

IPL - 2017

A Data Analysis on the Indian Premier League

Table of contents

1. What are we trying to solve ?
2. Data
3. Analysis of Data
4. Insights and Conclusion

What do we need to solve?

We need to get the analysis for the following:-

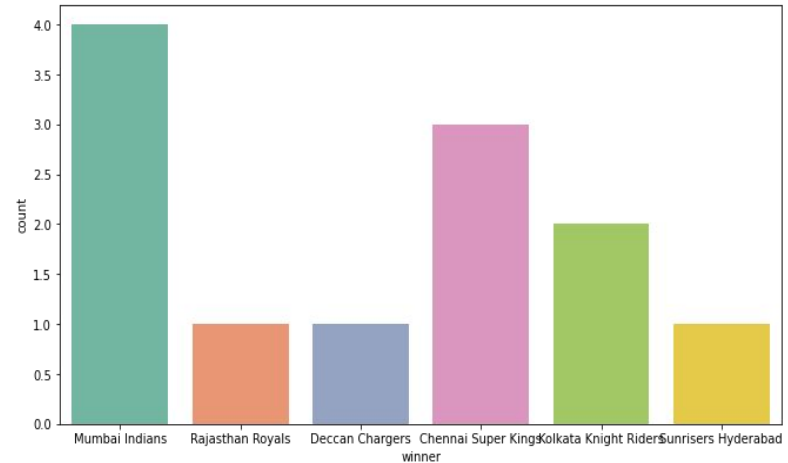
1. The most successful team in the tournament.
2. The most successful players in the tournament, with the most runs and wickets.
3. Teams and Players companies are most likely to endorse.

The DATA

The IPL data consists of factors such as the name of the teams, toss decisions, winners, man of the match, total runs, total wickets.

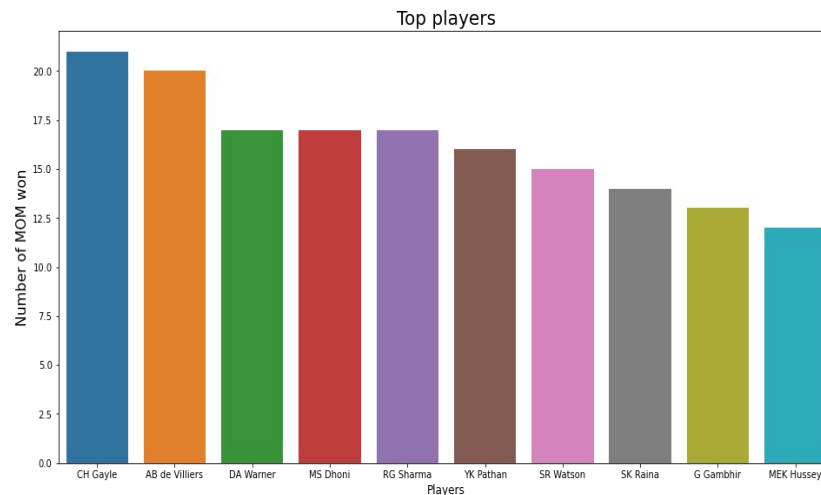
Number of IPL tournaments each teams have won over the years.

- Mumbai Indians have 4 wins, followed by CSK with 3 wins and KKR with 2 wins.
- The top 3 teams can be taken into account for company endorsements.

