EDA on Cricket Dataset from IPL

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

games = pd.read_csv("C:/Users/chroh/Downloads/Indian Premier
League/matches.csv")
score = pd.read_csv("C:/Users/chroh/Downloads/Indian Premier
League/deliveries.csv")

games.head()
```

```
   id  season       city        date                          team1  \
0   1    2017  Hyderabad  2017-04-05          Sunrisers Hyderabad
1   2    2017       Pune  2017-04-06               Mumbai Indians
2   3    2017     Rajkot  2017-04-07                Gujarat Lions
3   4    2017     Indore  2017-04-08       Rising Pune Supergiant
4   5    2017  Bangalore  2017-04-08  Royal Challengers Bangalore

                          team2                  toss_winner
toss_decision  \
0  Royal Challengers Bangalore  Royal Challengers Bangalore
field
1       Rising Pune Supergiant       Rising Pune Supergiant
field
2        Kolkata Knight Riders        Kolkata Knight Riders
field
3              Kings XI Punjab              Kings XI Punjab
field
4             Delhi Daredevils  Royal Challengers Bangalore
bat

   result  dl_applied                       winner  win_by_runs  \
0  normal           0          Sunrisers Hyderabad           35
1  normal           0       Rising Pune Supergiant            0
2  normal           0        Kolkata Knight Riders            0
3  normal           0              Kings XI Punjab            0
4  normal           0  Royal Challengers Bangalore           15

   win_by_wickets player_of_match
venue  \
0               0    Yuvraj Singh  Rajiv Gandhi International Stadium,
Uppal
1               7       SPD Smith    Maharashtra Cricket Association
Stadium
2              10         CA Lynn     Saurashtra Cricket Association
```

```
Stadium
3               6      GJ Maxwell                       Holkar Cricket
Stadium
4               0       KM Jadhav                        M Chinnaswamy
Stadium

          umpire1        umpire2 umpire3
0     AY Dandekar       NJ Llong     NaN
1  A Nand Kishore         S Ravi     NaN
2     Nitin Menon      CK Nandan     NaN
3    AK Chaudhary  C Shamshuddin     NaN
4             NaN            NaN     NaN

score.head()

   match_id  inning         batting_team                 bowling_team
over  \
0         1       1  Sunrisers Hyderabad  Royal Challengers Bangalore
1
1         1       1  Sunrisers Hyderabad  Royal Challengers Bangalore
1
2         1       1  Sunrisers Hyderabad  Royal Challengers Bangalore
1
3         1       1  Sunrisers Hyderabad  Royal Challengers Bangalore
1
4         1       1  Sunrisers Hyderabad  Royal Challengers Bangalore
1

   ball    batsman non_striker    bowler  is_super_over  ...  bye_runs
\
0     1  DA Warner    S Dhawan  TS Mills              0  ...         0

1     2  DA Warner    S Dhawan  TS Mills              0  ...         0

2     3  DA Warner    S Dhawan  TS Mills              0  ...         0

3     4  DA Warner    S Dhawan  TS Mills              0  ...         0

4     5  DA Warner    S Dhawan  TS Mills              0  ...         0


   legbye_runs  noball_runs  penalty_runs  batsman_runs  extra_runs  \
0            0            0             0             0           0
1            0            0             0             0           0
2            0            0             0             4           0
3            0            0             0             0           0
4            0            0             0             0           2

   total_runs  player_dismissed dismissal_kind fielder
0           0               NaN            NaN     NaN
1           0               NaN            NaN     NaN
```

```
2           4                NaN             NaN     NaN
3           0                NaN             NaN     NaN
4           2                NaN             NaN     NaN

[5 rows x 21 columns]
```

```
print(games.info())
print(score.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
None
<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
None
```

```
games["umpire3"].isnull().sum()
```

```
637
```

```
games['season'].unique()
```

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

```
games.describe()
```

```
                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
```

```
#winner of every edition
temp_df = games.drop_duplicates(subset=['season'], keep='last')
[['season', 'winner']].reset_index(drop=True)
temp_df
```

```
    season                  winner
0     2017          Mumbai Indians
1     2008        Rajasthan Royals
2     2009         Deccan Chargers
3     2010    Chennai Super Kings
4     2011    Chennai Super Kings
5     2012  Kolkata Knight Riders
6     2013          Mumbai Indians
7     2014  Kolkata Knight Riders
8     2015          Mumbai Indians
9     2016    Sunrisers Hyderabad
10    2018    Chennai Super Kings
11    2019          Mumbai Indians
```

```
#team with max runs
games.iloc[games['win_by_runs'].idxmax()]
```

```
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
umpire3                         NaN
Name: 43, dtype: object
```

