IPL Data Analysis

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
match_data = pd.read_csv("C:\\Users\\chitr\\Desktop\\IPL Python\\IPL Dataset and
ball_data =  pd.read_csv("C:\\Users\\chitr\\Desktop\\IPL Python\\IPL Dataset and
```

In [3]:
```
match_data.head() #Printing the top 5 rows
```

Out[3]:

| ıue | neutral_venue | team1 | team2 | toss_winner | toss_decision | winner | result | result_ı |
|---|---|---|---|---|---|---|---|---|
| M amy ium | 0 | Royal Challengers Bangalore | Kolkata Knight Riders | Royal Challengers Bangalore | field | Kolkata Knight Riders | runs | |
| ıjab :ket tion um, hali | 0 | Kings XI Punjab | Chennai Super Kings | Chennai Super Kings | bat | Chennai Super Kings | runs | |
| hah otla | 0 | Delhi Daredevils | Rajasthan Royals | Rajasthan Royals | bat | Delhi Daredevils | wickets | |
| ede ium | 0 | Mumbai Indians | Royal Challengers Bangalore | Mumbai Indians | bat | Royal Challengers Bangalore | wickets | |
| den ens | 0 | Kolkata Knight Riders | Deccan Chargers | Deccan Chargers | bat | Kolkata Knight Riders | wickets | |

In [4]:
```
ball_data.head() #Printing the top 5 rows
```

Out[4]:

| | id | inning | over | ball | batsman | non_striker | bowler | batsman_runs | extra_runs | total_runs |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 335982 | 1 | 6 | 5 | RT Ponting | BB McCullum | AA Noffke | 1 | 0 | 1 |
| **1** | 335982 | 1 | 6 | 6 | BB McCullum | RT Ponting | AA Noffke | 1 | 0 | 1 |
| **2** | 335982 | 1 | 7 | 1 | BB McCullum | RT Ponting | Z Khan | 0 | 0 | 0 |
| **3** | 335982 | 1 | 7 | 2 | BB McCullum | RT Ponting | Z Khan | 1 | 0 | 1 |
| **4** | 335982 | 1 | 7 | 3 | RT Ponting | BB McCullum | Z Khan | 1 | 0 | 1 |

In [83]:
```
match_data.isnull().sum() #Check for null values in data set
```

Out[83]:
```
id                   0
city                13
date                 0
player_of_match      4
venue                0
neutral_venue        0
team1                0
team2                0
toss_winner          0
toss_decision        0
winner               4
result               4
result_margin       17
eliminator           4
method             797
umpire1              0
umpire2              0
Season               0
dtype: int64
```

In [6]:
```
ball_data.isnull().sum() #Check for null values in data set
```

Out[6]:
```
id                       0
inning                   0
over                     0
ball                     0
batsman                  0
non_striker              0
bowler                   0
batsman_runs             0
extra_runs               0
total_runs               0
non_boundary             0
is_wicket                0
dismissal_kind      183973
player_dismissed    183973
fielder             186684
extras_type         183235
batting_team             0
bowling_team           191
dtype: int64
```

In [7]:
```
print('Total matches played:', match_data.shape[0]) #Total matches played
print('\n Cities played at:', match_data['city'].unique()) #unique venues for mat
print('\n Team participated:', match_data['team1'].unique()) #Teams participated
```

```
Total matches played: 816

 Cities played at: ['Bangalore' 'Chandigarh' 'Delhi' 'Mumbai' 'Kolkata' 'Jaipu
r' 'Hyderabad'
 'Chennai' 'Cape Town' 'Port Elizabeth' 'Durban' 'Centurion' 'East London'
 'Johannesburg' 'Kimberley' 'Bloemfontein' 'Ahmedabad' 'Cuttack' 'Nagpur'
 'Dharamsala' 'Kochi' 'Indore' 'Visakhapatnam' 'Pune' 'Raipur' 'Ranchi'
 'Abu Dhabi' nan 'Rajkot' 'Kanpur' 'Bengaluru' 'Dubai' 'Sharjah']