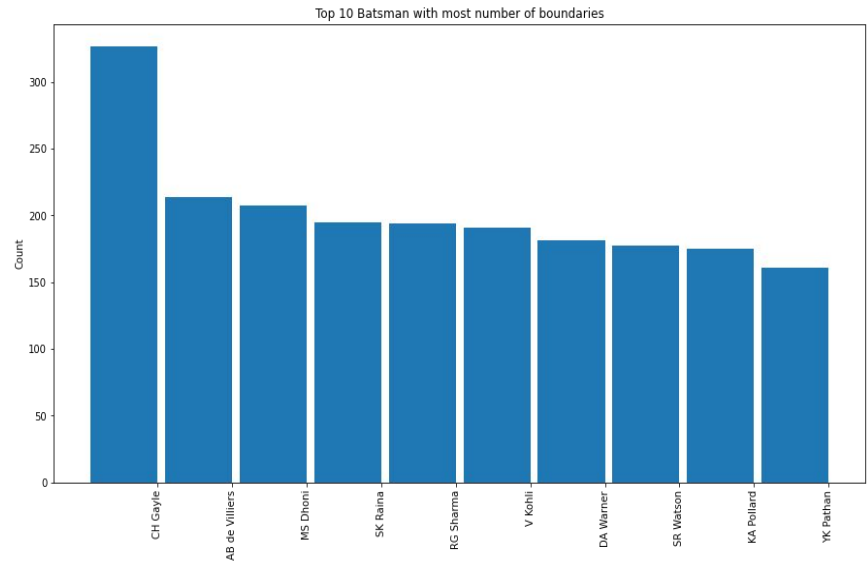


Top players.

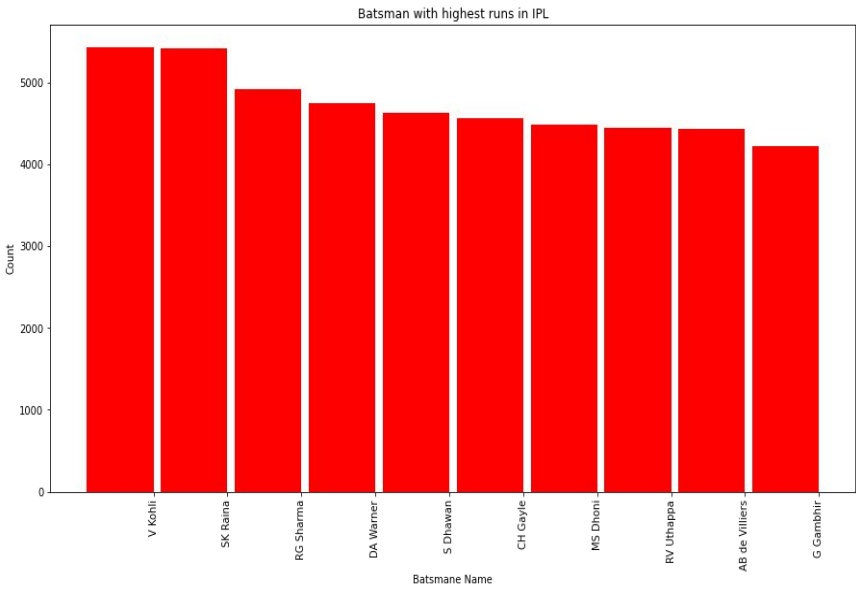
- Man of the matches won by each player is taken into account for determining the top players.
- Gayle, ABD and Warner are the top 3 players with the most man of the matches.