```
games.iloc[games['win_by_runs'].idxmax()]['winner']
```

```
'Mumbai Indians'
```

```
#team with max wickets
games.iloc[games['win_by_wickets'].idxmax()]
```

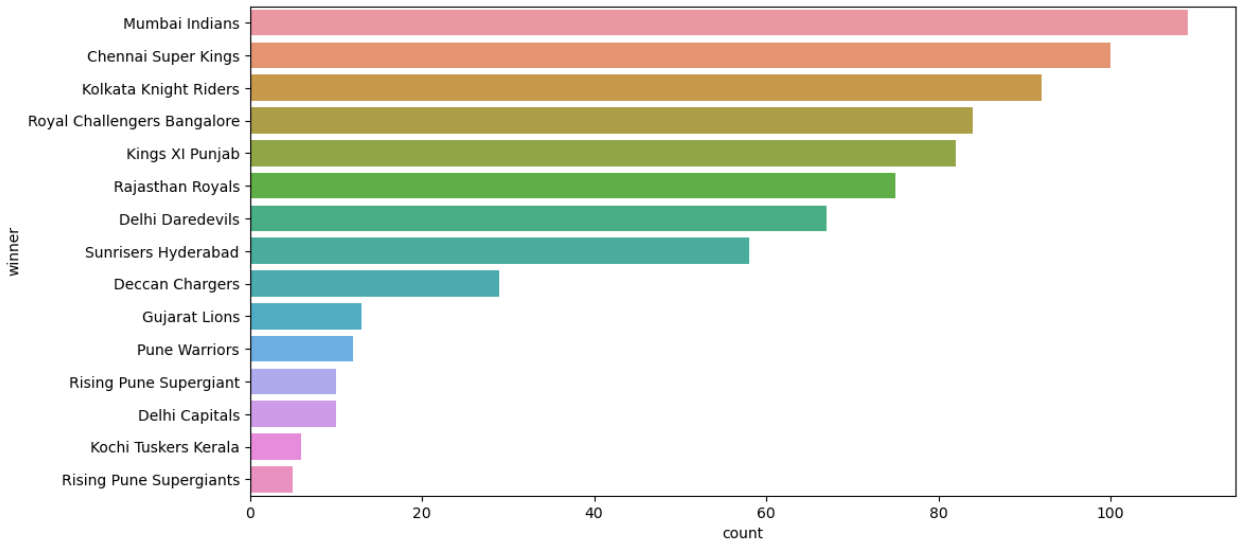
```
id                                                          3
season                                                   2017
city                                                   Rajkot
date                                                2017-04-07
team1                                          Gujarat 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
umpire3                                                      NaN
Name: 2, dtype: object
```

```
games.iloc[games['win_by_wickets'].idxmax()]['winner']
```

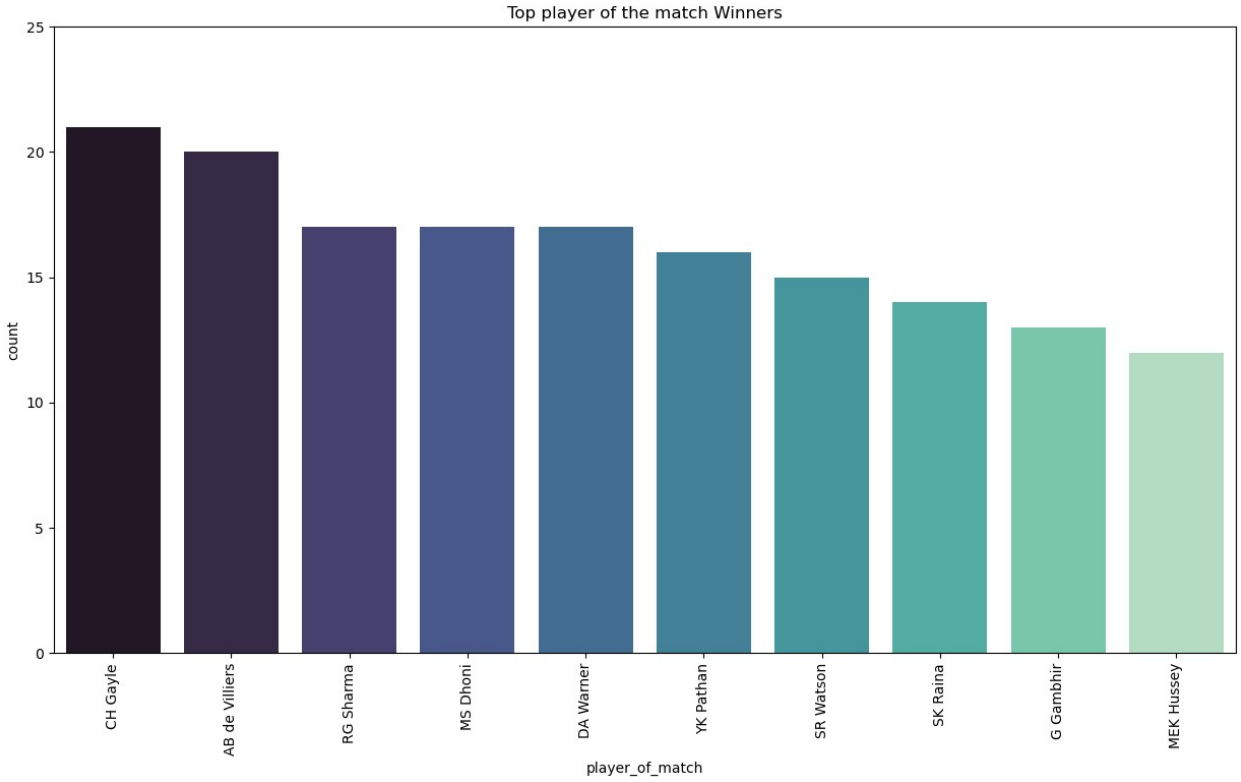
```
'Kolkata Knight Riders'
```