 Team participated: ['Royal Challengers Bangalore' 'Kings XI Punjab' 'Delhi Dar
edevils'
 'Mumbai Indians' 'Kolkata Knight Riders' 'Rajasthan Royals'
 'Deccan Chargers' 'Chennai Super Kings' 'Kochi Tuskers Kerala'
 'Pune Warriors' 'Sunrisers Hyderabad' 'Gujarat Lions'
 'Rising Pune Supergiants' 'Rising Pune Supergiant' 'Delhi Capitals']
```

In [84]:
```
#Extracting year values to get season or Year value

match_data['Season'] = pd.DatetimeIndex(match_data['date']).year
match_data.head()
```

Out[84]:

| | id | city | date | player_of_match | venue | neutral_venue | team1 | team2 |
|---|---|---|---|---|---|---|---|---|
| **0** | 335982 | Bangalore | 18-04-2008 | BB McCullum | M Chinnaswamy Stadium | 0 | Royal Challengers Bangalore | Kolkata Knight Riders |
| **1** | 335983 | Chandigarh | 19-04-2008 | MEK Hussey | Punjab Cricket Association Stadium, Mohali | 0 | Kings XI Punjab | Chennai Super Kings |
| **2** | 335984 | Delhi | 19-04-2008 | MF Maharoof | Feroz Shah Kotla | 0 | Delhi Daredevils | Rajasthan Royals |
| **3** | 335985 | Mumbai | 20-04-2008 | MV Boucher | Wankhede Stadium | 0 | Mumbai Indians | Royal Challengers Bangalore |
| **4** | 335986 | Kolkata | 20-04-2008 | DJ Hussey | Eden Gardens | 0 | Kolkata Knight Riders | Deccan Chargers |

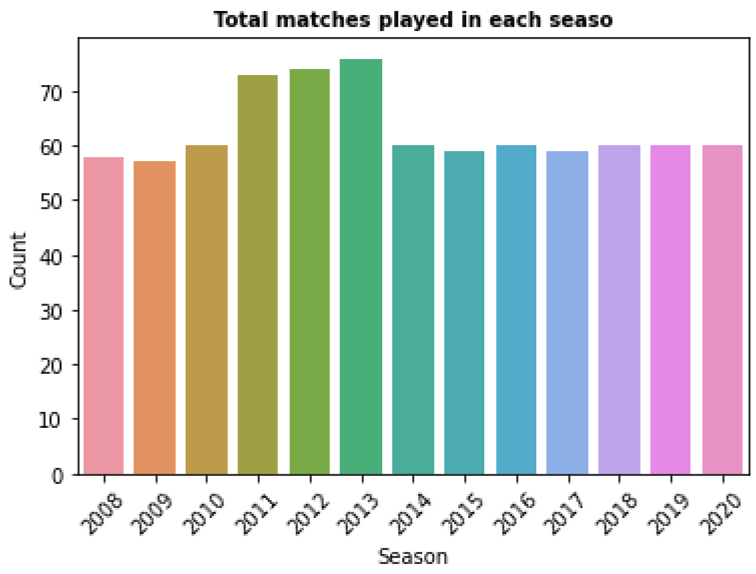
In [85]:
```
#total number of matches played each year

match_per_season = match_data.groupby(['Season'])['id'].count().reset_index().ren
match_per_season
```

Out[85]:

| | Season | matches |
|---|---|---|
| **0** | 2008 | 58 |
| **1** | 2009 | 57 |
| **2** | 2010 | 60 |
| **3** | 2011 | 73 |
| **4** | 2012 | 74 |
| **5** | 2013 | 76 |
| **6** | 2014 | 60 |
| **7** | 2015 | 59 |
| **8** | 2016 | 60 |
| **9** | 2017 | 59 |
| **10** | 2018 | 60 |
| **11** | 2019 | 60 |
| **12** | 2020 | 60 |

In [86]:
```
#plotting bar graph of total matches each year

sns.countplot(match_data['Season'])
plt.xticks(rotation = 45, fontsize = 10)
plt.yticks (fontsize = 10)
plt.xlabel('Season', fontsize = 10)
plt.ylabel('Count', fontsize = 10)
plt.title("Total matches played in each seaso", fontsize = 10, fontweight = "bolc
```

Out[86]:
```
Text(0.5, 1.0, 'Total matches played in each seaso')
```

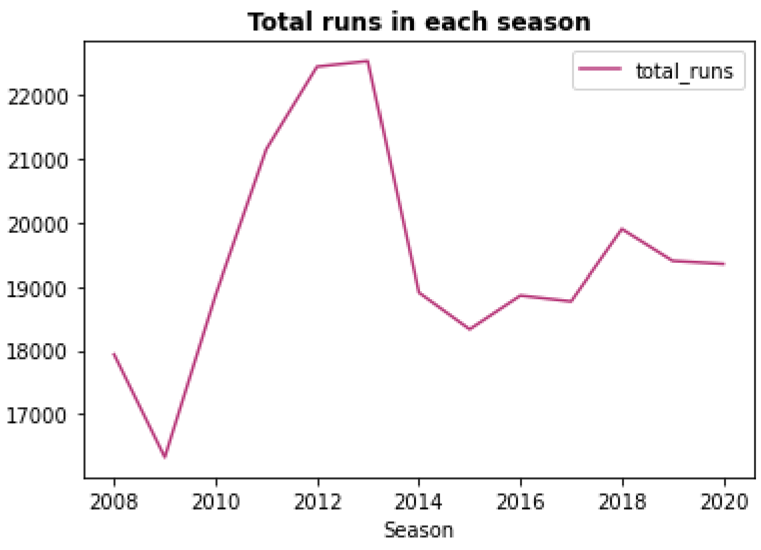


In [87]:
```
season_data=match_data[['id','Season']].merge(ball_data, left_on = 'id', right_on
season_data.head()
```

Out[87]:

| | Season | inning | over | ball | batsman | non_striker | bowler | batsman_runs | extra_runs | total_runs |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2008 | 1 | 6 | 5 | RT Ponting | BB McCullum | AA Noffke | 1 | 0 | 1 |
| **1** | 2008 | 1 | 6 | 6 | BB McCullum | RT Ponting | AA Noffke | 1 | 0 | 1 |
| **2** | 2008 | 1 | 7 | 1 | BB McCullum | RT Ponting | Z Khan | 0 | 0 | C |
| **3** | 2008 | 1 | 7 | 2 | BB McCullum | RT Ponting | Z Khan | 1 | 0 | 1 |
| **4** | 2008 | 1 | 7 | 3 | RT Ponting | BB McCullum | Z Khan | 1 | 0 | 1 |

In [90]:
```
#Total runs scored in each season