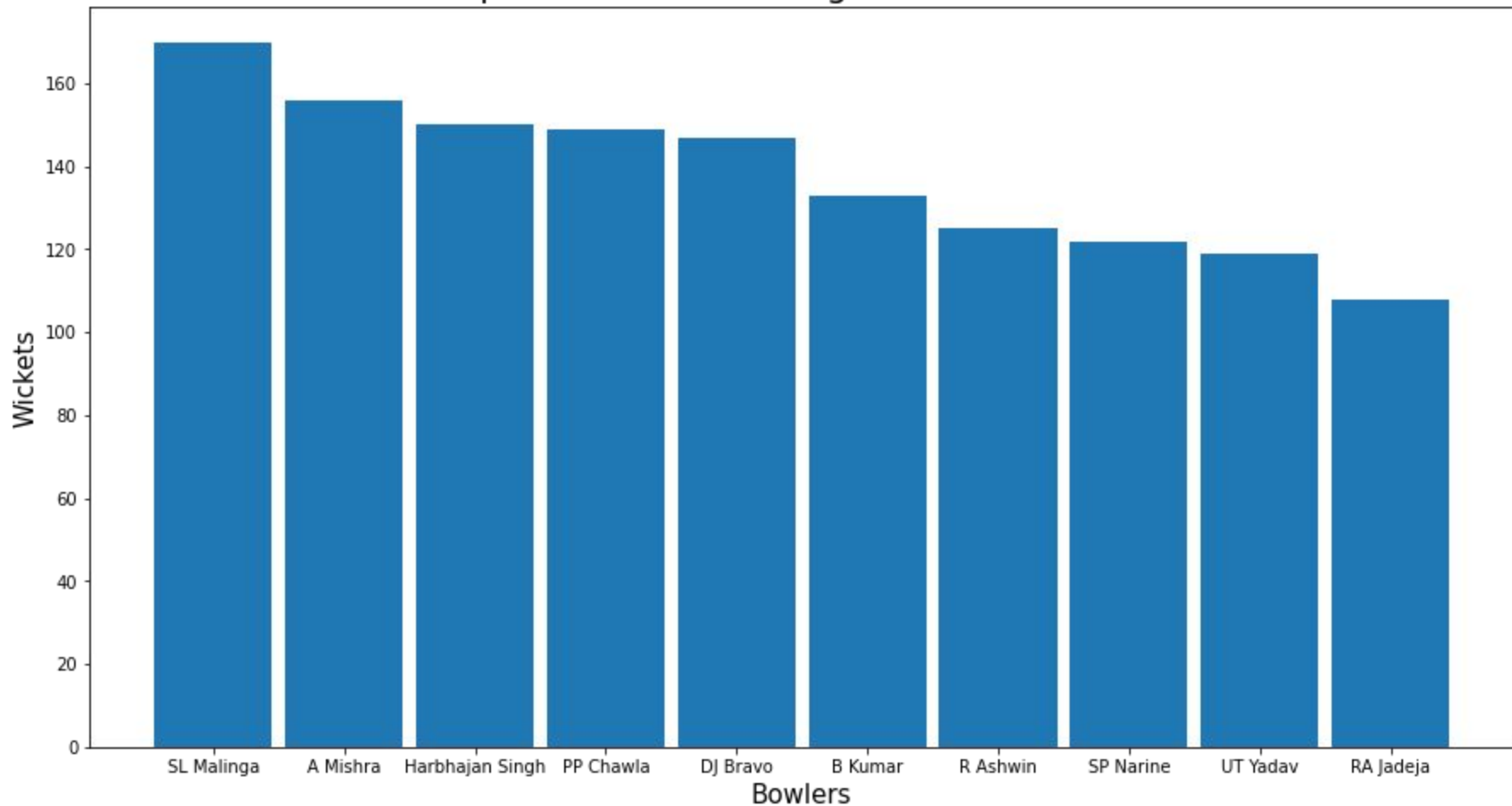
Top batsmen with most boundaries



Top batsmen with highest runs

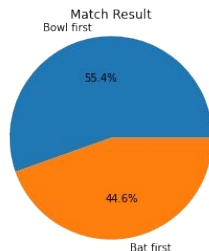


Top 10 Bowlers with highest wickets in IPL

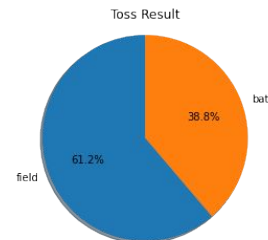


Toss and key decisions that affected the game

- 55.4 % teams that bowled first won the match.
- 44.6% teams won that batted first.



- 61.2% teams that won the toss and chose to field won the game.
- 38.8% teams that won the toss and chose to bat won the game.



