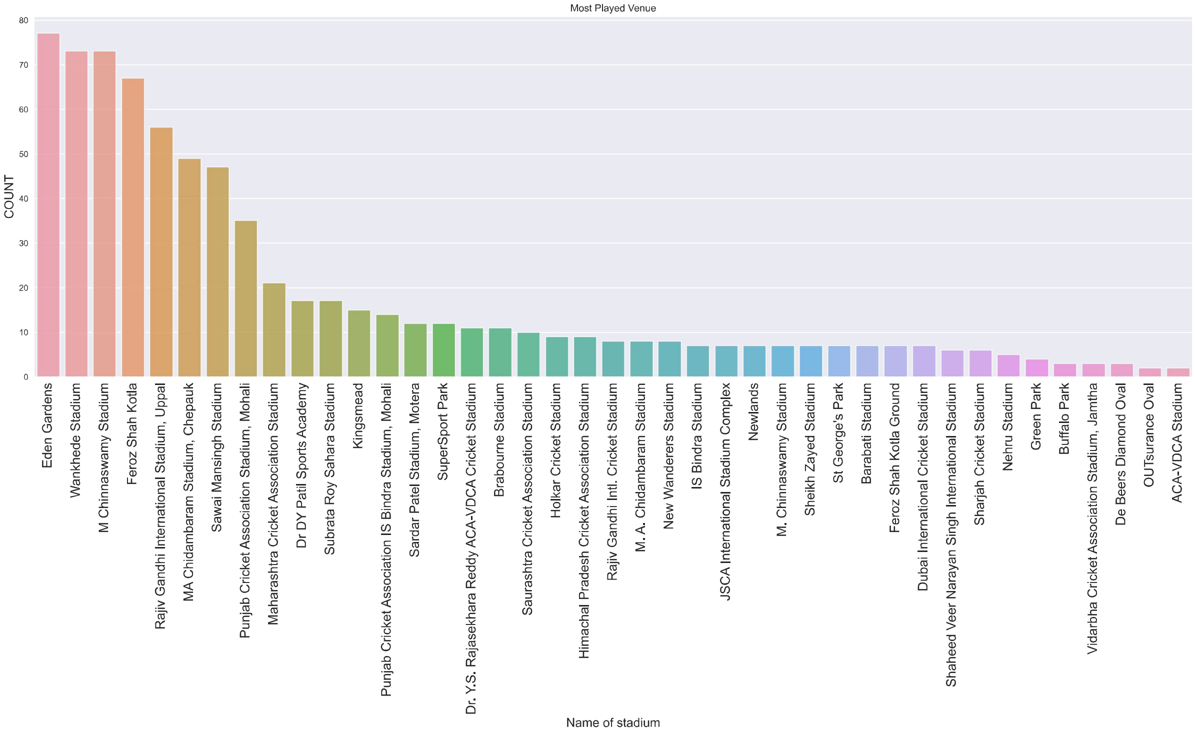
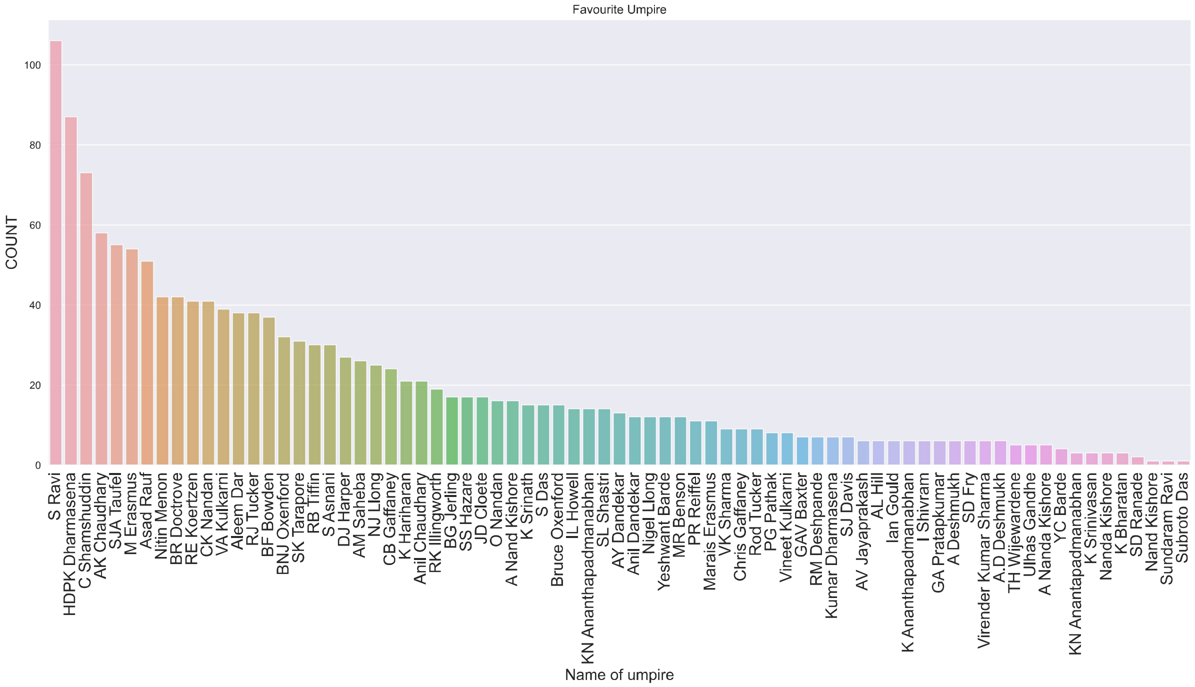
In [1]:

```python
# most used venue
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sb
import warnings

warnings.simplefilter(action ='ignore', category= FutureWarning)
del_df = pd.read_csv("deliveries.csv")
match_df = pd.read_csv("matches.csv")
sb.set_style('darkgrid')
ls = match_df['venue'].value_counts().sort_values(ascending = False)
plt.figure(figsize=(25,8))
temp = sb.barplot(ls.index, ls.values, alpha = 0.8)
plt.title("Most Played Venue")
plt.ylabel('COUNT', fontsize = 15)
plt.xlabel('Name of stadium', fontsize = 15)
temp.set_xticklabels(rotation = 90, labels = ls.index, fontsize = 16)
plt.show()
```



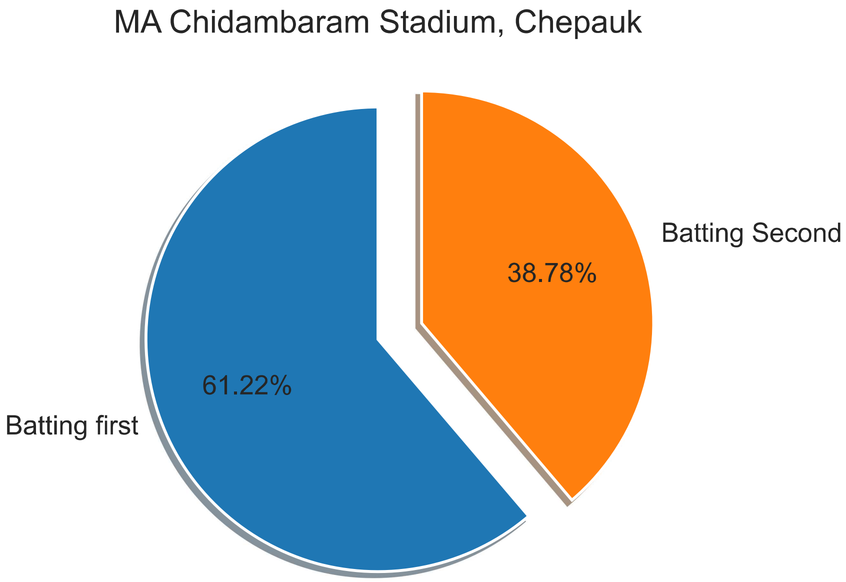
In [6]:

```
# favourite umpire
import numpy as numpy
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sb
import warnings

warnings.simplefilter(action ='ignore', category= FutureWarning)
del_df = pd.read_csv("deliveries.csv")
match_df = pd.read_csv("matches.csv")
sb.set_style('darkgrid')
ls = pd.concat([match_df['umpire1'],match_df['umpire2']]).value_counts().sort_
values(ascending = False)
plt.figure(figsize=(20,8))
most_umpired = sb.barplot(ls.index, ls.values, alpha = 0.7)
plt.title("Favourite Umpire")
plt.ylabel('COUNT', fontsize = 15)
plt.xlabel('Name of umpire', fontsize = 15)
most_umpired.set_xticklabels(rotation = 90, labels = ls.index, fontsize = 16)
#plt.figure(figsize=(25,8))
plt.show()
```



In [4]:

```