season=season_data.groupby(['Season'])['total_runs'].sum().reset_index()
p=season.set_index('Season')
ax = plt.axes()
ax.set(facecolor = "White")
sns.lineplot(data=p,palette="magma")
plt.title('Total runs in each season',fontsize=12,fontweight="bold")
plt.show()
```

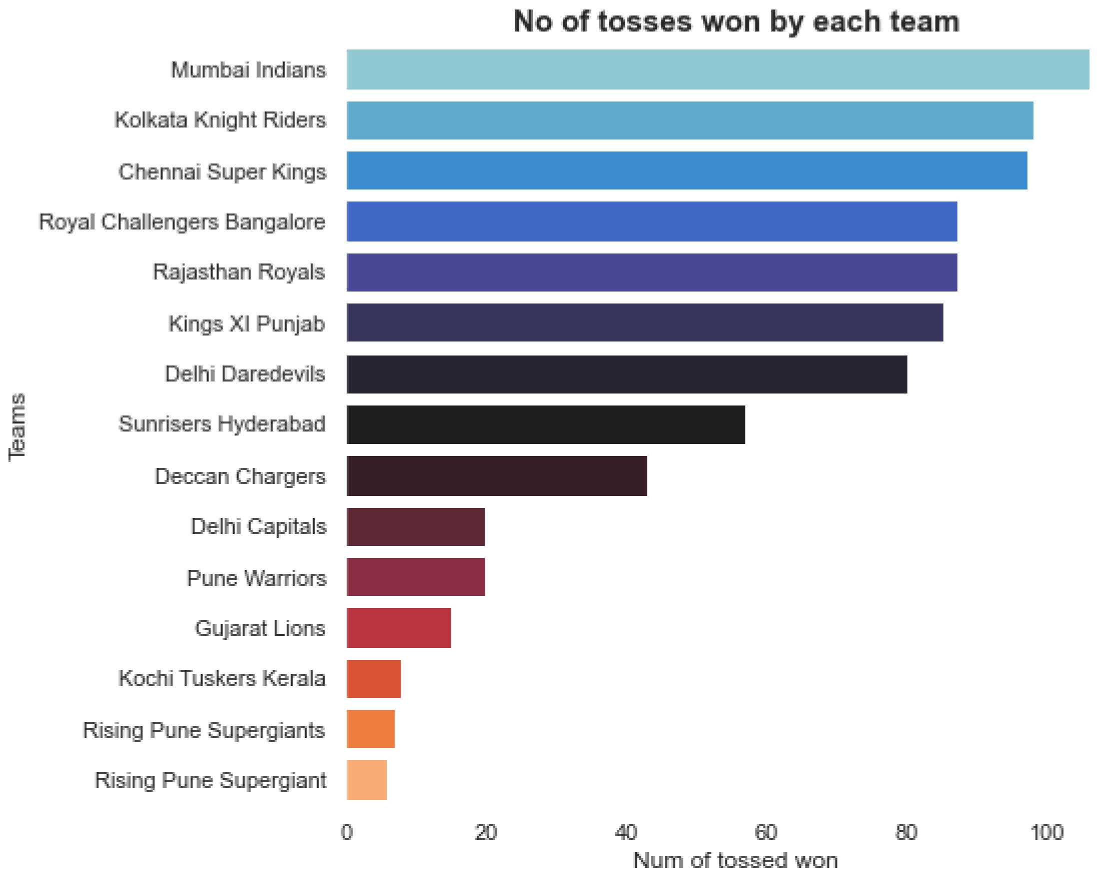


In [93]:
```
#runs scored per match

