

In [1]: `#2008-2019`

In []: `#!pip install pandas==1.4.3`

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

In [3]: `ipl_df = pd.read_csv("matches.csv")
ipl_df.head()`

Out[3]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0
1	2	IPL-2017	Pune	06-04-2017	Mumbai Indians	Rising Pune Supergiants	Rising Pune Supergiants	field	normal	0
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0
3	4	IPL-2017	Indore	08-04-2017	Rising Pune Supergiants	Kings XI Punjab	Kings XI Punjab	field	normal	0
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0

In [4]: `ipl_df.shape`

Out[4]: (756, 18)

In [5]: `ipl_df.info()`

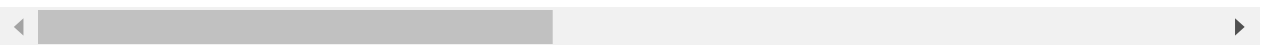
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    756 non-null    int64
1   Season                756 non-null    object
2   city                  749 non-null    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(4), object(14)
memory usage: 106.4+ KB
```

In [6]: `deliveries_df = pd.read_csv("deliveries.csv")`
`deliveries_df.head()`

Out[6]:

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

5 rows × 21 columns



In [7]: `deliveries_df.shape`

Out[7]: (179078, 21)

In [8]: `deliveries_df.info()`

```
<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
```

In [9]: `discard_columns = ["umpire1", "umpire2", "umpire3"]`
`ipl_df.drop(discard_columns, axis=1, inplace=True)`

In [10]: `ipl_df.shape`

Out[10]: (756, 15)

In [11]: `ipl_df.result.value_counts()`

Out[11]: normal 743
tie 9
no result 4
Name: result, dtype: int64

Number of matches hosted in different cities

```
In [12]: city_host = ipl_df.city.value_counts()
city_host.head()
```

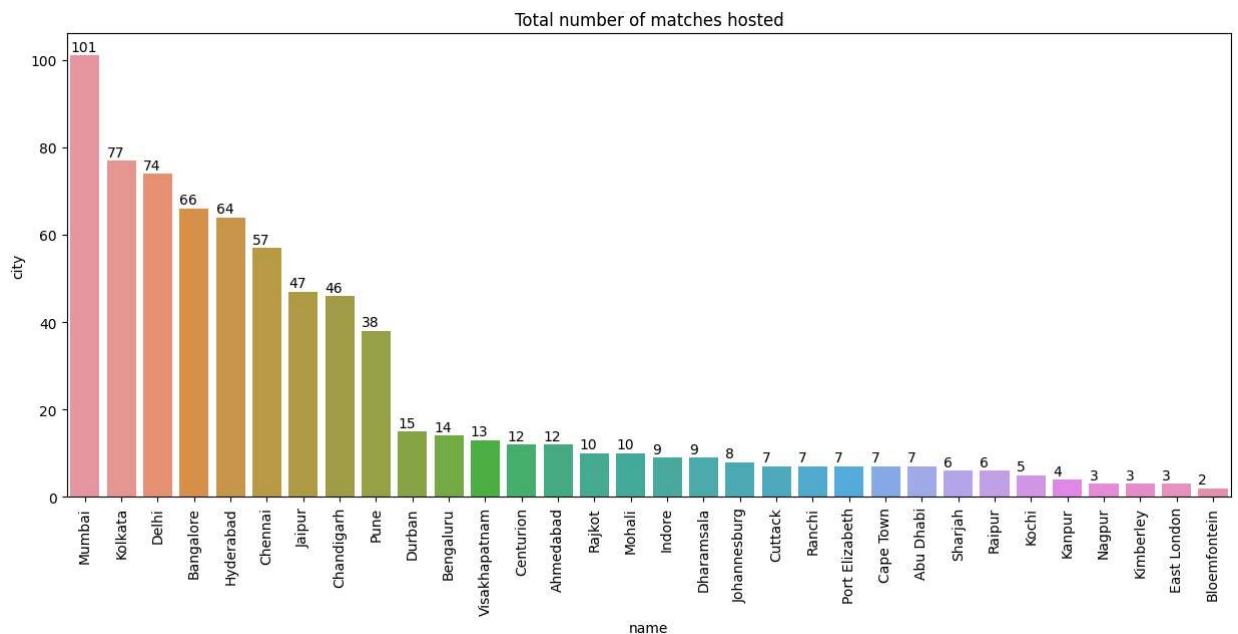
```
Out[12]: Mumbai      101
Kolkata      77
Delhi        74
Bangalore    66
Hyderabad    64
Name: city, dtype: int64
```

```
In [13]: city_host.index
```

```
Out[13]: Index(['Mumbai', 'Kolkata', 'Delhi', 'Bangalore', 'Hyderabad', 'Chennai',
'Jaipur', 'Chandigarh', 'Pune', 'Durban', 'Bengaluru', 'Visakhapatnam',
'Centurion', 'Ahmedabad', 'Rajkot', 'Mohali', 'Indore', 'Dharamsala',
'Johannesburg', 'Cuttack', 'Ranchi', 'Port Elizabeth', 'Cape Town',
'Abu Dhabi', 'Sharjah', 'Raipur', 'Kochi', 'Kanpur', 'Nagpur',
'Kimberley', 'East London', 'Bloemfontein'],
dtype='object')
```

```
In [15]: # cities
```

```
In [16]: plt.figure(figsize=(15,6))
plt.xticks(rotation = 90)
plt.title("Total number of matches hosted")
count = 0
cities = pd.DataFrame(city_host)
cities['name'] = cities.index
for i in cities['city']:
    plt.text(count- 0.4,i+1, str(i),color='black')
    count+=1
sns.barplot(cities, y= 'city',x='name' )
plt.show()
```



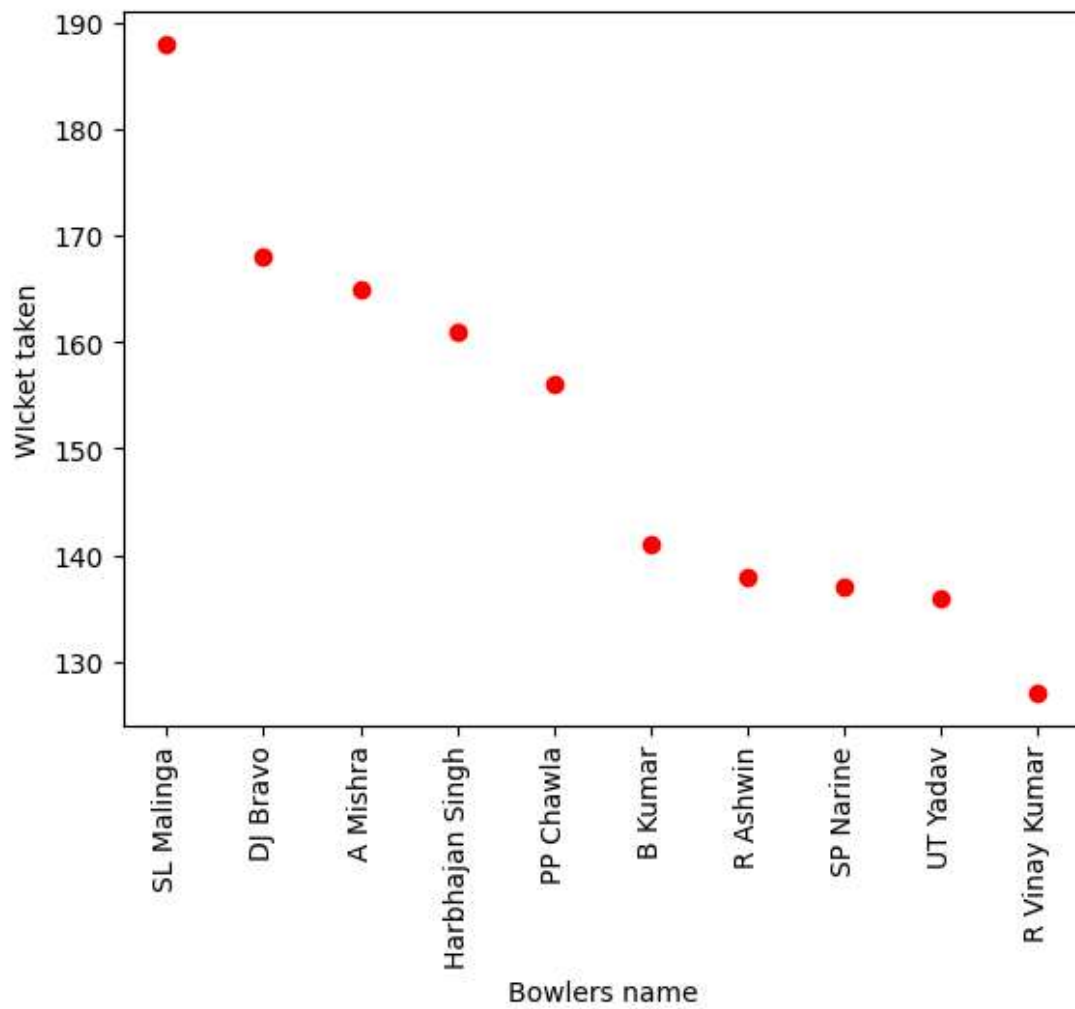
Maximum wickets taken by bowler in last 10 seasons

```
In [17]: bowling_wickets = deliveries_df[deliveries_df["dismissal_kind"]!='run_out']
bowling_total = bowling_wickets.groupby('bowler').apply(lambda x : x["dismissal_kind"]
bowling_wicket_count = bowling_total.groupby('bowler').count().reset_index()
bowling_top = bowling_wicket_count.sort_values(by='wickets',ascending=False)
top_bowlers = bowling_top.loc[:,["bowler","wickets"]][0:10]
top_bowlers
```

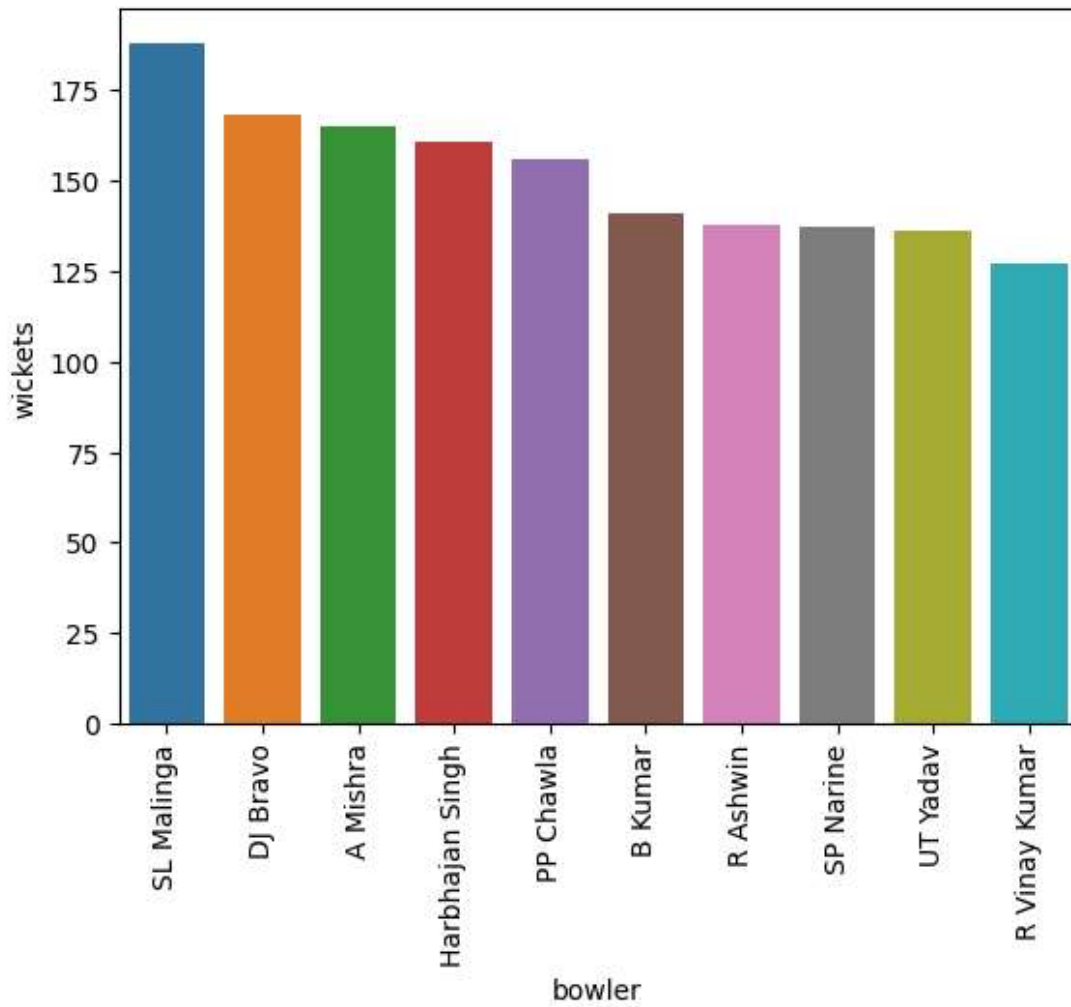
Out[17]:

	bowler	wickets
294	SL Malinga	188
84	DJ Bravo	168
6	A Mishra	165
113	Harbhajan Singh	161
231	PP Chawla	156
48	B Kumar	141
236	R Ashwin	138
300	SP Narine	137
330	UT Yadav	136
247	R Vinay Kumar	127

```
In [18]: plt.scatter(top_bowlers["bowler"],top_bowlers["wickets"],color='r')
plt.xticks(rotation= 90)
plt.xlabel("Bowlers name")
plt.ylabel("Wicket taken")
plt.show()
```



```
In [19]: sns.barplot(top_bowlers ,x = top_bowlers["bowler"],y = top_bowlers["wickets"])  
plt.xticks(rotation= 90)  
plt.show()
```



Matches Played vs Matches Won

```
In [41]: matches_played=pd.concat([ipl_df['team1'],ipl_df['team2']])
matches_played=matches_played.value_counts().reset_index()
matches_played.columns=['Team','Total Matches']
matches_played['wins']=ipl_df['winner'].value_counts().reset_index()['winner']
matches_played.set_index('Team',inplace=True)
matches_played
```

Out[41]:

	Total Matches	wins
Team		
Mumbai Indians	187	109
Royal Challengers Bangalore	180	100
Kolkata Knight Riders	178	92
Kings XI Punjab	176	84
Chennai Super Kings	164	82
Delhi Daredevils	161	75
Rajasthan Royals	147	67
Sunrisers Hyderabad	108	58
Deccan Chargers	75	29
Pune Warriors	46	15
Gujarat Lions	30	13
Rising Pune Supergiants	30	12
Delhi Capitals	16	10
Kochi Tuskers Kerala	14	6


```
In [42]: matches_played.reset_index()
```

```
Out[42]:
```

	Team	Total Matches	wins
0	Mumbai Indians	187	109
1	Royal Challengers Bangalore	180	100
2	Kolkata Knight Riders	178	92
3	Kings XI Punjab	176	84
4	Chennai Super Kings	164	82
5	Delhi Daredevils	161	75
6	Rajasthan Royals	147	67
7	Sunrisers Hyderabad	108	58
8	Deccan Chargers	75	29
9	Pune Warriors	46	15
10	Gujarat Lions	30	13
11	Rising Pune Supergiants	30	12
12	Delhi Capitals	16	10
13	Kochi Tuskers Kerala	14	6

```
In [46]: win_percentage = round(matches_played['wins']/matches_played['Total Matches'],3) *100
win_percentage
```

```
Out[46]: Team
Mumbai Indians          58.3
Royal Challengers Bangalore  55.6
Kolkata Knight Riders     51.7
Kings XI Punjab          47.7
Chennai Super Kings       50.0
Delhi Daredevils          46.6
Rajasthan Royals          45.6
Sunrisers Hyderabad       53.7
Deccan Chargers           38.7
Pune Warriors             32.6
Gujarat Lions             43.3
Rising Pune Supergiants    40.0
Delhi Capitals            62.5
Kochi Tuskers Kerala      42.9
dtype: float64
```

Maximum Runs Scored By A Team

```
In [61]: high_scores = deliveries_df.groupby(['match_id', 'inning', 'batting_team', 'bowling_team'])
high_scores = high_scores[high_scores['total_runs'] >= 200]
# high_scores
high_scores.nlargest(10, 'total_runs')
```

Out[61]:

	match_id	inning	batting_team	bowling_team	total_runs
829	411	1	Royal Challengers Bangalore	Pune Warriors	263
1370	7937	1	Kolkata Knight Riders	Kings XI Punjab	250
1250	620	1	Royal Challengers Bangalore	Gujarat Lions	248
416	206	1	Chennai Super Kings	Rajasthan Royals	246
1498	11338	1	Kolkata Knight Riders	Mumbai Indians	241
122	61	1	Chennai Super Kings	Kings XI Punjab	240
1134	562	1	Royal Challengers Bangalore	Mumbai Indians	235
1426	11147	1	Sunrisers Hyderabad	Royal Challengers Bangalore	233
596	296	1	Kings XI Punjab	Royal Challengers Bangalore	232
1500	11339	1	Sunrisers Hyderabad	Kings XI Punjab	232

Q1. which team won the maximum number of matches in all seasons?

```
In [62]: winner = ipl_df.winner.value_counts()
winner
```

```
Out[62]: Mumbai Indians      109
Chennai Super Kings        100
Kolkata Knight Riders       92
Royal Challengers Bangalore  84
Kings XI Punjab             82
Rajasthan Royals            75
Delhi Daredevils            67
Sunrisers Hyderabad         58
Deccan Chargers             29
Rising Pune Supergiants     15
Gujarat Lions               13
Pune Warriors               12
Delhi Capitals              10
Kochi Tuskers Kerala         6
Name: winner, dtype: int64
```

```
In [74]: winner = winner.reset_index()
```

```
In [75]: winner
```

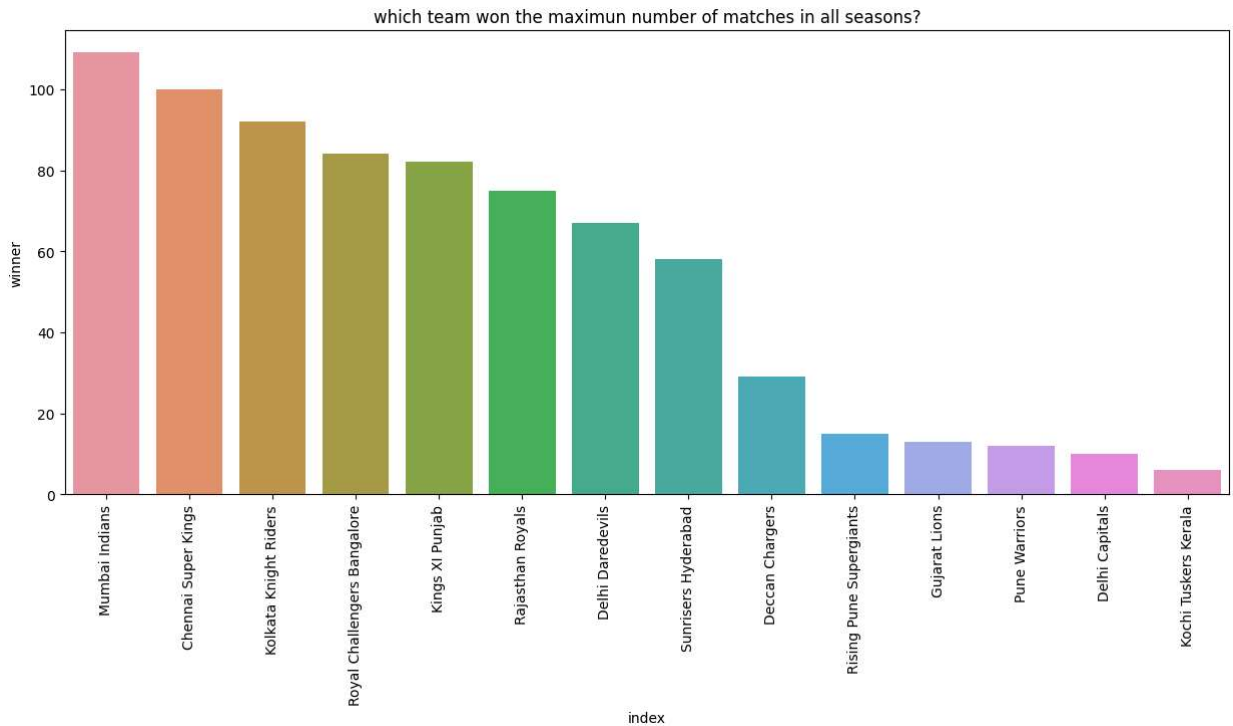
```
Out[75]:
```

	index	winner
0	Mumbai Indians	109
1	Chennai Super Kings	100
2	Kolkata Knight Riders	92
3	Royal Challengers Bangalore	84
4	Kings XI Punjab	82
5	Rajasthan Royals	75
6	Delhi Daredevils	67
7	Sunrisers Hyderabad	58
8	Deccan Chargers	29
9	Rising Pune Supergiants	15
10	Gujarat Lions	13
11	Pune Warriors	12
12	Delhi Capitals	10
13	Kochi Tuskers Kerala	6

```
In [79]: winner.set_index('index',inplace=True)
```

```
In [80]: plt.figure(figsize=(15,6))
plt.xticks(rotation=90)
plt.title("which team won the maximun number of matches in all seasons?")
sns.barplot(winner,x=winner.index,y=winner.winner)
```

```
Out[80]: <Axes: title={'center': 'which team won the maximun number of matches in all seasons?'}, xlabel='index', ylabel='winner'>
```



Q2 which player has become man of the match most number of times?

```
In [88]: m_o_m = ipl_df.player_of_match.value_counts()
m_o_m = m_o_m.reset_index()
m_o_m.set_index('index', inplace=True)
```

In [89]: m_o_m

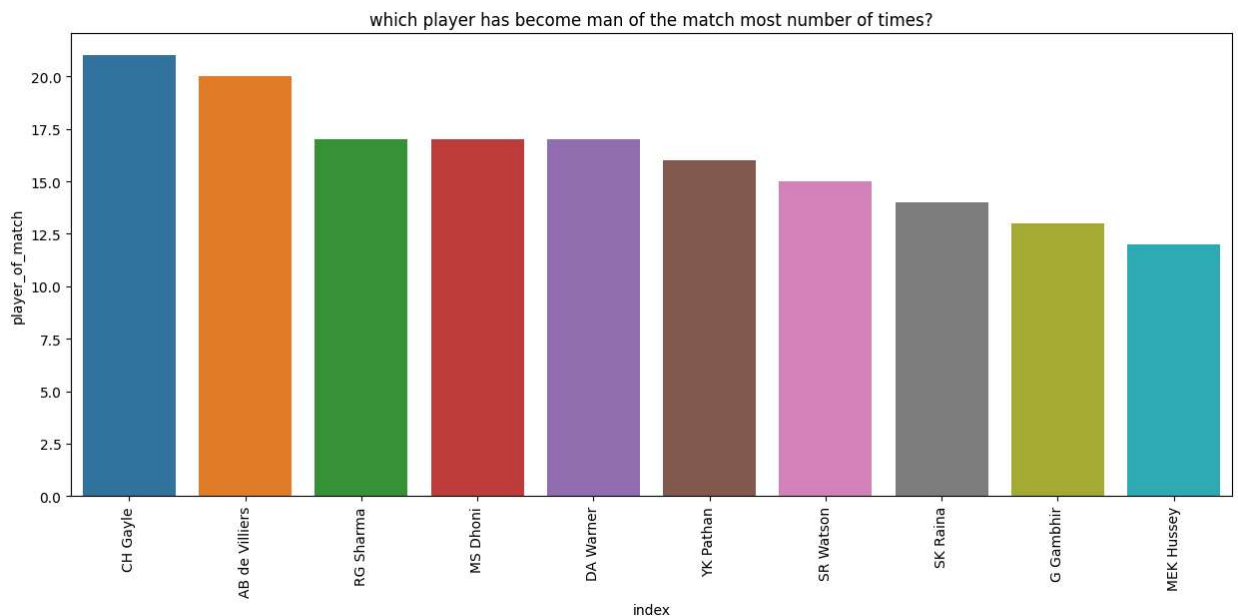
Out[89]:

player_of_match	
index	
CH Gayle	21
AB de Villiers	20
RG Sharma	17
MS Dhoni	17
DA Warner	17
...	...
PD Collingwood	1
NV Ojha	1
AC Voges	1
J Theron	1
S Hetmyer	1

226 rows × 1 columns

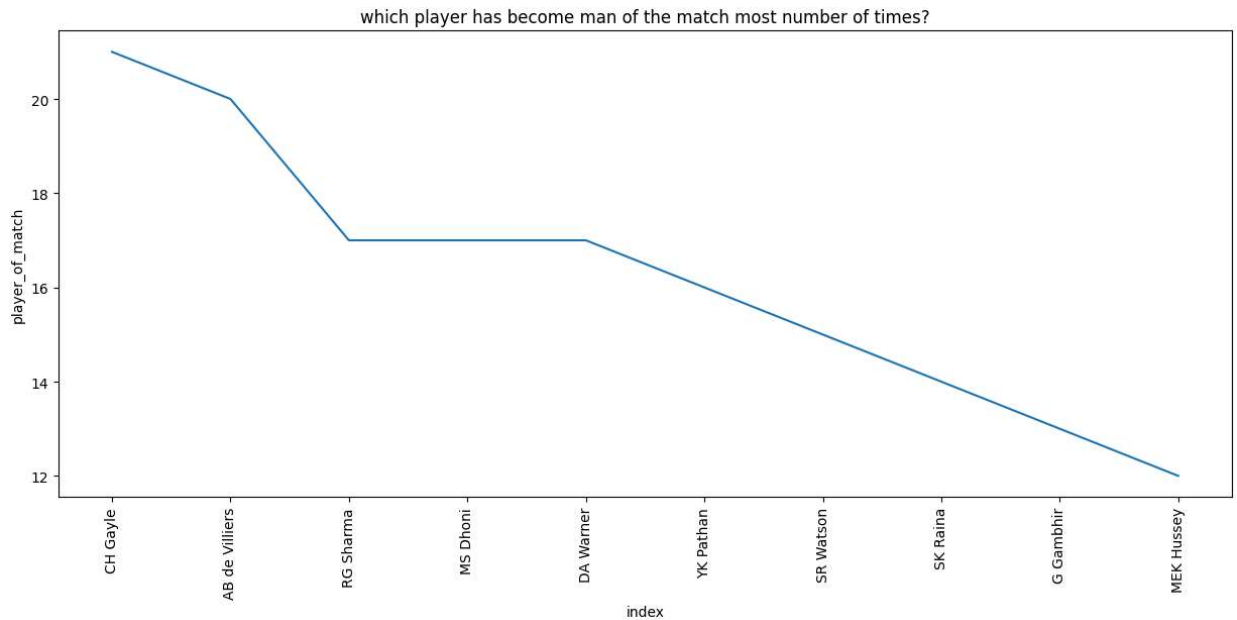
```
In [105]: plt.figure(figsize=(15,6))
plt.xticks(rotation=90)
plt.title("which player has become man of the match most number of times?")
sns.barplot(m_o_m[0:10],x=m_o_m.index[0:10],y=m_o_m.player_of_match[0:10])
```

Out[105]: <Axes: title={'center': 'which player has become man of the match most number of times?'}, xlabel='index', ylabel='player_of_match'>



```
In [106]: plt.figure(figsize=(15,6))
plt.xticks(rotation=90)
plt.title("which player has become man of the match most number of times")
sns.lineplot(m_o_m[0:10],x=m_o_m.index[0:10],y=m_o_m.player_of_match[0:10])
```

```
Out[106]: <Axes: title={'center': 'which player has become man of the match most number of ti
mes?'}, xlabel='index', ylabel='player_of_match'>
```



Q3 Does wining the toss the chances of winning the match?

```
In [98]: ipl_df.winner
```

```
Out[98]: 0      Sunrisers Hyderabad
1      Rising Pune Supergiants
2      Kolkata Knight Riders
3      Kings XI Punjab
4      Royal Challengers Bangalore
...
751     Mumbai Indians
752     Mumbai Indians
753     Delhi Capitals
754     Chennai Super Kings
755     Mumbai Indians
Name: winner, Length: 756, dtype: object
```

```
In [99]: ipl_df.toss_winner
```

```
Out[99]: 0      Royal Challengers Bangalore
1      Rising Pune Supergiants
2      Kolkata Knight Riders
3      Kings XI Punjab
4      Royal Challengers Bangalore
...
751     Mumbai Indians
752     Chennai Super Kings
753     Delhi Capitals
754     Chennai Super Kings
755     Mumbai Indians
Name: toss_winner, Length: 756, dtype: object
```

```
In [102]: win = ipl_df.winner == ipl_df.toss_winner
win_counts = win.value_counts()
win_counts
```

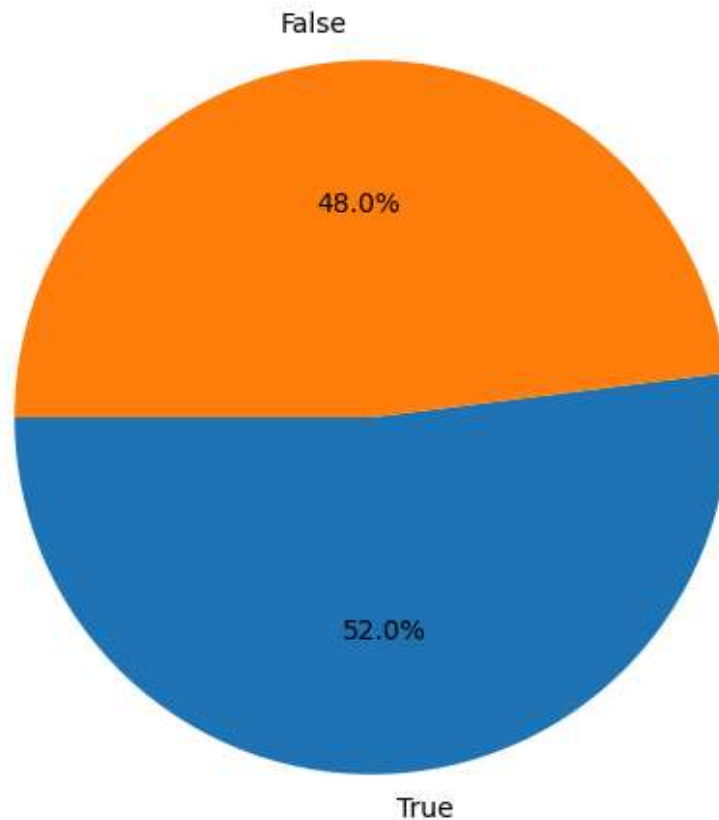
```
Out[102]: True      393
False    363
dtype: int64
```

```
In [104]: round((win_counts/win.count())*100,2)
```

```
Out[104]: True      51.98
False    48.02
dtype: float64
```

```
In [110]: plt.figure(figsize=(15,6))  
plt.xticks(rotation=90)  
plt.title("Does wining the toss the chances of winning the match?")  
plt.pie(win_counts,labels=win_counts.index,autopct='%1.1f%%',startangle=180)  
plt.show()
```

Does wining the toss the chances of winning the match?



```
In [ ]: ## Who are the top 5 scoring batsman?
```