat Lions	15
11	Delhi Capitals	10
12	Kochi Tuskers Kerala	8
13	Rising Pune Supergiants	7
14	Rising Pune Supergiant	6

In [27]:
```
plt.title("How IPL Teams fared in toss?")
sns.barplot(x='wins', y='team', data=toss_df, palette='Paired');
```



In [28]:
```
mvp_ten_df
```

Out[28]:

	player	wins
0	CH Gayle	21
1	AB de Villiers	20
2	RG Sharma	17
3	MS Dhoni	17
4	DA Warner	17
5	YK Pathan	16
6	SR Watson	15
7	SK Raina	14
8	G Gambhir	13
9	AM Rahane	12

In [29]:
```
umpire1_ser = ipl_matches_df['umpire1'].value_counts()
umpire2_ser = ipl_matches_df['umpire2'].value_counts()
```

In [30]:
```
umpires_df = pd.concat([umpire1_ser, umpire2_ser], axis=1)
umpires_df
```

Out[30]:

	umpire1	umpire2
HDPK Dharmasena	73.0	14.0
Asad Rauf	51.0	NaN
S Ravi	49.0	57.0
AK Chaudhary	43.0	15.0
Aleem Dar	38.0	NaN
...	...	...
K Srinivasan	NaN	3.0
KN Anantapadmanabhan	NaN	3.0
SD Ranade	NaN	2.0
Nand Kishore	NaN	1.0
Subroto Das	NaN	1.0

75 rows × 2 columns

In [31]:
```
umpire_ser = umpires_df.sum(axis=1)

umpire_df = pd.DataFrame(columns=["umpire", "matches"])

for items in umpire_ser.iteritems():
    temp_df4 = pd.DataFrame({
        'umpire':[items[0]],
        'matches':[items[1]]
    })
    umpire_df= umpire_df.append(temp_df4, ignore_index=True)
```

In [32]:
```
umpire_df.sort_values('matches', ascending=False).head()
```

Out[32]:

	umpire	matches
2	S Ravi	106.0
0	HDPK Dharmasena	87.0
11	C Shamshuddin	73.0
3	AK Chaudhary	58.0
56	SJA Taufel	55.0

In [33]:
```
win_per_season_df
```

Out[33]:

	year	team	wins
0	2008	Rajasthan Royals	13
0	2009	Delhi Daredevils	10
0	2010	Mumbai Indians	11
0	2011	Chennai Super Kings	11
0	2012	Kolkata Knight Riders	12
0	2013	Mumbai Indians	13
0	2014	Kings XI Punjab	12
0	2015	Chennai Super Kings	10
0	2016	Sunrisers Hyderabad	11
0	2017	Mumbai Indians	12
0	2018	Chennai Super Kings	11
0	2019	Mumbai Indians	11

In [34]:
```
team_wins_df
```

Out[34]:

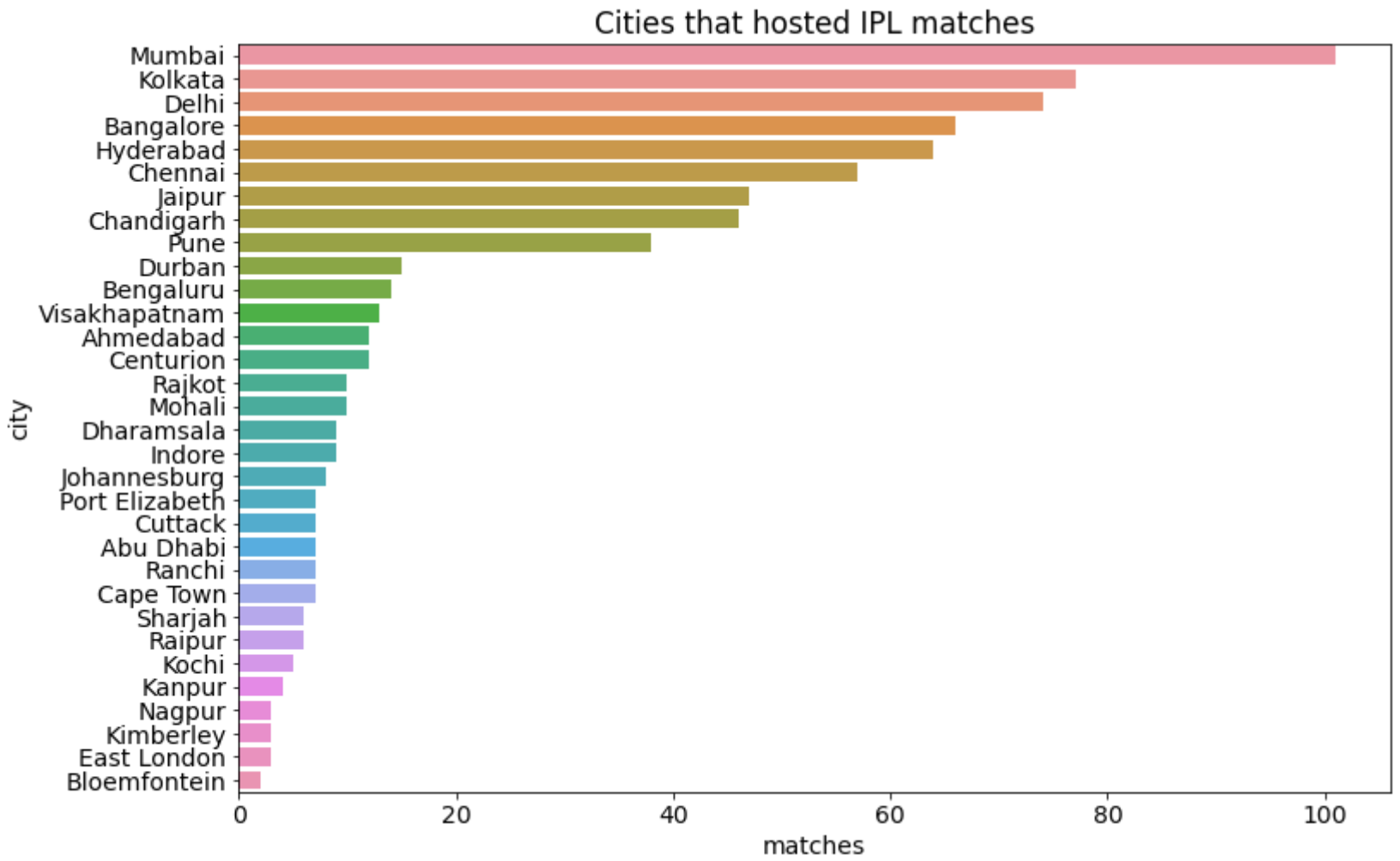
	team	wins
0	Mumbai Indians	109
1	Chennai Super Kings	100
2	Kolkata Knight Riders	92
3	Royal Challengers Bangalore	84
4	Kings XI Punjab	82
5	Rajasthan Royals	75
6	Delhi Daredevils	67
7	Sunrisers Hyderabad	58
8	Deccan Chargers	29
9	Gujarat Lions	13
10	Pune Warriors	12
11	Rising Pune Supergiant	10
12	Delhi Capitals	10
13	Kochi Tuskers Kerala	6
14	Rising Pune Supergiants	5

In [35]:
```
ipl_matches_df['city'].value_counts()
```

Out[35]:
```
Mumbai            101
Kolkata            77
Delhi              74
Bangalore          66
Hyderabad          64
Chennai            57
Jaipur             47
Chandigarh         46
Pune               38
Durban             15
Bengaluru          14
Visakhapatnam      13
Ahmedabad          12
Centurion          12
Rajkot             10
Mohali             10
Dharamsala          9
Indore              9
Johannesburg        8
Port Elizabeth      7
Cuttack             7
Abu Dhabi           7
Ranchi              7
Cape Town           7
Sharjah             6
Raipur              6
Kochi               5
Kanpur              4
Nagpur              3
Kimberley           3
East London         3
Bloemfontein        2
Name: city, dtype: int64
```