runs_per_season = pd.concat([match_per_season,season.iloc[:,1]], axis = 1)
runs_per_season['Runs scored per match'] = runs_per_season['total_runs']/runs_per
runs_per_season.set_index('Season', inplace = True)
runs_per_season
```

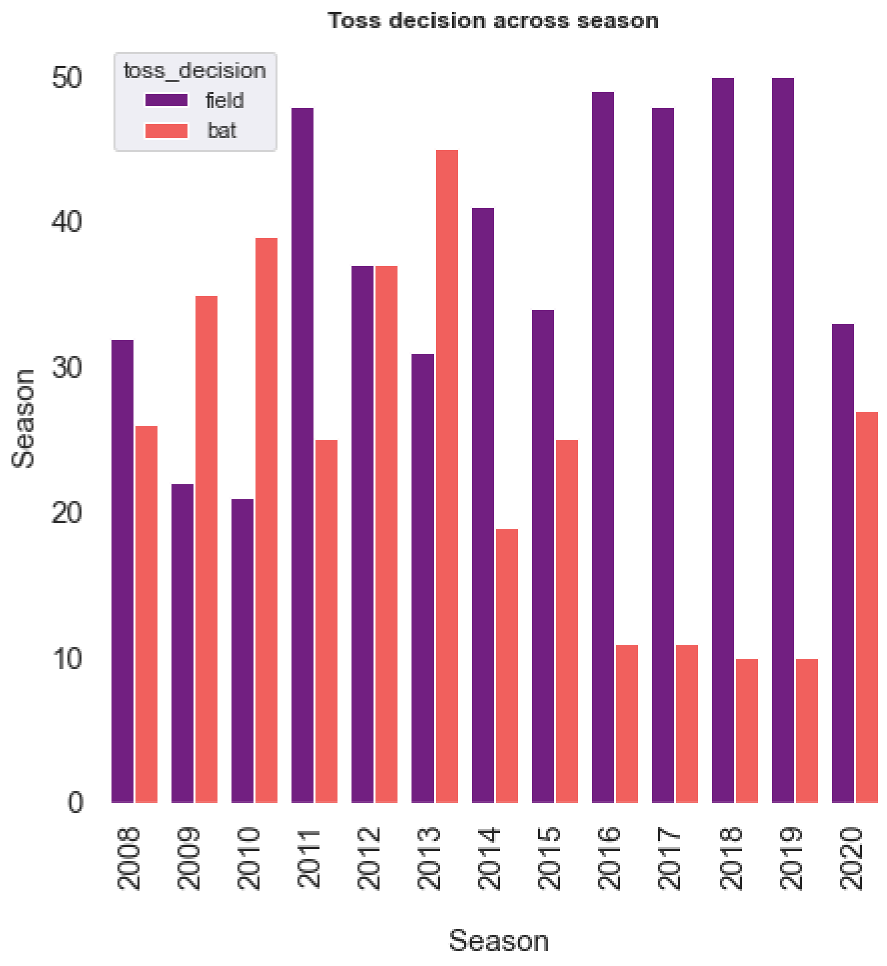
Out[93]:

| | matches | total_runs | Runs scored per match |
|---|---|---|---|
| **Season** | | | |
| **2008** | 58 | 17937 | 309.258621 |
| **2009** | 57 | 16320 | 286.315789 |
| **2010** | 60 | 18864 | 314.400000 |
| **2011** | 73 | 21154 | 289.780822 |
| **2012** | 74 | 22453 | 303.418919 |
| **2013** | 76 | 22541 | 296.592105 |
| **2014** | 60 | 18909 | 315.150000 |
| **2015** | 59 | 18332 | 310.711864 |
| **2016** | 60 | 18862 | 314.366667 |
| **2017** | 59 | 18769 | 318.118644 |
| **2018** | 60 | 19901 | 331.683333 |
| **2019** | 60 | 19400 | 323.333333 |
| **2020** | 60 | 19352 | 322.533333 |

In [122]:
```
#number of toss won by each team

toss = match_data['toss_winner'].value_counts()
ax = plt.axes()
ax.set(facecolor = 'white')
sns.set(rc = {'figure.figsize':(7,7)},style = 'darkgrid')
ax.set_title('No of tosses won by each team', fontsize = 15 , fontweight = "bold'
sns.barplot(x = toss , y = toss.index, orient = 'h', palette = "icefire", saturat
plt.xlabel('Num of tossed won')
plt.ylabel ('Teams')
plt.show()
```



In [141]:
```
#toss decision across seasons