```
#games won by teams
plt.figure(figsize=(12,6))
data = games.winner.value_counts()
sns.barplot(y = data.index, x = data, orient='h',)
plt.show()
```

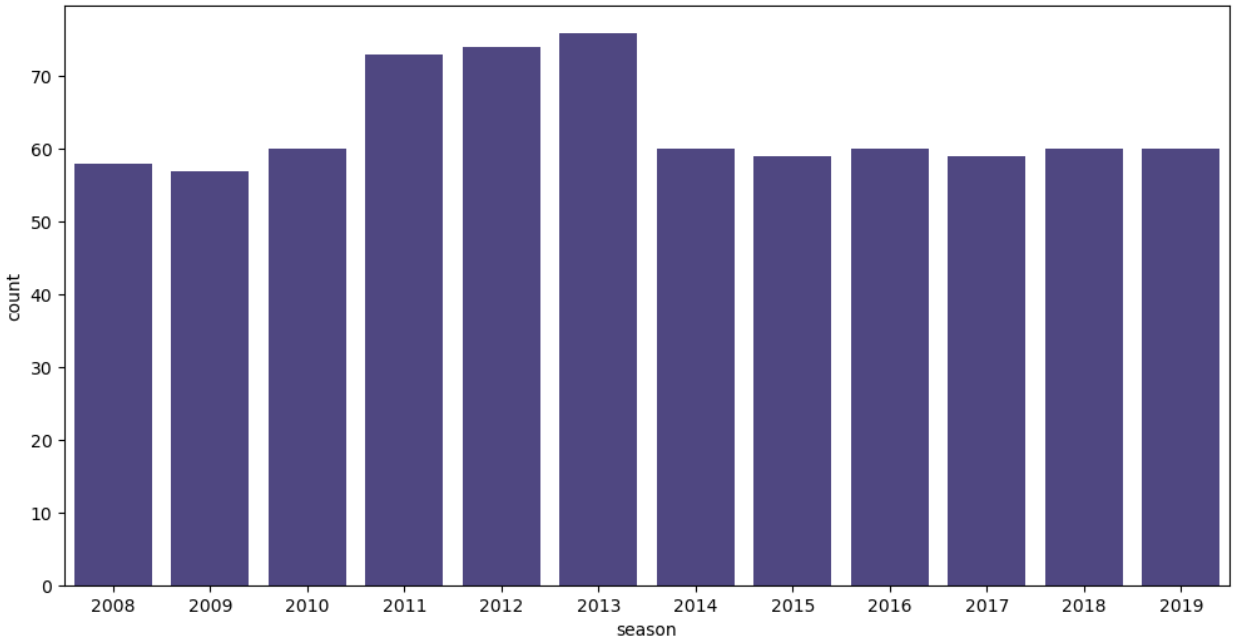