In [36]:
```
city_ser = ipl_matches_df['city'].value_counts()

city_df = pd.DataFrame(columns=['city', 'matches'])
for items in city_ser.iteritems():
    temp_df6 = pd.DataFrame({
        'city':[items[0]],
        'matches':[items[1]]
    })
    city_df = city_df.append(temp_df6, ignore_index=True)
```

In [37]:
```
plt.title("Cities that hosted IPL matches")
sns.barplot(x='matches', y='city', data=city_df);
```



In [38]:
```
win_count = 0
for index, value in ipl_matches_df.iterrows():
    if(value['toss_winner']==value['winner']):
#         print(value['winner'])
        win_count += 1

print(f'The number of times the team winning toss have won: {win_count}')
prob = win_count/len(ipl_matches_df)
print('The probability of winning if won the toss: {:.2f}' .format(prob))
```

```
The number of times the team winning toss have won: 393
The probability of winning if won the toss: 0.52
```

In [39]:
```
len(ipl_matches_df)
```

Out[39]:
```
756
```

In [40]:
```
defend_vict_ser = ipl_matches_df['win_by_runs'].value_counts()
defend_vict_ser.sort_values(ascending=True)
```

Out[40]:
```
146      1
98       1
77       1
93       1
102      1
      ...
1       10
10      11
4       11
14      13
0      419
Name: win_by_runs, Length: 89, dtype: int64
```

In [41]:
```
score = 146
for index, row in ipl_matches_df.iterrows():
    if(row['win_by_runs'] == score):
        print(row)
```

```
id                              44
season                        2017
city                         Delhi
date                    2017-05-06
team1              Mumbai Indians
team2            Delhi Daredevils
toss_winner      Delhi Daredevils
toss_decision               field
result                     normal
dl_applied                      0
winner             Mumbai Indians
win_by_runs                   146
win_by_wickets                  0
player_of_match       LMP Simmons
venue             Feroz Shah Kotla
umpire1                Nitin Menon
umpire2                  CK Nandan
Name: 43, dtype: object
```

In [42]:
```
chasing_vict_ser = ipl_matches_df['win_by_wickets'].value_counts()
chasing_vict_ser
```

Out[42]:
```
0     350
6      85
7      80
5      71
8      54
4      41
9      37
3      18
10     11
2       6
1       3
Name: win_by_wickets, dtype: int64
```

In [43]:
```
for index, row in ipl_matches_df.iterrows():
    if(row['win_by_wickets'] == 10):
        print(row)
```