ax = plt.axes()
ax.set(facecolor = "white")
sns.countplot(data = match_data, x = 'Season', hue = 'toss_decision', palette = '
sns.set(rc = {'figure.figsize':(7,7)},style = 'darkgrid')
plt.xticks(rotation = 90, fontsize = 15)
plt.yticks(fontsize = 15)
plt.xlabel('\n Season', fontsize = 15)
plt.ylabel('\n Season', fontsize = 15)
plt.title("Toss decision across season", fontsize = 12, fontweight = "bold")
plt.show()
```



In [145]:
```
match_data['result'].value_counts() #winner : chasing team or Bowling team
```

Out[145]:
```
wickets    435
runs       364
tie         13
Name: result, dtype: int64
```

In [146]:
```
match_data.venue[match_data.result != 'runs'].mode() #best venue to run chase
```
Out[146]:
```
0    Eden Gardens
dtype: object
```

In [147]:
```
match_data.venue[match_data.result != 'wickets'].mode() #best venue to bat first
```
Out[147]:
```
0    Feroz Shah Kotla
dtype: object
```

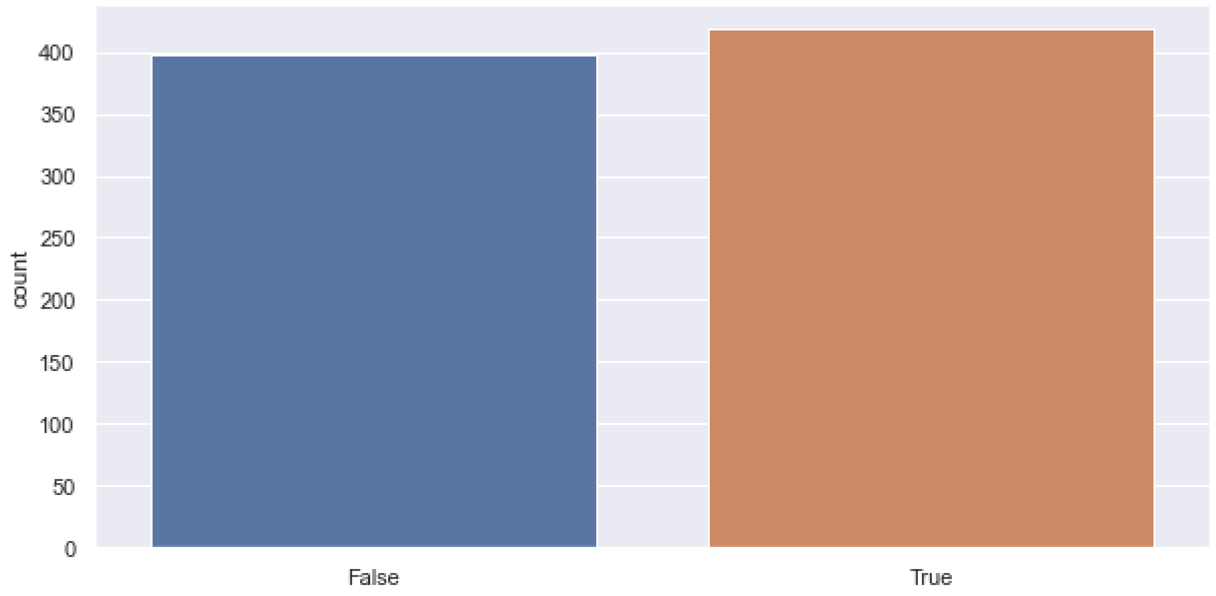
In [204]:
```
match_data.venue[match_data.toss_winner=='Kolkata Knight Riders'][match_data.winr
```
Out[204]:
```
0    Eden Gardens
dtype: object
```

In [205]:
```
match_data.winner[match_data.result != 'runs'].mode() #best run chasing teams
```
Out[205]:
```
0    Kolkata Knight Riders
1           Mumbai Indians
dtype: object
```

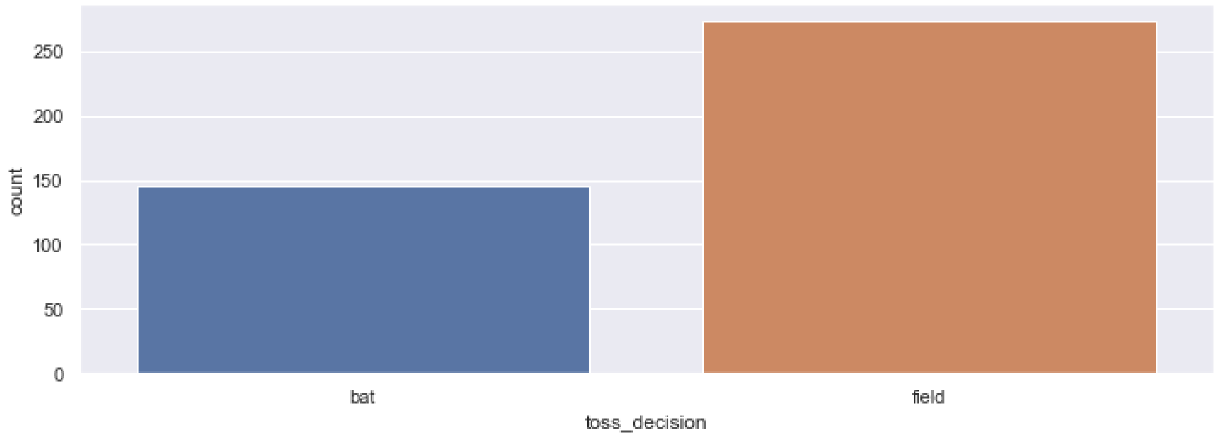
In [206]:
```
match_data.winner[match_data.result != 'wicket'].mode() #best defending team
```
Out[206]:
```
0    Mumbai Indians
dtype: object
```

In [208]:
```
#winning the toss means winning the match ?