# Teams winning batting first or second on a particular venue
chennai_stadium = match_df.loc[(match_df['venue'] == 'MA Chidambaram Stadium, Chepauk')]
chennai_stadium_win_by_runs = chennai_stadium[chennai_stadium['win_by_runs'] > 0]
slices = [len(chennai_stadium_win_by_runs), len(chennai_stadium)-len(chennai_stadium_win_by_runs)]
labels = ['Batting first', 'Batting Second']
plt.pie(slices, labels = labels, startangle = 90 , shadow = 1, explode = (0,0.2), autopct = '%4.2f%%')
plt.title('MA Chidambaram Stadium, Chepauk')
plt.figure(figsize=(25,8))
plt.show()
```

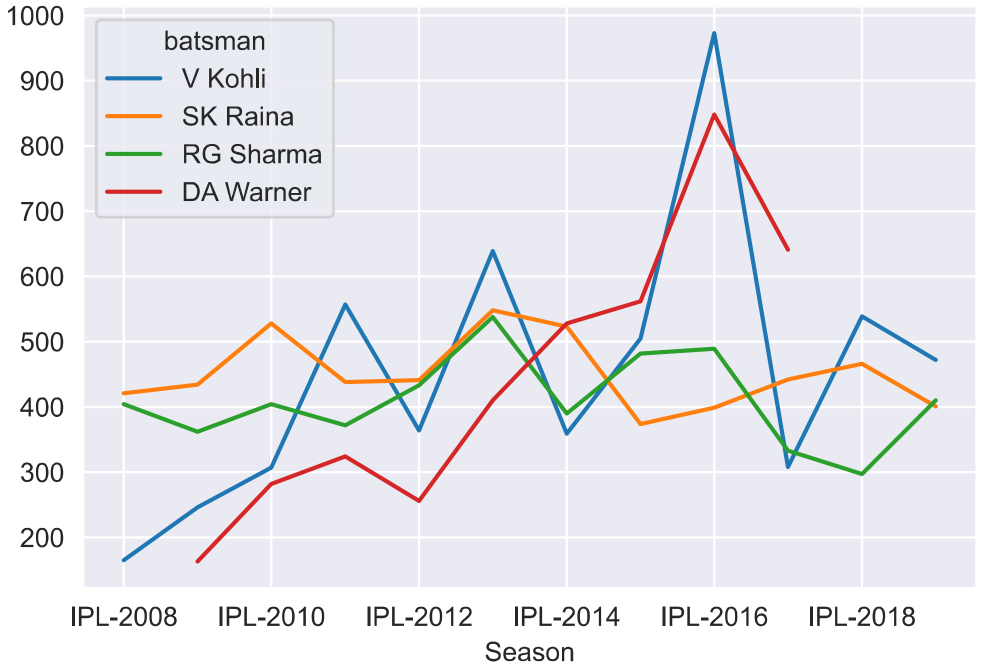


```
<Figure size 1800x576 with 0 Axes>
```

In [7]:

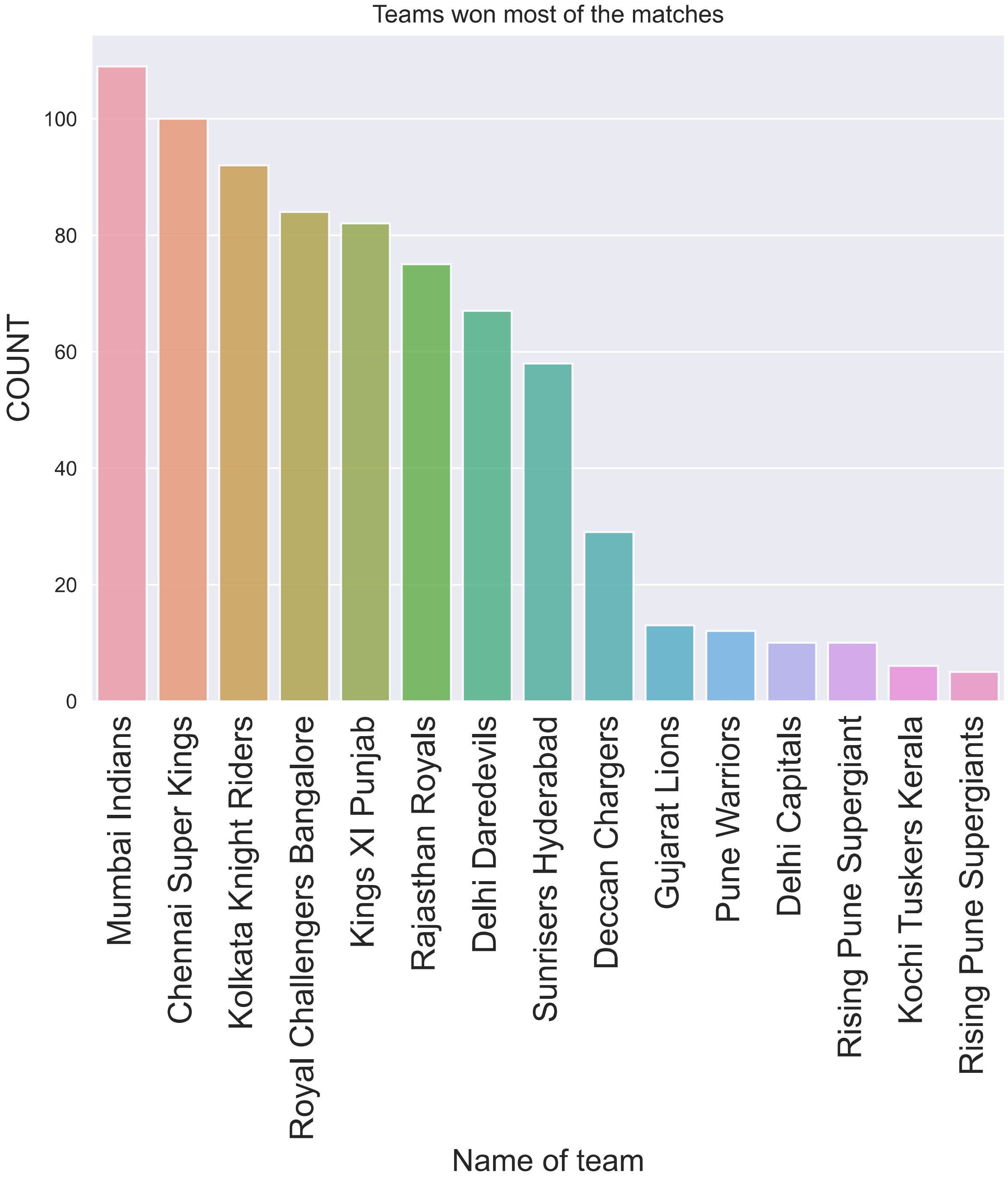
```
#performance of a particular batsman over years
import numpy as numpy
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sb
import warnings

warnings.simplefilter(action ='ignore', category= FutureWarning)
del_df = pd.read_csv("deliveries.csv")
match_df = pd.read_csv("matches.csv")

batsman_grp = del_df.groupby(['match_id', 'inning','batting_team','batsman'])
batsman = batsman_grp['batsman_runs'].sum().reset_index()
batsman = match_df[['id','Season']].merge(batsman, left_on ='id', right_on =
'match_id', how = 'left').drop('id', axis =1)

batsman_run_per_season = batsman.groupby(['Season','batting_team','batsman'])[
'batsman_runs'].sum().reset_index()
batsman_run_per_season = batsman_run_per_season.groupby(['Season', 'batsman'])
['batsman_runs'].sum().unstack().T
batsman_run_per_season['Total'] = batsman_run_per_season.sum(axis = 1)