INSIGHTS

- Companies should focus on endorsing Mumbai Indians, Chennai Super Kings and KKR. The top three teams to win the IPL.
- Batsmen to endorse are Gayle, ABD, MS Dhoni, V Kohli and Raina.
- Bowlers to endorse Malinga, Mishra and Harbhajan Singh.



CODE

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
In [2]: matches = pd.read_csv("../input/ipl-dataset/matches.csv")
dels = pd.read_csv("../input/ipl-dataset/deliveries.csv")
matches.head()
```

Out[2]:

| | id | season | city | date | team1 | team2 | toss_winner | toss_decision | result | dl_applied | winner |
|---|----|--------|-----------|------------|-----------------------------|-----------------------------|-----------------------------|---------------|--------|------------|-----------------------------|
| 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 |

```
In [3]: dels.head()
```

Out[3]:

| | match_id | inning | batting_team | bowling_team | over | ball | batsman | non_striker | bowler | is_super_over | ... | bye_runs |
|--|----------|--------|--------------|--------------|------|------|---------|-------------|--------|---------------|-----|----------|
|--|----------|--------|--------------|--------------|------|------|---------|-------------|--------|---------------|-----|----------|

```
In [3]: dels.head()
```

Out[3]:

| | match_id | inning | batting_team | bowling_team | over | ball | batsman | non_striker | bowler | is_super_over | ... | bye_runs |
|---|----------|--------|---------------------|-----------------------------|------|------|-----------|-------------|----------|---------------|-----|----------|
| 0 | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 1 | DA Warner | S Dhawan | TS Mills | 0 | ... | 0 |
| 1 | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 2 | DA Warner | S Dhawan | TS Mills | 0 | ... | 0 |
| 2 | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 3 | DA Warner | S Dhawan | TS Mills | 0 | ... | 0 |
| 3 | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 4 | DA Warner | S Dhawan | TS Mills | 0 | ... | 0 |
| 4 | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 5 | DA Warner | S Dhawan | TS Mills | 0 | ... | 0 |

5 rows × 21 columns

```
In [4]: matches.isnull().sum()
```

Out[4]:

```
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
```

```
In [5]: dels.isnull().sum()
```

```
Out[5]:
```

| | |
|------------------|--------|
| match_id | 0 |
| inning | 0 |
| batting_team | 0 |
| bowling_team | 0 |
| over | 0 |
| ball | 0 |
| batsman | 0 |
| non_striker | 0 |
| bowler | 0 |
| is_super_over | 0 |
| wide_runs | 0 |
| bye_runs | 0 |
| legbye_runs | 0 |
| noball_runs | 0 |
| penalty_runs | 0 |
| batsman_runs | 0 |
| extra_runs | 0 |
| total_runs | 0 |
| player_dismissed | 170244 |
| dismissal_kind | 170244 |
| fielder | 172630 |

dtype: int64

```
In [6]: matches["team1"].unique()
```

```
Out[6]:
```

```
array(['Sunrisers Hyderabad', 'Mumbai Indians', 'Gujarat Lions',  
      'Rising Pune Supergiant', 'Royal Challengers Bangalore',  
      'Kolkata Knight Riders', 'Delhi Daredevils', 'Kings XI Punjab',  
      'Chennai Super Kings', 'Rajasthan Royals', 'Deccan Chargers',  
      'Kochi Tuskers Kerala', 'Pune Warriors', 'Rising Pune Supergiants',  
      'Delhi Capitals'], dtype=object)
```

```
In [7]: matches.replace('Rising Pune Supergiant', 'Rising Pune Supergiants', inplace=True)
```

```
In [8]: matches["team1"].unique()
```

```
Out[8]:
```

```
array(['Sunrisers Hyderabad', 'Mumbai Indians', 'Gujarat Lions',  
      'Rising Pune Supergiants', 'Royal Challengers Bangalore',  
      'Kolkata Knight Riders', 'Delhi Daredevils', 'Kings XI Punjab',  
      'Chennai Super Kings', 'Rajasthan Royals', 'Deccan Chargers',  
      'Kochi Tuskers Kerala', 'Pune Warriors', 'Delhi Capitals'],  
      dtype=object)
```