```
#succesful players
top_players = games.player_of_match.value_counts()[:10]
fig, ax = plt.subplots(figsize=(15,8))
ax.set_ylim([0,25])
ax.set_ylabel("Count")
ax.set_title("Top player of the match Winners")
top_players.plot.bar()
sns.barplot(x = top_players.index, y = top_players, orient='v',
```

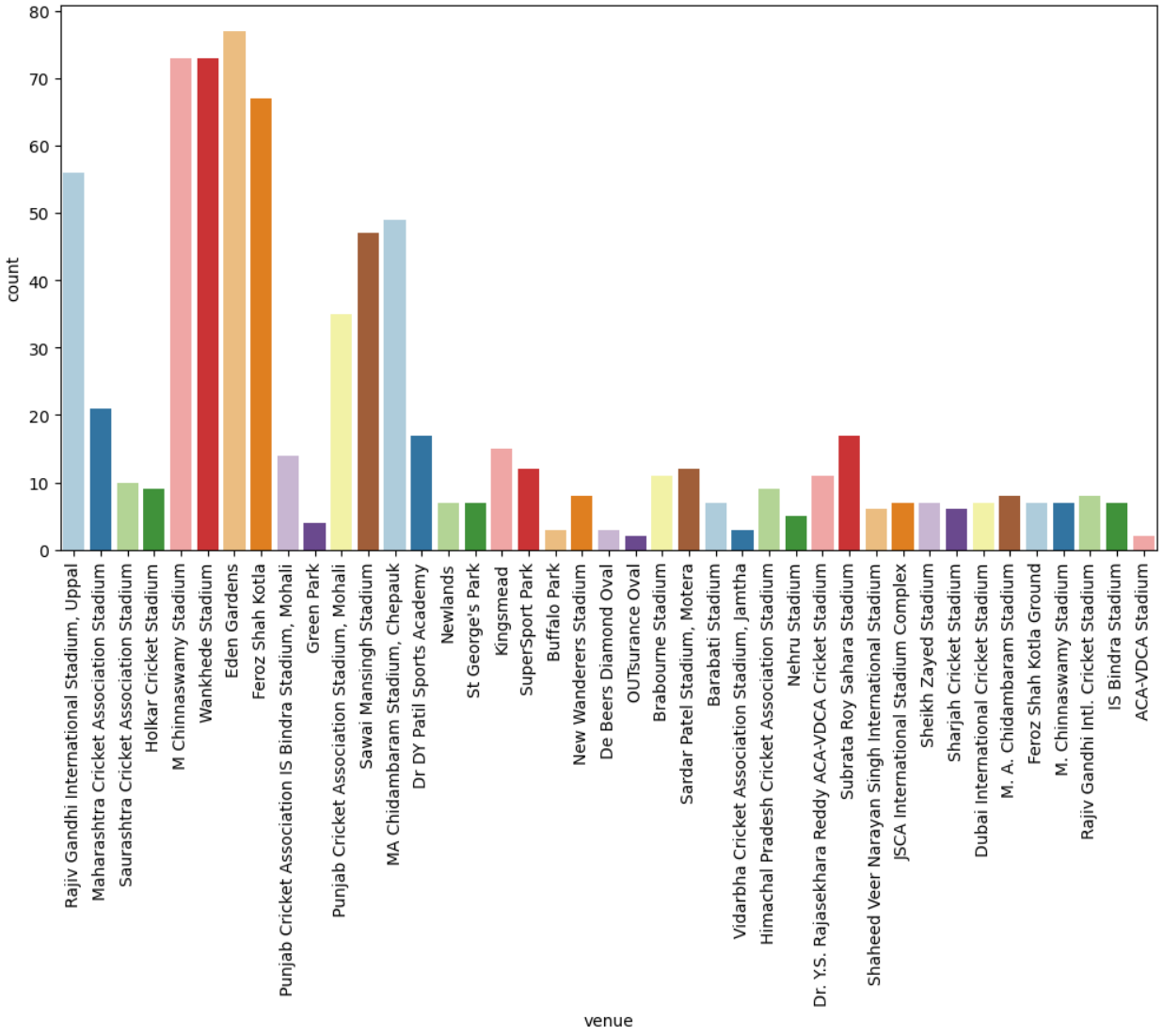
```
palette="mako");
plt.show()
```




```
#seasons & no of games
plt.figure(figsize=(12,6))
sns.countplot(x='season', data=games,color='darkslateblue')
plt.show()
```

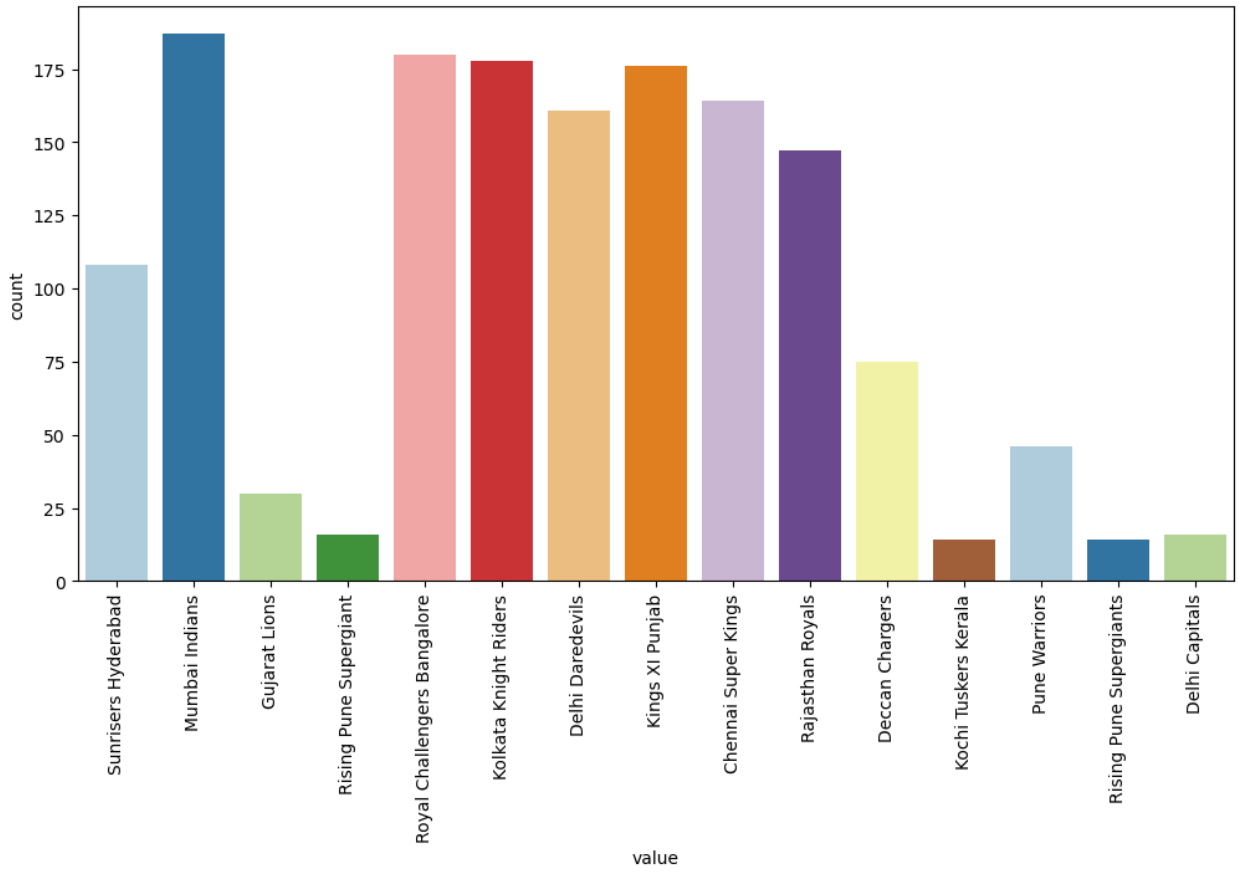