toss = match_data['toss_winner'] == match_data['winner']
plt.figure(figsize= (10,5))
sns.countplot(toss)
plt.show

#output no clear evidence that winning the toss will result in winning the match
```

Out[208]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [210]:
```
#Choosing batting or bowling will result in win ?

plt.figure(figsize = (12,4))
sns.countplot(match_data.toss_decision[match_data.toss_winner == match_data.winne
plt.show()

#Output - maximum number of time team deciding to field first turned out to be wi
#i.e.,there are higher chances of winning if you field first
```

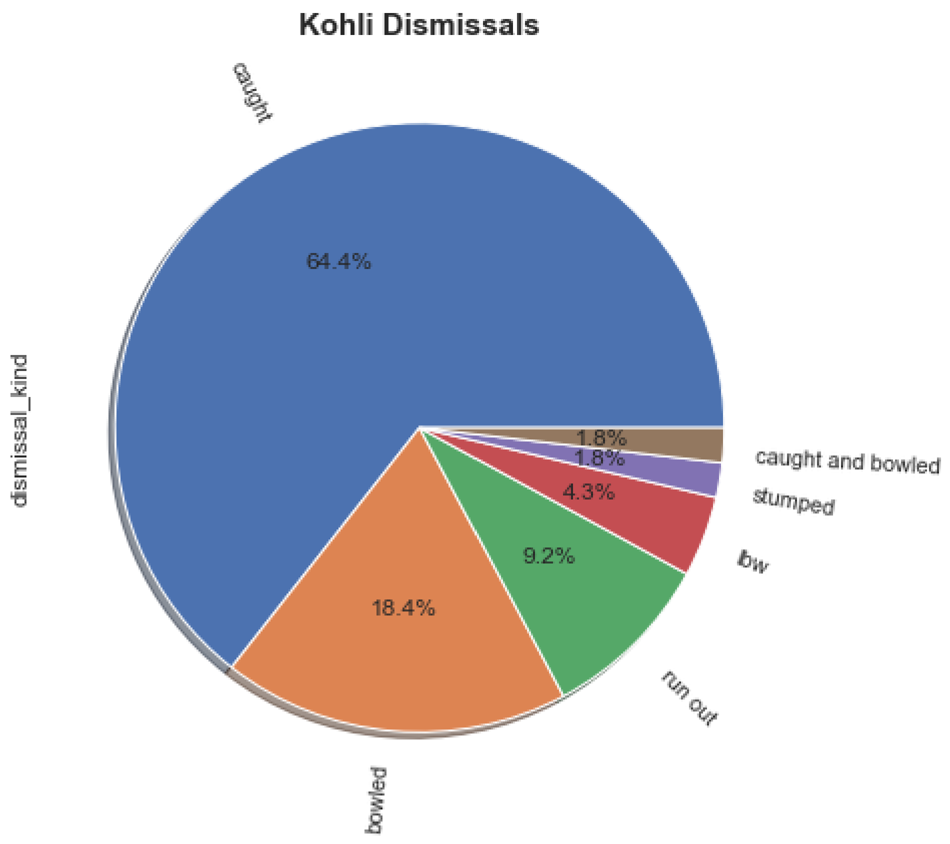


In [215]:
```
#player analysis : Virat Kohli

player = (ball_data['batsman'] == 'V Kohli')
df_kohli = ball_data[player]
df_kohli.head()
```

Out[215]:

| | id | inning | over | ball | batsman | non_striker | bowler | batsman_runs | extra_runs | total_ru |
|---|---|---|---|---|---|---|---|---|---|---|
| **211** | 335982 | 2 | 1 | 2 | V Kohli | W Jaffer | I Sharma | 0 | 0 | |
| **212** | 335982 | 2 | 1 | 3 | V Kohli | W Jaffer | I Sharma | 0 | 4 | |
| **213** | 335982 | 2 | 1 | 4 | V Kohli | W Jaffer | I Sharma | 1 | 0 | |
| **216** | 335982 | 2 | 2 | 1 | V Kohli | W Jaffer | AB Dinda | 0 | 0 | |
| **217** | 335982 | 2 | 2 | 2 | V Kohli | W Jaffer | AB Dinda | 0 | 0 | |

In [220]:
```
df_kohli['dismissal_kind'].value_counts().plot.pie(autopct= '%1.1f%%', shadow = T
plt.title("Kohli Dismissals", fontweight = "bold", fontsize=15)
plt.show()
```



In [221]:
```
def count(df_kohli, runs):
    return len(df_kohli[df_kohli['batsman_runs'] == runs])*runs
```

In [225]:
```
print("Runs scored from 1's:",count(df_kohli,1))
print("Runs scored from 2's:",count(df_kohli,2))
print("Runs scored from 3's:",count(df_kohli,3))
print("Runs scored from 4's:",count(df_kohli,4))
print("Runs scored from 6's:",count(df_kohli,6))
```

```
Runs scored from 1's: 1919
Runs scored from 2's: 692
Runs scored from 3's: 39
Runs scored from 4's: 2016
Runs scored from 6's: 1212
```

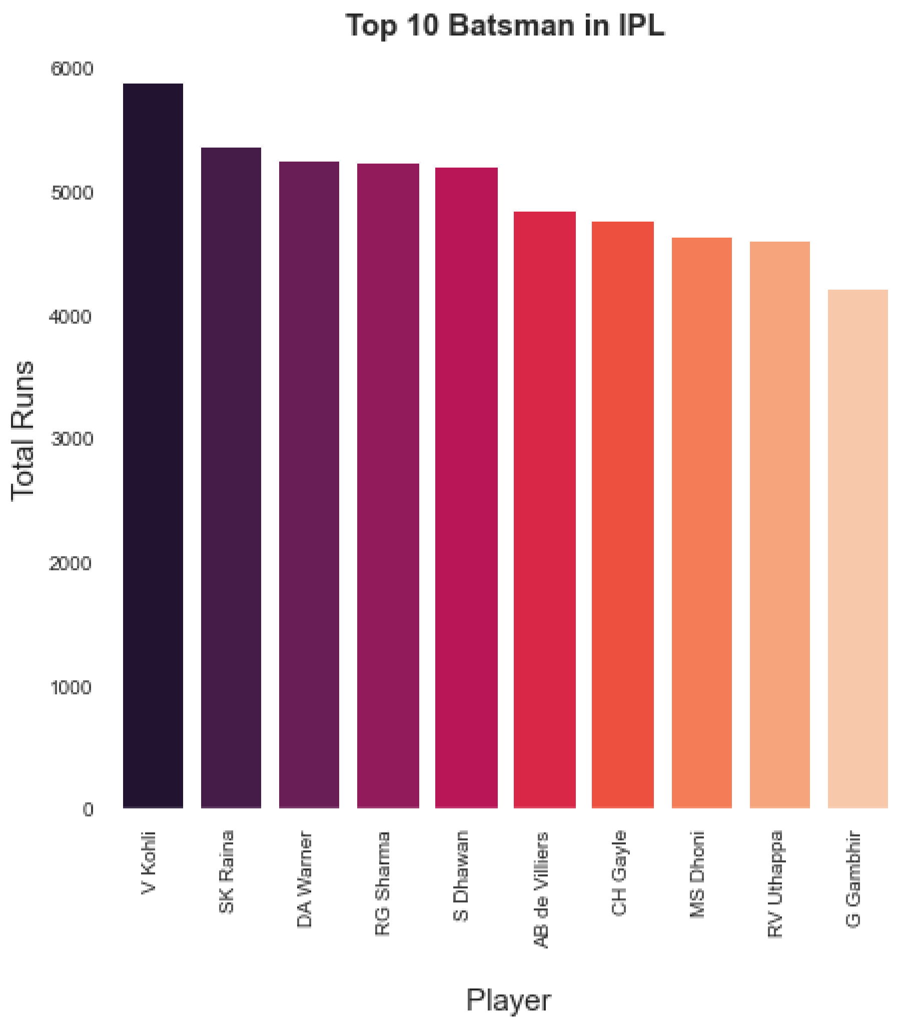
In [232]:
```
#maximum number of runs scored by player (top 10 scorer)

runs = ball_data.groupby(['batsman'])['batsman_runs'].sum().reset_index()
runs.columns = ['Batsman', 'runs']
y = runs.sort_values(by='runs',ascending = False).head(10).reset_index().drop('ir
y
```

Out[232]:

| | Batsman | runs |
|---|---|---|
| **0** | V Kohli | 5878 |
| **1** | SK Raina | 5368 |
| **2** | DA Warner | 5254 |
| **3** | RG Sharma | 5230 |
| **4** | S Dhawan | 5197 |
| **5** | AB de Villiers | 4849 |
| **6** | CH Gayle | 4772 |
| **7** | MS Dhoni | 4632 |
| **8** | RV Uthappa | 4607 |
| **9** | G Gambhir | 4217 |

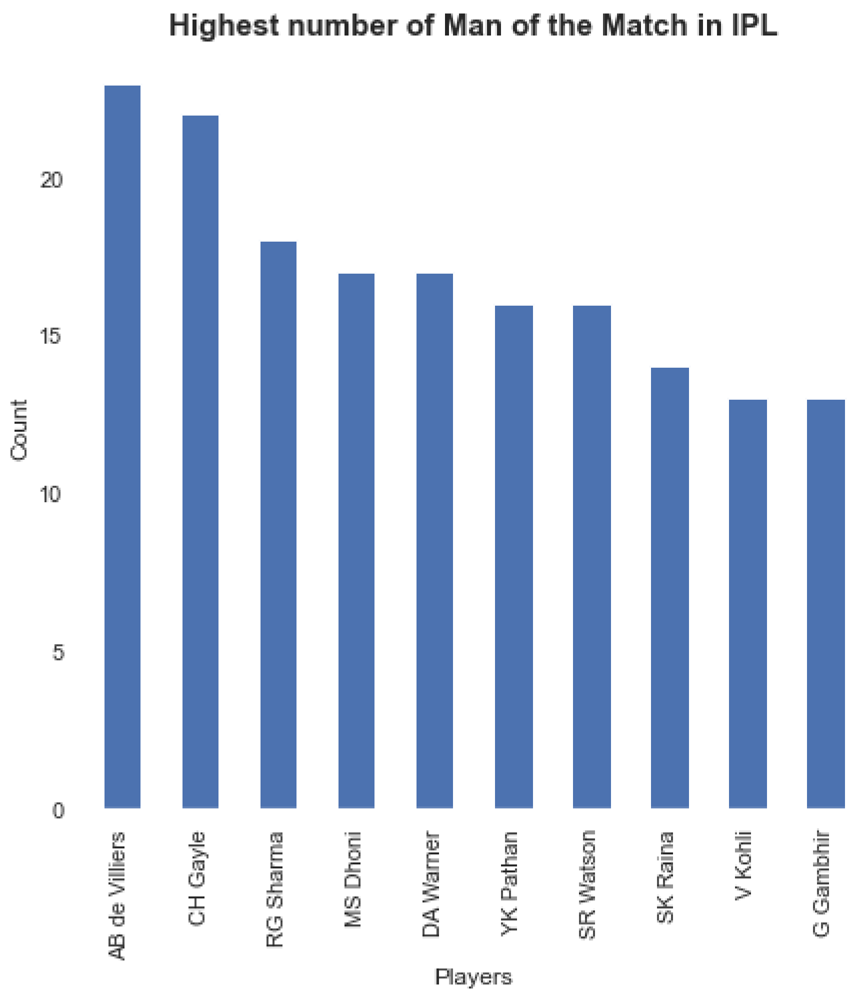
In [237]:
```
#plotting

ax=plt.axes()
ax.set(facecolor = "white")
sns.barplot(x=y['Batsman'], y = y['runs'], palette = "rocket", saturation = 1)
plt.xticks (rotation = 90,fontsize = 10)
plt.yticks (fontsize = 10)
plt.xlabel('\n Player', fontsize = 15)
plt.ylabel('Total Runs', fontsize = 15)
plt.title('Top 10 Batsman in IPL', fontsize = 15,fontweight = "bold")
```

Out[237]:
```
Text(0.5, 1.0, 'Top 10 Batsman in IPL')
```



In [242]:
```
#man of the match maximum times

ax = plt.axes()
ax.set(facecolor = "white")
match_data.player_of_match.value_counts()[:10].plot(kind = 'bar')
plt.xlabel('Players')
plt.ylabel("Count")
plt.title("Highest number of Man of the Match in IPL", fontsize = 15, fontweight
```

Out[242]:
```
Text(0.5, 1.0, 'Highest number of Man of the Match in IPL')
```



```
In [ ]:
```