```
In [9]: dels["bowling_team"].unique()
```

```
Out[9]:
```

```
array(['Royal Challengers Bangalore', 'Sunrisers Hyderabad',  
      'Rising Pune Supergiant', 'Mumbai Indians',  
      'Kolkata Knight Riders', 'Gujarat Lions', 'Kings XI Punjab',  
      'Delhi Daredevils', 'Chennai Super Kings', 'Rajasthan Royals',  
      'Deccan Chargers', 'Kochi Tuskers Kerala', 'Pune Warriors',  
      'Rising Pune Supergiants', 'Delhi Capitals'], dtype=object)
```

```
In [10]: dels.replace('Rising Pune Supergiant', 'Rising Pune Supergiants', inplace=True)
```

```
In [11]: dels["bowling_team"].unique()
```

```
Out[11]:
```

```
array(['Royal Challengers Bangalore', 'Sunrisers Hyderabad',  
      'Rising Pune Supergiants', 'Mumbai Indians',  
      'Kolkata Knight Riders', 'Gujarat Lions', 'Kings XI Punjab',  
      'Delhi Daredevils', 'Chennai Super Kings', 'Rajasthan Royals',  
      'Deccan Chargers', 'Kochi Tuskers Kerala', 'Pune Warriors',  
      'Delhi Capitals'], dtype=object)
```

```
In [12]: matches["city"].unique()
```

```
Out[12]:
```

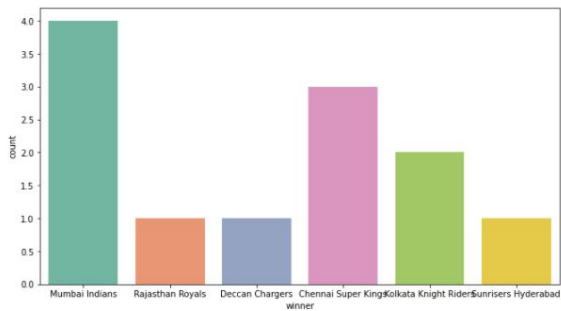
```
array(['Hyderabad', 'Pune', 'Rajkot', 'Indore', 'Bangalore', 'Mumbai',  
      'Kolkata', 'Delhi', 'Chandigarh', 'Kanpur', 'Jaipur', 'Chennai',  
      'Cape Town', 'Port Elizabeth', 'Durban', 'Centurion'],  
      dtype=object)
```

```
In [13]: matches.replace('Bangalore','Bengaluru', inplace=True)
```

```
In [14]: matches["city"].unique()
```

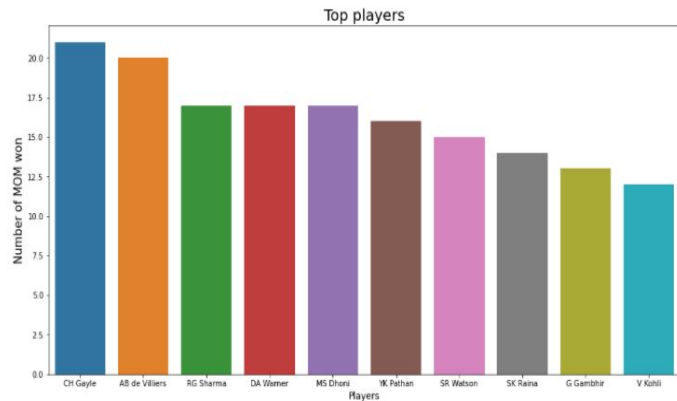
```
Out[14]: array(['Hyderabad', 'Pune', 'Rajkot', 'Indore', 'Bengaluru', 'Mumbai',  
       'Kolkata', 'Delhi', 'Chandigarh', 'Kanpur', 'Jaipur', 'Chennai',  
       'Cape Town', 'Port Elizabeth', 'Durban', 'Centurion',  
       'East London', 'Johannesburg', 'Kimberley', 'Bloemfontein',  
       'Ahmedabad', 'Cuttack', 'Nagpur', 'Dharamsala', 'Kochi',  
       'Visakhapatnam', 'Raipur', 'Ranchi', 'Abu Dhabi', 'Sharjah', nan,  
       'Mohali'], dtype=object)
```

