TASK-5

Author: ISHWARYA D

THE SPARKS FOUNDATION INTERNSHIP

Exploratory Data Analysis - Sports

● Perform 'Exploratory Data Analysis' on dataset 'Indian Premier League' ● As a sports analysts, find out the most successful teams, players and factors contributing win or loss of a team.

EXPLORING THE DATA

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
data = pd.read_csv('matches.csv')
data.head(10)
```

Out[3]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0	Royal Challengers Bangalore	
5	6	2017	Hyderabad	2017-04-09	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal	0	Sunrisers Hyderabad	
6	7	2017	Mumbai	2017-04-09	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	normal	0	Mumbai Indians	
7	8	2017	Indore	2017-04-10	Royal Challengers Bangalore	Kings XI Punjab	Royal Challengers Bangalore	bat	normal	0	Kings XI Punjab	
8	9	2017	Pune	2017-04-11	Delhi Daredevils	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Delhi Daredevils	
9	10	2017	Mumbai	2017-04-12	Sunrisers Hyderabad	Mumbai Indians	Mumbai Indians	field	normal	0	Mumbai Indians	

In [4]:
```
data.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 [5]:
```
data.describe()
```

Out[5]:

	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 [6]:
```
data['id'].max()
```

Out[6]:
```
11415
```

In [7]:
```
data['season'].unique()
```

Out[7]:
```
array([2017, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2018,
       2019], dtype=int64)
```

In [8]:
```
len(data['season'].unique())
```

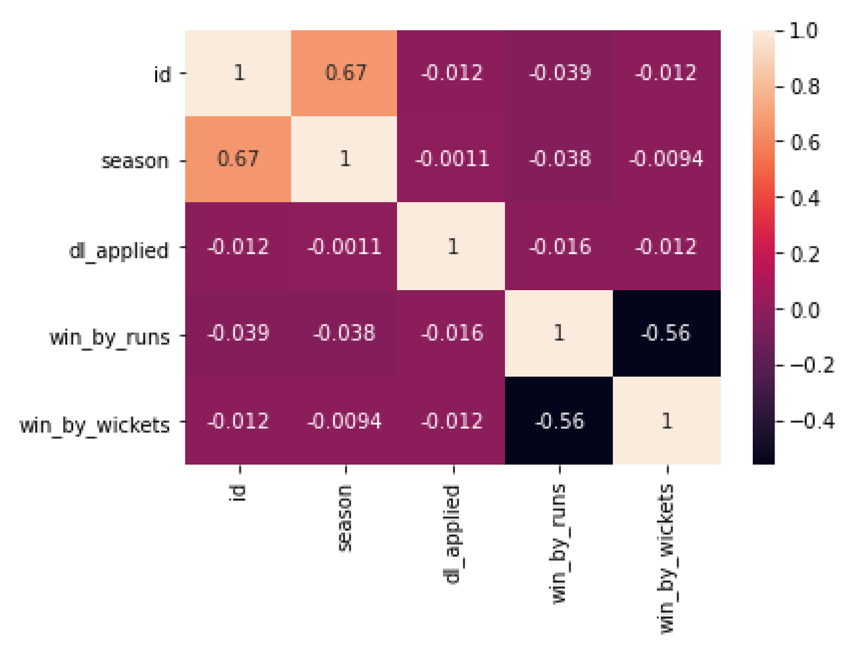
Out[8]:
```
12
```

In [9]:
```
data.iloc[data['win_by_runs'].idxmax()]
```

Out[9]:
```
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
```

In [10]:
```
sns.heatmap(data.corr(),annot=True)
```

Out[10]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x22f7ddddfd0>
```



In [11]:
```
data['player_of_match']
```

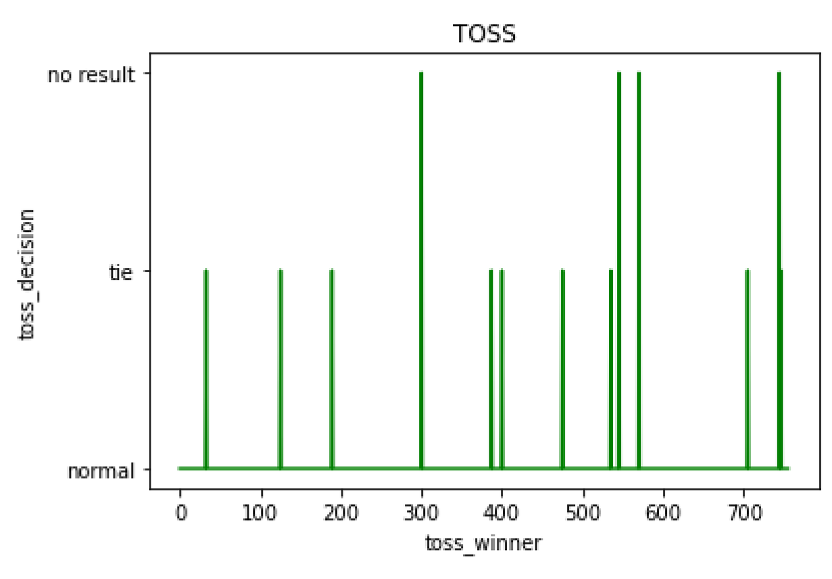
Out[11]:
```
0      Yuvraj Singh
1         SPD Smith
2           CA Lynn
3        GJ Maxwell
4         KM Jadhav
          ...
751       HH Pandya
752        AS Yadav
753         RR Pant
754    F du Plessis
755       JJ Bumrah
Name: player_of_match, Length: 756, dtype: object
```

In [12]:
```
data['player_of_match'].value_counts
```

Out[12]:
```
<bound method IndexOpsMixin.value_counts of 0      Yuvraj Singh
1         SPD Smith
2           CA Lynn
3        GJ Maxwell
4         KM Jadhav
          ...
751       HH Pandya
752        AS Yadav
753         RR Pant
754    F du Plessis
755       JJ Bumrah
Name: player_of_match, Length: 756, dtype: object>
```

PLOTTING GRAPH FOR TOSS PREDICTION

In [13]:
```
plt.plot(data['result'],color='green')
plt.xlabel('toss_winner')
plt.ylabel('toss_decision')
plt.title("TOSS")
plt.show()
```

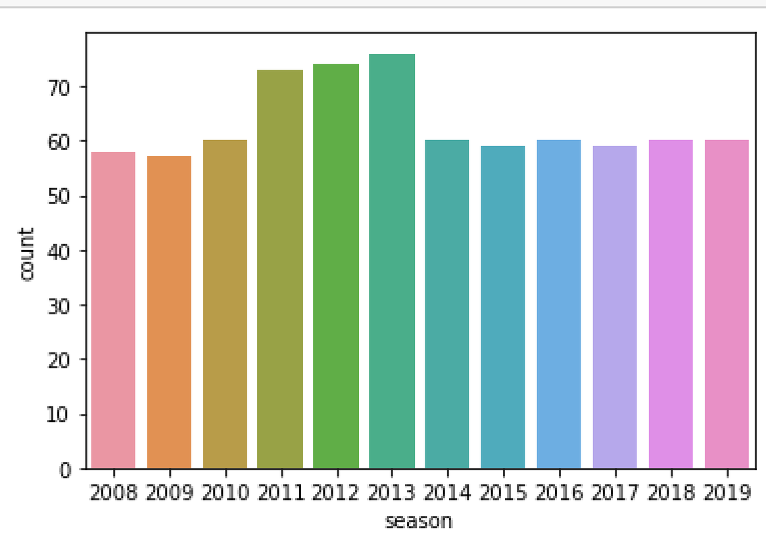


In [14]:
```
data.iloc[data[data['win_by_runs'].ge(1)].win_by_runs.idxmin()]['winner']
```

Out[14]:
```
'Mumbai Indians'
```

SEASONS

In [15]:
```
sns.countplot(x='season', data=data)
plt.show()
```



PIVOT TABLE

In [16]:
```
grades_mean=data.pivot_table(values='win_by_wickets',columns='win_by_runs',aggfunc=np.mean)
grades_mean
```

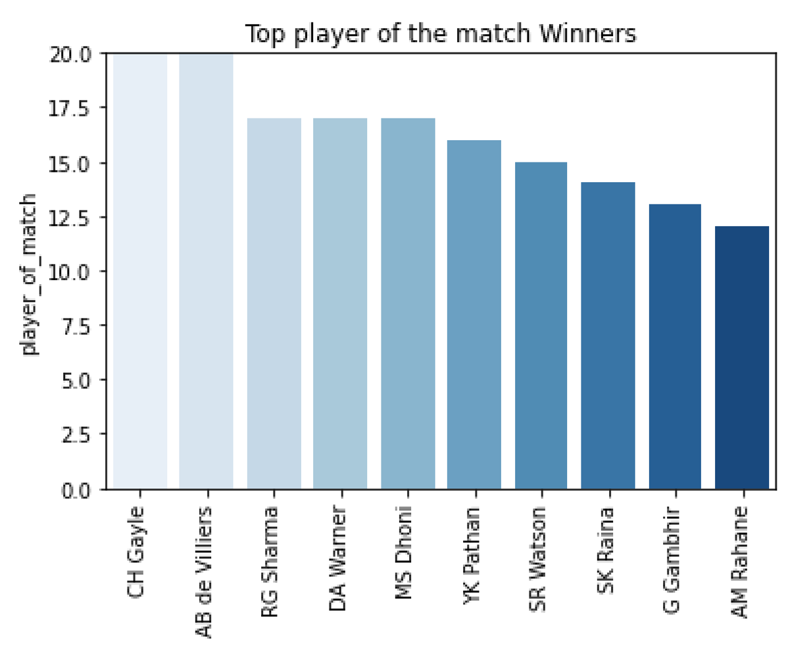
Out[16]:

win_by_runs	0	1	2	3	4	5	6	7	8	9	...	98	102	105	111	118	130	138	140	144	146
win_by_wickets	6.045346	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

1 rows × 89 columns

Top Player of the match winners

In [17]:
```
top_players = data.player_of_match.value_counts()[:10]
#sns.barplot(x="day", y="total_bill", data=df)
fig, ax = plt.subplots()
ax.set_ylim([0,20])
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="Blues");
plt.show()
```



By the above Observations we can see that CH Gayle is the most Successful player in all match winners