batsman_run_per_season = batsman_run_per_season.sort_values(by = 'Total', asce
nding = False).drop('Total',1)
ax = batsman_run_per_season[:4].T.plot()
```

```
<Figure size 1800x576 with 0 Axes>
```



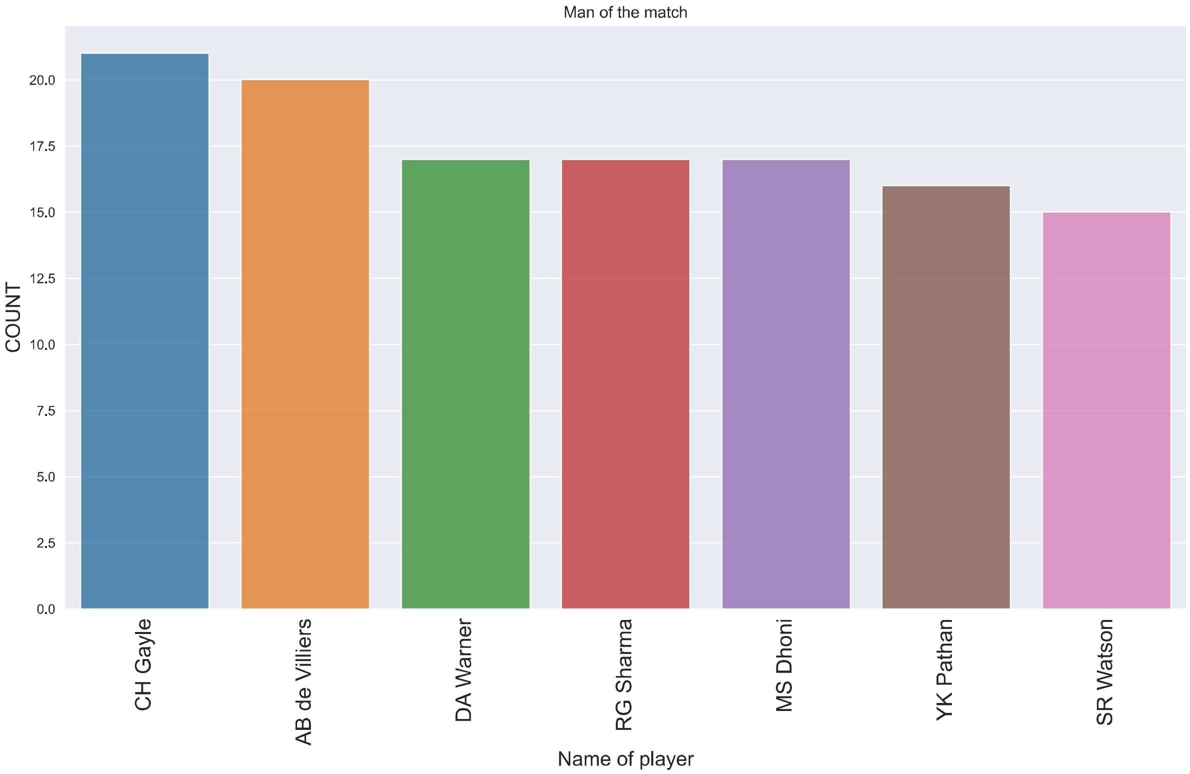
In [27]:

```python
# Team to win most number of matches
sb.set_style('darkgrid')
ls = match_df['winner'].value_counts().sort_values(ascending = False)
plt.figure(figsize=(8,6))
temp = sb.barplot(ls.index, ls.values, alpha = 0.8)
plt.title("Teams won most of the matches")
plt.ylabel('COUNT', fontsize = 15)
plt.xlabel('Name of team', fontsize = 15)
temp.set_xticklabels(rotation = 90, labels = ls.index, fontsize = 16)

plt.show()
```



In [31]:

```
# Most number of man of the match award
sb.set_style('darkgrid')
ls = match_df['player_of_match'].value_counts().sort_values(ascending = False)
ls1 = ls[:7]
plt.figure(figsize=(15,8))
temp = sb.barplot(ls1.index, ls1.values, alpha = 0.8)
plt.title("Man of the match")
plt.ylabel('COUNT', fontsize = 15)
plt.xlabel('Name of player', fontsize = 15)
temp.set_xticklabels(rotation = 90, labels = ls1.index, fontsize = 16)
plt.show()
```



In [76]:

```
#batsmans performance in a particular match
import numpy as numpy
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sb
import warnings
import mysql.connector

warnings.simplefilter(action ='ignore', category= FutureWarning)
del_df = pd.read_csv("deliveries.csv")
match_df = pd.read_csv("matches.csv")

batsman = del_df[del_df['batsman'] == 'Yuvraj Singh']
x = batsman[batsman['match_id'] == 1]
total_runs=x.groupby(['match_id'])['batsman_runs'].sum()
name = str(batsman['batsman'].unique())[1:-1]
print("Total runs scored: ",total_runs[1]," by ",name)
sixs = x[x['batsman_runs'] == 6]
print("Number of six hit:",sixs['batsman_runs'].count())
six = int(sixs['batsman_runs'].count())
fours = x[x['batsman_runs'] == 4]
print("Number of fours hit:",fours['batsman_runs'].count())
four = int(fours['batsman_runs'].count())
threes = x[x['batsman_runs'] == 3]
print("Number of three's taken:",threes['batsman_runs'].count())
three = int(threes['batsman_runs'].count())
twos = x[x['batsman_runs'] == 2]
print("Number of doubles taken:",twos['batsman_runs'].count())
two = int(twos['batsman_runs'].count())
ones = x[x['batsman_runs'] == 1]
print("Number of singles taken:",ones['batsman_runs'].count())
one = int(ones['batsman_runs'].count())

totalruns = int(total_runs[1])
# creating an connection
conn = sqlite3.connect("ipl.db") # db - database

# Cursor object
cursor = conn.cursor()

# code to create a databse table
# executing the above SQL code
#cursor.execute(create_table_sql)

# inserting data into the students table
insert_student_one_sql = """INSERT into batsman (id,name,runs,six,four,three,two,one) values (?,?,?,?,?,?,?,?)""";
cursor.execute(insert_student_one_sql , (1,name,totalruns,six,four,three,two,one))

# saving the changes using commit method of connection
conn.commit()

# closing the connection
conn.close()
```

```
Total runs scored:  62  by  'Yuvraj Singh'
Number of six hit: 3
Number of fours hit: 7
Number of three's taken: 0
Number of doubles taken: 3
Number of singles taken: 10
```

In [78]:

```
#batsmans overall performance
batsman = del_df[del_df['batsman'] == 'RG Sharma']
x = batsman
name = str(batsman['batsman'].unique())[1:-1]
total_runs=batsman['batsman_runs'].sum()
print("Total runs scored by",name,":",total_runs)
balls = x['batsman'].count()
print("Number of balls faced:",balls)
sr = (total_runs/balls)*100
print("Strike rate:%.2f"%sr)
sixs = x[x['batsman_runs'] == 6]
six = int(sixs['batsman_runs'].count())
print("Number of six hit:",sixs['batsman_runs'].count())
fours = x[x['batsman_runs'] == 4]
four = int(fours['batsman_runs'].count())
print("Number of fours hit:",fours['batsman_runs'].count())
threes = x[x['batsman_runs'] == 3]
three = int(threes['batsman_runs'].count())
print("Number of three's taken:",threes['batsman_runs'].count())
twos = x[x['batsman_runs'] == 2]
print("Number of doubles taken:",twos['batsman_runs'].count())
ones = x[x['batsman_runs'] == 1]
print("Number of singles taken:",ones['batsman_runs'].count())

two = int(twos['batsman_runs'].count())
one = int(ones['batsman_runs'].count())
totalruns = int(total_runs)

# creating an connection
conn = sqlite3.connect("ipl.db") # db - database
# Cursor object
cursor = conn.cursor()
#cursor.execute(create_table_sql)
#create_table_sql = "CREATE TABLE Overall_batsman (name VARCHAR(30),runs INTEG
ER,six INTEGER,four INTEGER,three INTEGER,two INTEGER,one INTEGER);"
#cursor.execute(create_table_sql)

# inserting data into the students table

insert_student_one_sql = """INSERT into Overall_batsman (name,runs,six,four,th
ree,two,one) values (?,?,?,?,?,?,?)""";
cursor.execute(insert_student_one_sql , (name,totalruns,six,four,three,two,one
))

# saving the changes using commit method of connection
conn.commit()

# closing the connection
conn.close()
```

```
Total runs scored by 'RG Sharma' : 4914
Number of balls faced: 3816
Strike rate:128.77
Number of six hit: 194
Number of fours hit: 431
Number of three's taken: 5
Number of doubles taken: 205
Number of singles taken: 1589
```

In [86]:

```
# batsman vs bowler
batsman = del_df[del_df['batsman'] == 'MS Dhoni']
#x = batsman[batsman['batsman']]
y = batsman[batsman['bowler'] == 'AB Dinda']
coun = int(y['player_dismissed'].count())
btname = str(batsman['batsman'].unique())[1:-1]
blname = str(y['bowler'].unique())[1:-1]
print(btname, "VS", blname)
balls_faced = int(y['batsman_runs'].count())
print("Balls Faced:",balls_faced)
runs_scored = int(y['batsman_runs'].sum())
SR = int((runs_scored/balls_faced)*100)
print("Strike Rate:",SR)
print("Runs scored:",runs_scored)
print("Got out:",coun,"times")

# creating an connection
conn = sqlite3.connect("ipl.db") # db - database
# Cursor object
cursor = conn.cursor()
#cursor.execute(create_table_sql)
#create_table_sql = "CREATE TABLE btvbl (batsman VARCHAR(30),bowler VARCHAR(3
0),balls INTEGER,sr INTEGER,runs_scored INTEGER,out INTEGER);"
#cursor.execute(create_table_sql)

# inserting data into the students table

insert_student_one_sql = """INSERT into btvbl (batsman,bowler,balls,sr,runs_sc
ored,out) values (?,?,?,?,?,?)""";
cursor.execute(insert_student_one_sql , (btname,blname,balls_faced,SR,runs_sco
red,coun))

# saving the changes using commit method of connection
conn.commit()

# closing the connection
conn.close()
```

```
'MS Dhoni' VS 'AB Dinda'
Balls Faced: 24
Strike Rate: 170
Runs scored: 41
Got out: 2 times
```

In [103]:

```
#Bowlers performance
bowler = del_df[del_df['bowler'] == 'R Ashwin']
blname = str(bowler['bowler'].unique())[1:-1]
print("Name of bowler:",blname)
deliveries = int(bowler['bowler'].count())
print("Number of deliveries bowled:",deliveries)
run = int(bowler['total_runs'].sum())
econ = run/(deliveries/6)
print("Economy rate: %.2f"% econ)
print("Total number of runs consumed:",run)
wickets = int(bowler['player_dismissed'].count())
print("Wickets Taken:",wickets)


# creating an connection
conn = sqlite3.connect("ipl.db") # db - database
# Cursor object
cursor = conn.cursor()
#create_table_sql = "CREATE TABLE Overall_bowler (name VARCHAR(30),bowls INTEG
ER,economy_rate INTEGER,runs_consumed INTEGER,wickets INTEGER);"
#cursor.execute(create_table_sql)

insert_student_one_sql = """INSERT into Overall_bowler (name,bowls,economy_rat
e,runs_consumed,wickets) values (?,?,?,?,?)""";
cursor.execute(insert_student_one_sql , (blname,deliveries,econ,run,wickets))
conn.commit()

# closing the connection
conn.close()
```

```
Name of bowler: 'R Ashwin'
Number of deliveries bowled: 3016
Economy rate: 6.75
Total number of runs consumed: 3391
Wickets Taken: 138
```