```
#games played in venues
plt.figure(figsize=(12,6))
sns.countplot(x='venue', data=games,palette='Paired')
plt.xticks(rotation='vertical')
plt.show()
```

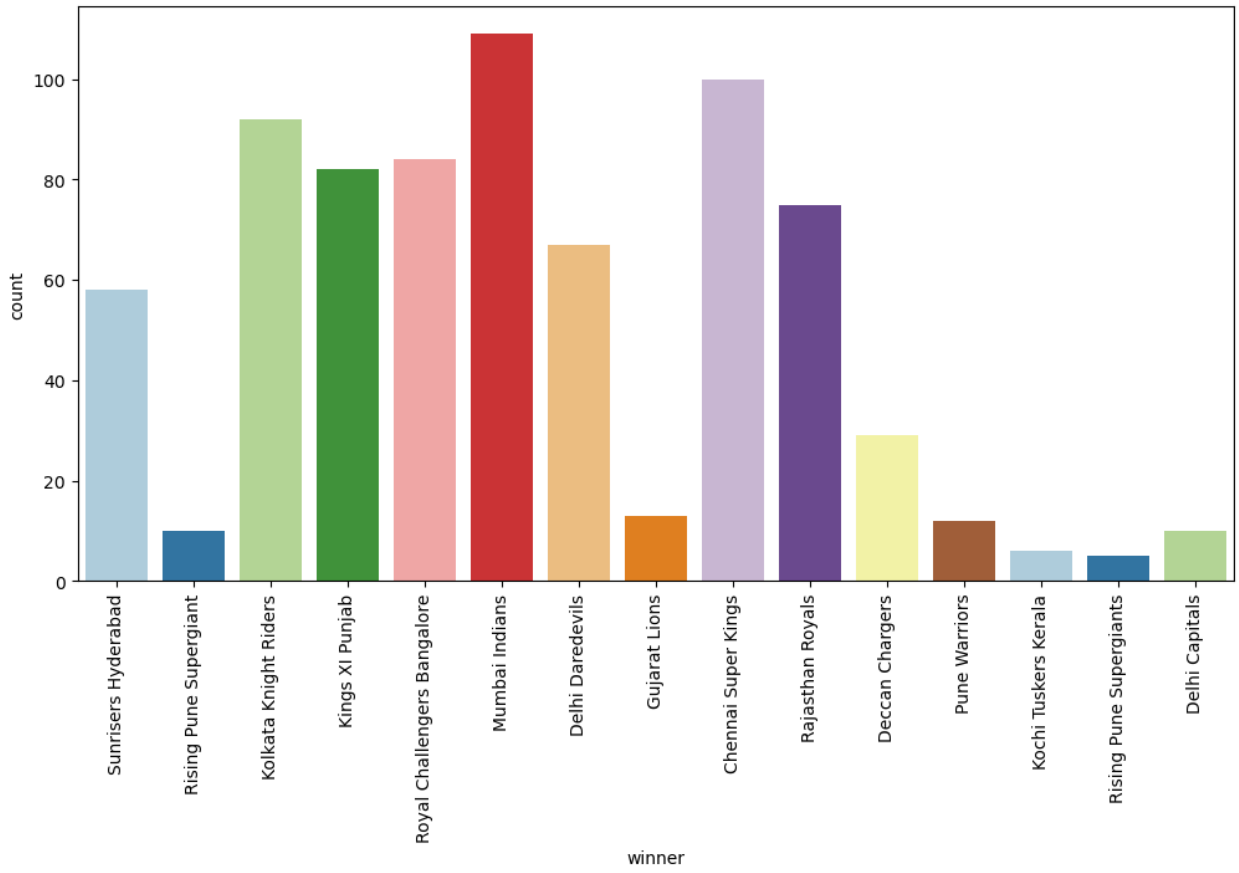



```
#games played by each team
temp_df = pd.melt(games, id_vars=['id','season'], value_vars=['team1',
'team2'])

plt.figure(figsize=(12,6))
sns.countplot(x='value', data=temp_df,palette='Paired')
plt.xticks(rotation='vertical')
plt.show()
```

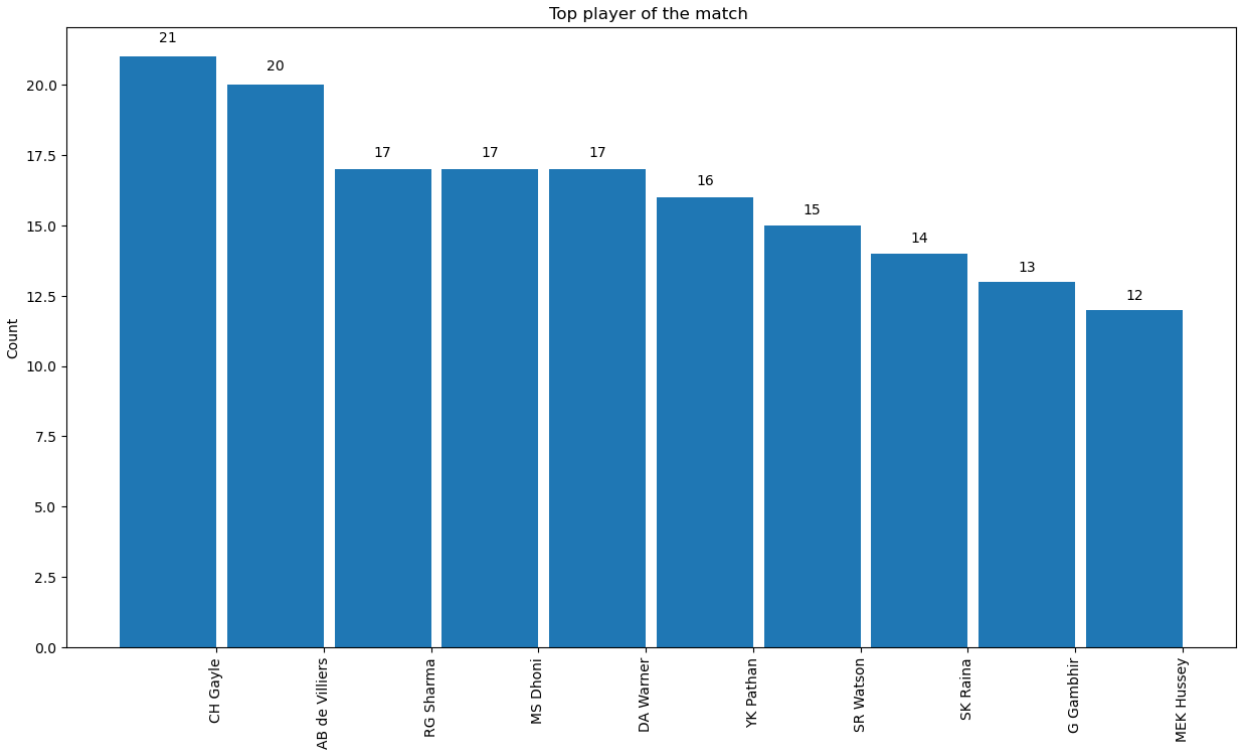



```
#Victory per team
plt.figure(figsize=(12,6))
sns.countplot(x='winner', data=games,palette='Paired')
plt.xticks(rotation=90)
plt.show()
```



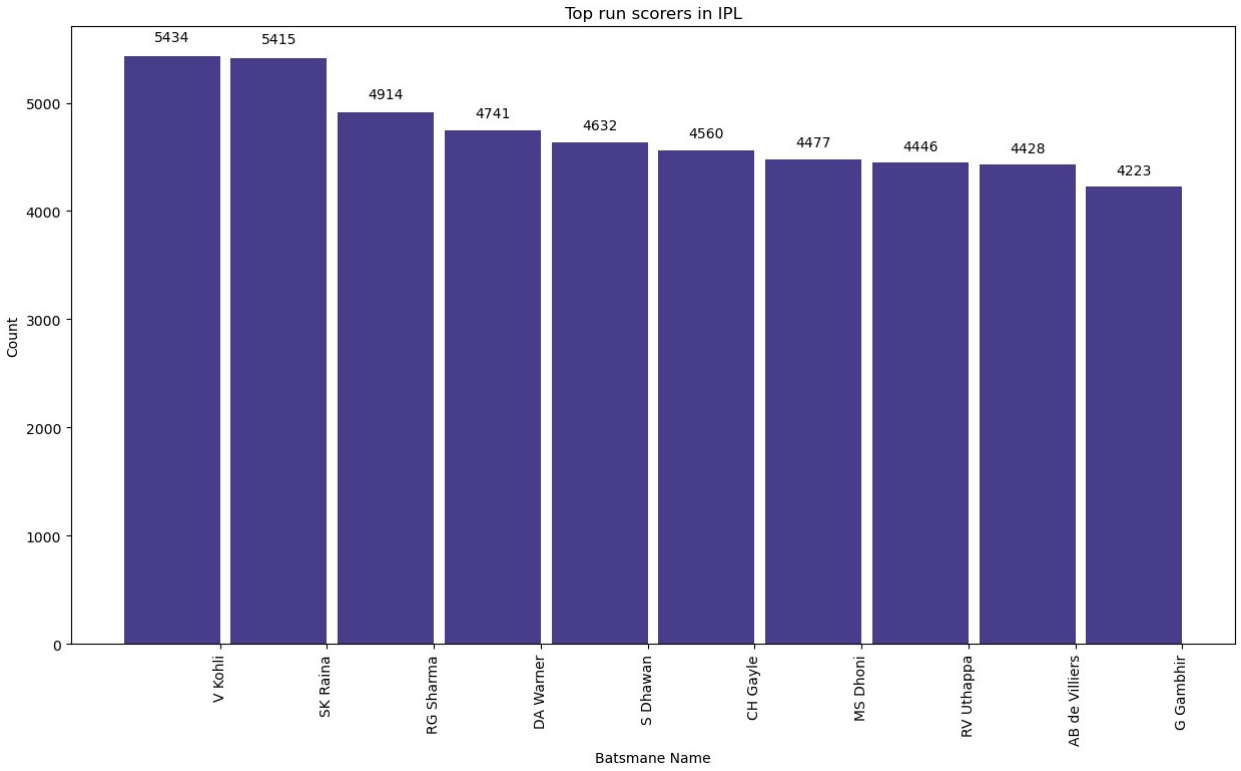

```
# TOP PLAYERS
def autolabel(rects):
    for rect in rects:
        height = rect.get_height()
        ax.text(rect.get_x() + rect.get_width()/2., 1.02*height,
                '%d' % int(height),
                ha='center', va='bottom')

temp_series = games.player_of_match.value_counts()[:10]
labels = np.array(temp_series.index)
ind = np.arange(len(labels))
width = 0.9
fig, ax = plt.subplots(figsize=(15,8))
rects = ax.bar(ind, np.array(temp_series), width=width)
ax.set_xticks(ind+((width)/2.))
ax.set_xticklabels(labels, rotation='vertical')
ax.set_ylabel("Count")
ax.set_title("Top player of the match ")
autolabel(rects)
plt.show()
```



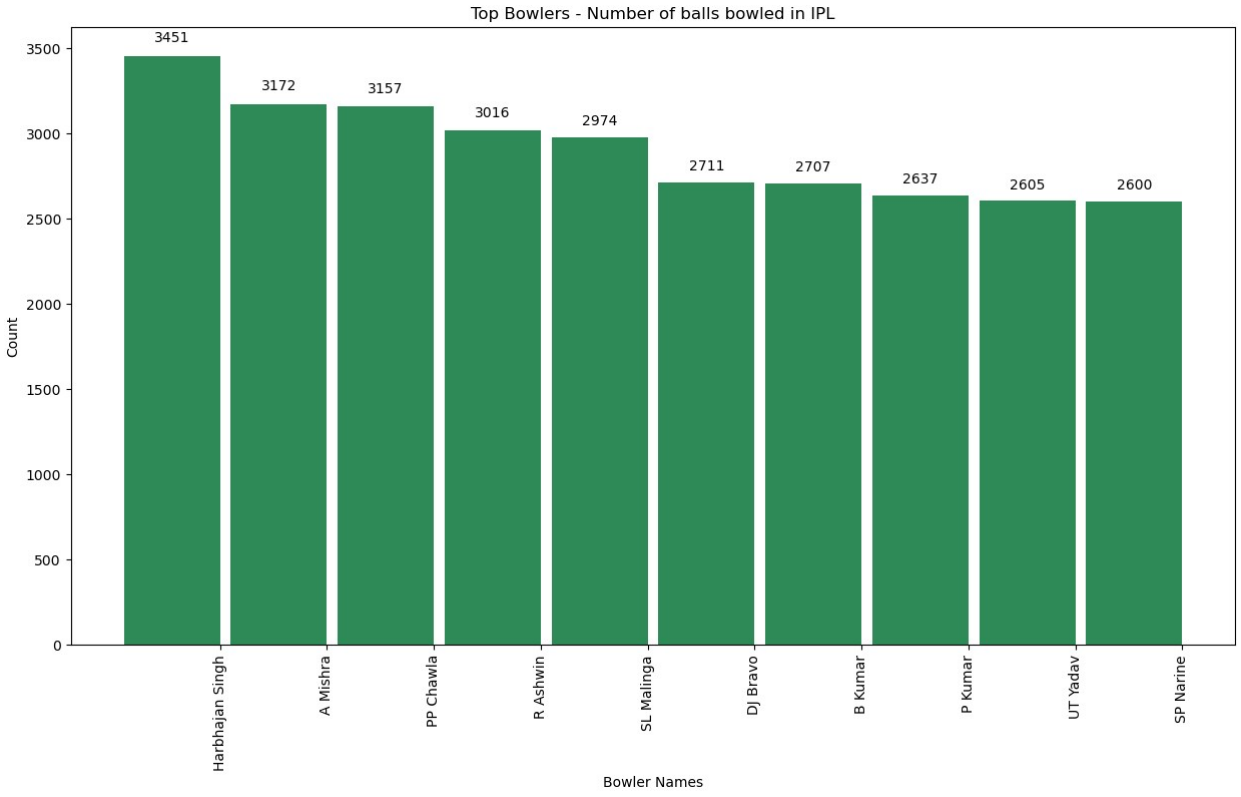

```
#Batsman & Runs scored
temp_df = score.groupby('batsman')
['batsman_runs'].agg('sum').reset_index().sort_values(by='batsman_runs
', ascending=False).reset_index(drop=True)
temp_df = temp_df.iloc[:10,:]

labels = np.array(temp_df['batsman'])
ind = np.arange(len(labels))
width = 0.9
fig, ax = plt.subplots(figsize=(15,8))
rects = ax.bar(ind, np.array(temp_df['batsman_runs']), width=width,
color='darkslateblue')
ax.set_xticks(ind+((width)/2.))
ax.set_xticklabels(labels, rotation='vertical')
ax.set_ylabel("Count")
ax.set_title("Top run scorers in IPL")
ax.set_xlabel('Batsmane Name')
autolabel(rects)
plt.show()
```

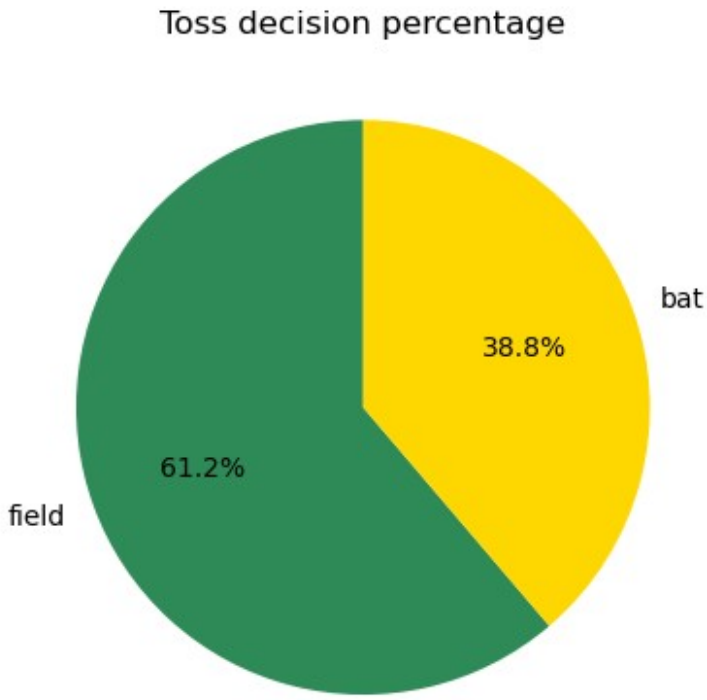



```
#Bowlers & bowls balled
temp_df = score.groupby('bowler')
['ball'].agg('count').reset_index().sort_values(by='ball',
ascending=False).reset_index(drop=True)
temp_df = temp_df.iloc[:10,:]

labels = np.array(temp_df['bowler'])
ind = np.arange(len(labels))
width = 0.9
fig, ax = plt.subplots(figsize=(15,8))
rects = ax.bar(ind, np.array(temp_df['ball']), width=width,
color='Seagreen')
ax.set_xticks(ind+((width)/2.))
ax.set_xticklabels(labels, rotation='vertical')
ax.set_ylabel("Count")
ax.set_title("Top Bowlers - Number of balls bowled in IPL")
ax.set_xlabel('Bowler Names')
autolabel(rects)
plt.show()
```




```
# Toss data
temp_series = games.toss_decision.value_counts()
labels = (np.array(temp_series.index))
sizes = (np.array((temp_series / temp_series.sum())*100))
colors = ['Seagreen', 'gold']
plt.pie(sizes, labels=labels, colors=colors,
        autopct='%1.1f%%',  startangle=90)
plt.title("Toss decision percentage")
plt.show()
```


Toss decision percentage
bat
38.8%
61.2%
field