```
In [15]: plt.subplots(figsize=(11,6))  
wins = matches.drop_duplicates('season', keep='last')  
ax=sns.countplot(x='winner', data = wins, palette = 'Set2')  
plt.show()
```



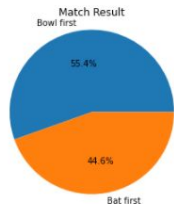
```
In [16]: plt.subplots(figsize=(16,8))
```

```
In [16]: plt.subplots(figsize=(16,8))  
ax = sns.barplot(x = matches['player_of_match'].value_counts()[:10].index, y = matches['player_of_match'].value_counts()[:10])  
plt.ylabel('Number of MOM won', fontsize=15)  
plt.title('Top players', fontsize=20)  
plt.xlabel('Players', fontsize=12)  
plt.show()
```

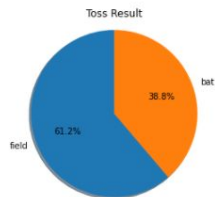


```
In [17]: matches['win_by']=np.where(matches['win_by_runs']>0,'Bat first','Bow1 first')  
match=matches.win_by.value_counts()  
labels=np.array(match.index)  
sizes = match.values  
plt.pie(sizes, labels=labels, autopct='%1.1f%%',
```

```
In [17]: matches['win_by']=np.where(matches['win_by_runs']>0,'Bat first','Bowl first')
match=matches.win_by.value_counts()
labels=np.array(match.index)
sizes = match.values
#colors = ['red', 'blue']
plt.pie(sizes, labels=labels, autopct='%1.1f%%' )
plt.title('Match Result')
plt.axis('equal')
plt.show()
```

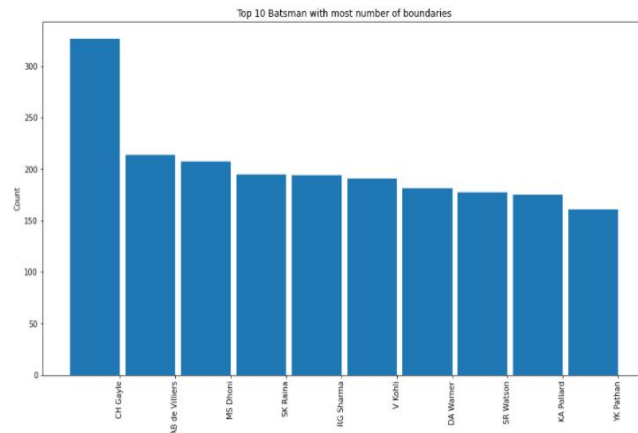


```
In [18]: toss = matches.toss_decision.value_counts()
labels = np.array(toss.index)
sizes = toss.values
plt.pie(sizes, labels=labels, autopct='%1.1f%%', shadow=True, startangle=90)
plt.title('Toss Result')
plt.axis('equal')
plt.show()
```



```
In [19]: batsman_df = dels.groupby('batsman')['batsman_runs'].agg(lambda x: (x==4).sum() and (x==6).sum())
.reset_index().sort_values(by='batsman_runs', ascending=False).reset_index(drop=True)
batsman_df = batsman_df.iloc[:10,:]
```