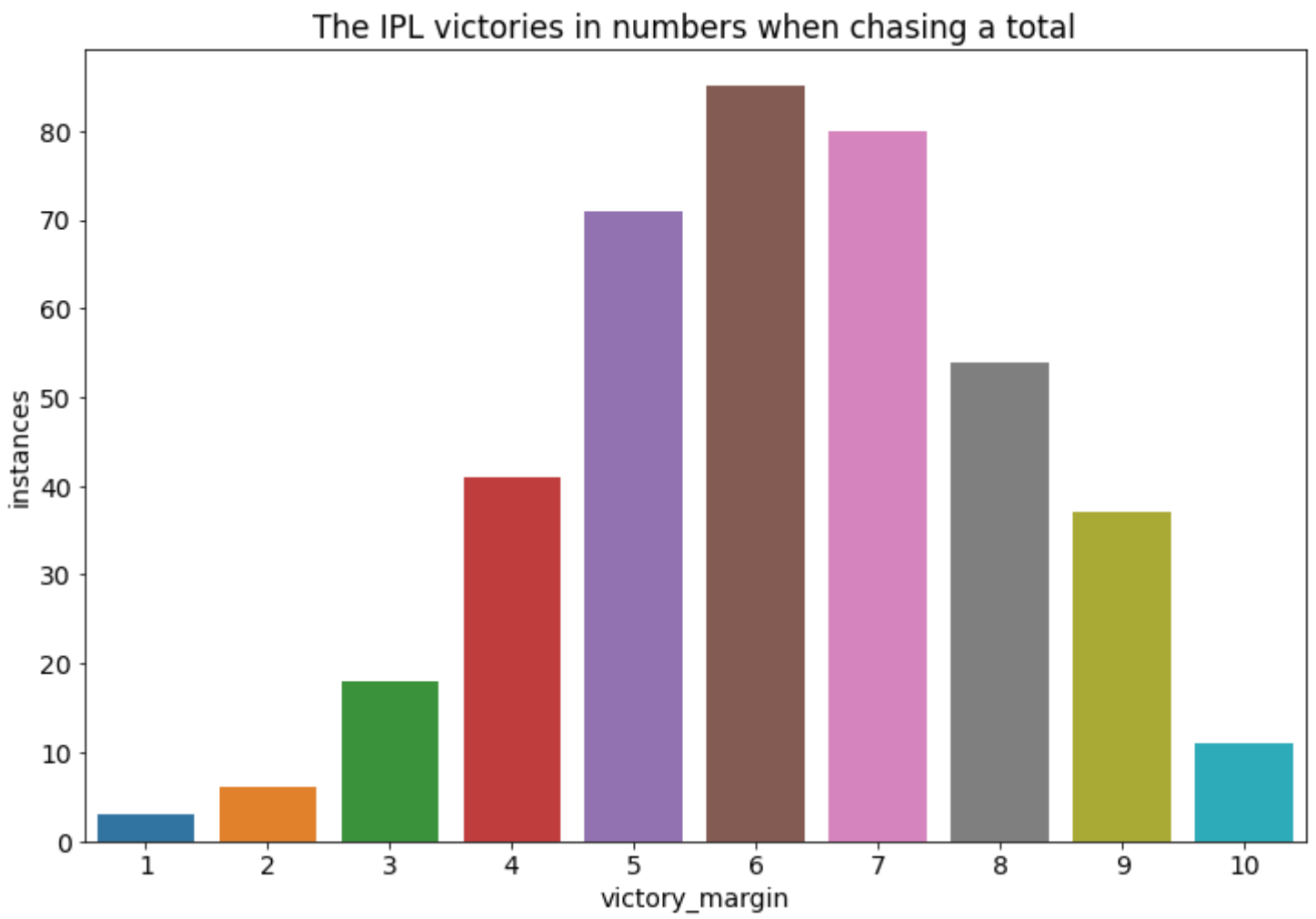
```
id                                                3
season                                         2017
city                                         Rajkot
date                                     2017-04-07
team1                                  Gujarat Lions
team2                        Kolkata Knight Riders
toss_winner                  Kolkata Knight Riders
toss_decision                                field
result                                      normal
dl_applied                                       0
winner                       Kolkata Knight Riders
win_by_runs                                      0
win_by_wickets                                  10
player_of_match                             CA Lynn
venue              Saurashtra Cricket Association Stadium
umpire1                                  Nitin Menon
umpire2                                    CK Nandan
Name: 2, dtype: object
id                                               35
```

In [44]:
```
chasing_vict_df = pd.DataFrame(columns=['victory_margin', 'instances'])

for items in chasing_vict_ser.iteritems():
    temp_df7 = pd.DataFrame({
        'victory_margin': [items[0]],
        'instances': [items[1]]
    })
    chasing_vict_df = chasing_vict_df.append(temp_df7, ignore_index=True)

#to drop the first row as it represents victory margin of zero wickets(victory by defending a total)
chasing_vict_df2 = chasing_vict_df.drop([0])
chasing_vict_df2
```

Out[44]:

	victory_margin	instances
1	6	85
2	7	80
3	5	71
4	8	54
5	4	41
6	9	37
7	3	18
8	10	11
9	2	6
10	1	3

In [45]:
```
plt.title('The IPL victories in numbers when chasing a total')
sns.barplot(x='victory_margin', y='instances', data=chasing_vict_df2);
```



In [ ]:
```

```