```
labels = np.array(batsman_df['batsman'])
ind = np.arange(len(labels))
width = 0.9
fig, ax = plt.subplots(figsize=(15,8))
rects = ax.bar(ind, np.array(batsman_df['batsman_runs']), width=width)
ax.set_xticks(ind+(width)/2.)
ax.set_xticklabels(labels, rotation='vertical')
ax.set_ylabel("Count")
ax.set_title("Top 10 Batsman with most number of boundaries")
plt.show()
```

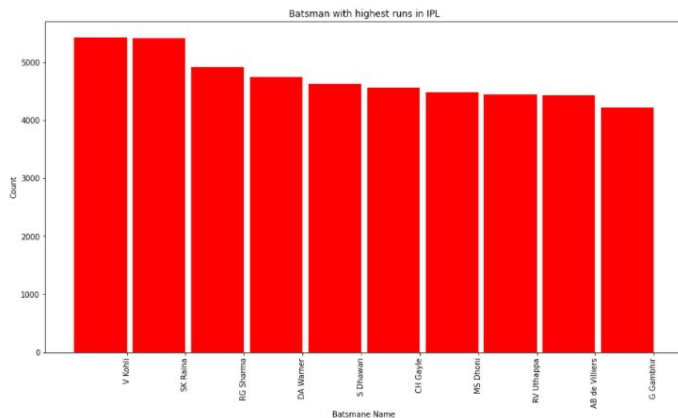


```
In [20]:
```

In [20]:

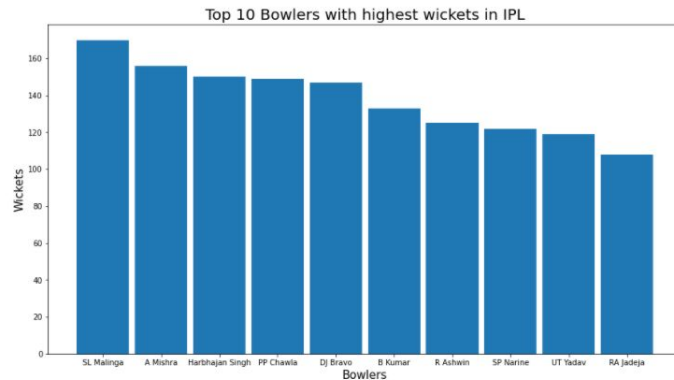
```
batsman_df = dels.groupby('batsman')['batsman_runs'].agg('sum').reset_index().sort_values(by='batsman_runs', ascending=False).reset_index(drop=True)
batsman_df = batsman_df.iloc[:10,:]
```

```
labels = np.array(batsman_df['batsman'])
ind = np.arange(len(labels))
width = 0.9
fig, ax = plt.subplots(figsize=(15,8))
rects = ax.bar(ind, np.array(batsman_df['batsman_runs']), width=width, color = 'red')
ax.set_xticks(ind+(width)/2.)
ax.set_xticklabels(labels, rotation='vertical')
ax.set_ylabel("Count")
ax.set_title("Batsman with highest runs in IPL")
ax.set_xlabel('Batsman Name')
plt.show()
```



In [21]:

```
bowling_wickets= dels[dels['dismissal_kind']!='run out']
bowling_tot=bowling_wickets.groupby('bowler').apply(lambda x:x['dismissal_kind'].dropna()).reset_index(name='Wickets')
bowling_wick_count=bowling_tot.groupby('bowler').count().reset_index()
bowling_top=bowling_wick_count.sort_values(by='Wickets', ascending=False)
top_bowlers=bowling_top.loc[:,['bowler','Wickets']][0:10]
fig, ax = plt.subplots(figsize=(15,8))
ax.bar(top_bowlers['bowler'],top_bowlers['Wickets'], width = 0.9)
ax.set_xlabel('Bowlers',size=15)
ax.set_ylabel('Wickets',size=15)
ax.set_title('Top 10 Bowlers with highest wickets in IPL',size=20)
plt.show()
```



In []: