TASK 5 Exploratory Data Analysis - Sports

BY SANA TALYARKHAN TADVI

In [134…
```
#Importing Required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [2]:
```
# UPLOADING DATA

match_data= pd.read_csv('matches.csv')
```

In [3]:
```
D_Data= pd.read_csv('deliveries.csv')
```

In [4]:
```
match_data
```

Out[4]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	r
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	n
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	n
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	n
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	n
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	n
...	...	...	...	...	...	...	...	...	
751	11347	2019	Mumbai	05/05/19	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	n
752	11412	2019	Chennai	07/05/19	Chennai Super Kings	Mumbai Indians	Chennai Super Kings	bat	n
753	11413	2019	Visakhapatnam	08/05/19	Sunrisers Hyderabad	Delhi Capitals	Delhi Capitals	field	n
754	11414	2019	Visakhapatnam	10/05/19	Delhi Capitals	Chennai Super Kings	Chennai Super Kings	field	n

	id	season	city	date	team1	team2	toss_winner	toss_decision	r
755	11415	2019	Hyderabad	12/05/19	Mumbai Indians	Chennai Super Kings	Mumbai Indians	bat	n

756 rows × 18 columns

In [5]:
```
D_Data
```

Out[5]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler
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
...	...	...	...	...	...	...	...	...	...
179073	11415	2	Chennai Super Kings	Mumbai Indians	20	2	RA Jadeja	SR Watson	SL Malinga
179074	11415	2	Chennai Super Kings	Mumbai Indians	20	3	SR Watson	RA Jadeja	SL Malinga
179075	11415	2	Chennai Super Kings	Mumbai Indians	20	4	SR Watson	RA Jadeja	SL Malinga
179076	11415	2	Chennai Super Kings	Mumbai Indians	20	5	SN Thakur	RA Jadeja	SL Malinga
179077	11415	2	Chennai Super Kings	Mumbai Indians	20	6	SN Thakur	RA Jadeja	SL Malinga

179078 rows × 21 columns

In [8]:
```
match_data.shape
```

Out[8]:
```
(756, 19)
```

In [9]:
```
D_Data.shape
```

Out[9]:
```
(179078, 21)
```

In [12]:
```
# KNOW THE DIFFERENT TYPES OF DATA/VARIABLES IN THE DATASET
```

```
match_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 19 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
 18  win_by           756 non-null    object
dtypes: int64(5), object(14)
memory usage: 70.9+ KB
```

In [13]:
```
D_Data.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: 23.2+ MB
```

In [14]:
```
#SUMMARY
match_data.describe()
```

Out[14]:

	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

	id	season	dl_applied	win_by_runs	win_by_wickets
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 [15]:
```
D_Data.describe()
```

Out[15]:

	match_id	inning	over	ball	is_super_over	wide_runs
count	179078.000000	179078.000000	179078.000000	179078.000000	179078.000000	179078.000000
mean	1802.252957	1.482952	10.162488	3.615587	0.000452	0.036721
std	3472.322805	0.502074	5.677684	1.806966	0.021263	0.251161
min	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000
25%	190.000000	1.000000	5.000000	2.000000	0.000000	0.000000
50%	379.000000	1.000000	10.000000	4.000000	0.000000	0.000000
75%	567.000000	2.000000	15.000000	5.000000	0.000000	0.000000
max	11415.000000	5.000000	20.000000	9.000000	1.000000	5.000000

In [22]:
```
#MATCHES WE'VE GOT IN THE DATASET

match_data['id'].value_counts()
```

Out[22]:
```
11311    1
248      1
257      1
256      1
255      1
        ..
503      1
502      1
501      1
500      1
1        1
Name: id, Length: 756, dtype: int64
```

756 IPL Matches is what we've got in our dataset.

In [23]:
```
# NUMBER OF SEASON IN DATASE

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

Out[23]:
```
array([2017, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2018,
       2019], dtype=int64)
```

In [25]:
```
# TEAM WON BY MAXIMUM RUNS

match_data.iloc[match_data['win_by_runs'].idxmax()]
```

Out[25]:
```
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
win_by                   Bat first
Name: 43, dtype: object
```

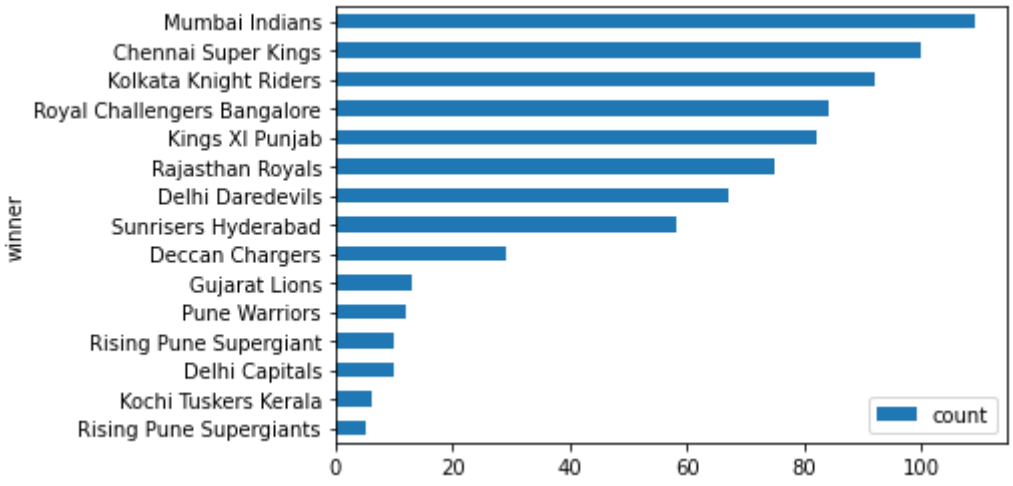
In [26]:
```
# If we're interested only in the winning team

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

Out[26]:
```
'Mumbai Indians'
```

In [52]:
```
match_data.groupby('winner')['winner'].agg(['count']).sort_values('count').reset_ind
```

Out[52]:
```
<AxesSubplot:ylabel='winner'>
```



In [27]:
```
#TEAM WON BY MAX WICKETS
match_data.iloc[match_data['win_by_wickets'].idxmax()]['winner']
```

Out[27]:
```
'Kolkata Knight Riders'
```

In [28]:
```
# TEAM WON BY MINIMUM RUNS

match_data.iloc[match_data[match_data['win_by_runs'].ge(1)].win_by_runs.idxmin()]['w
```

Out[28]:
```
'Mumbai Indians'
```

In [34]:
```
#TEAM WON BY MINIMUM WICKETS

match_data.iloc[match_data[match_data['win_by_wickets'].ge(1)].win_by_wickets.idxmin
```

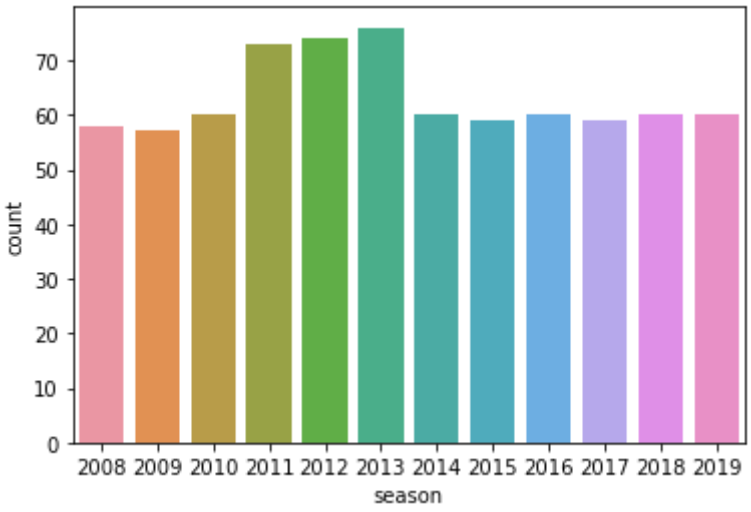
Out[34]:
```
'Kolkata Knight Riders'
```

In [35]:
```
match_data.iloc[match_data[match_data['win_by_wickets'].ge(1)].win_by_wickets.idxmin
```

Out[35]:
```
id                                560
season                           2015
city                          Kolkata
date                       2015-05-09
team1                 Kings XI Punjab
team2             Kolkata Knight Riders
toss_winner           Kings XI Punjab
toss_decision                     bat
result                         normal
dl_applied                          0
winner            Kolkata Knight Riders
win_by_runs                         0
win_by_wickets                      1
player_of_match            AD Russell
venue                     Eden Gardens
umpire1                   AK Chaudhary
umpire2                 HDPK Dharmasena
umpire3                             NaN
win_by                     Bowl first
Name: 559, dtype: object
```

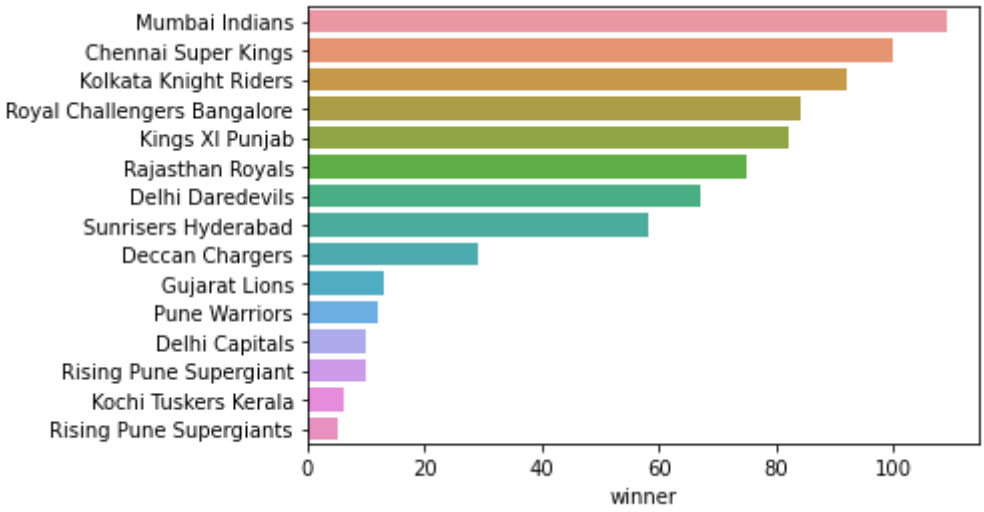
In [37]:
```
# SEASONS WITH MOST NUMBER OF MATCHES

sns.countplot(x='season', data=match_data)
plt.show()
```



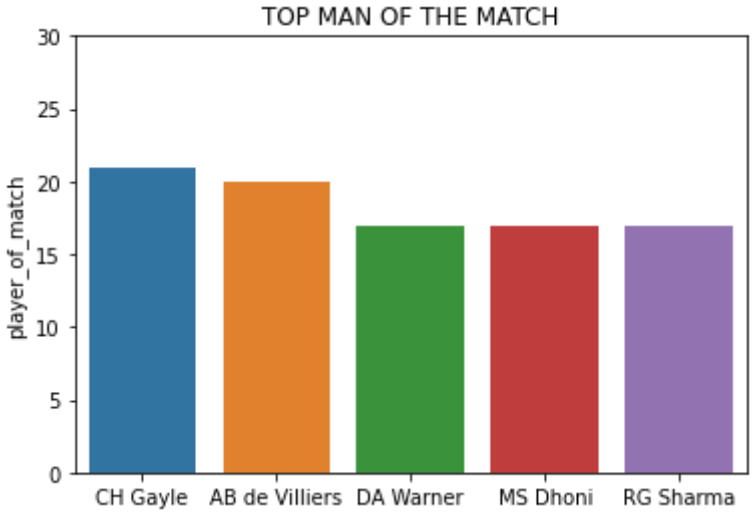
In [39]:
```
#SUCCESSFUL IPL TEAM

data = match_data.winner.value_counts()
sns.barplot(y = data.index, x = data, orient='h');
```



In [49]:
```
#TOP MAN OF THE MATCH

top_players = match_data.player_of_match.value_counts()[:5]
#sns.barplot(x="day", y="total_bill", data=tips)
fig, ax = plt.subplots()
ax.set_ylim([0,30])
ax.set_ylabel("Count")
ax.set_title("TOP MAN OF THE MATCH")
#top_players.plot.bar()
sns.barplot(x = top_players.index, y = top_players, orient='v');
plt.show()
```



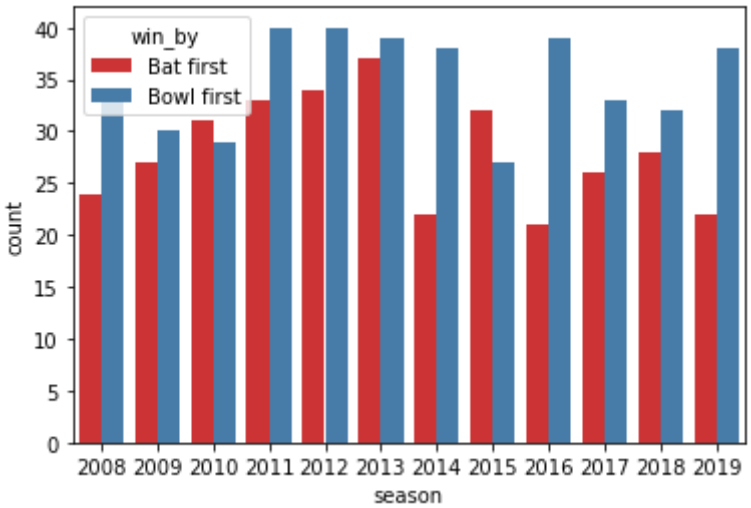
In [ ]:
```

```

In [54]:
```
# BOWL FIRST WIN OR BAT FIRST WIN

sns.countplot('season',hue='win_by',data=match_data,palette="Set1")
```

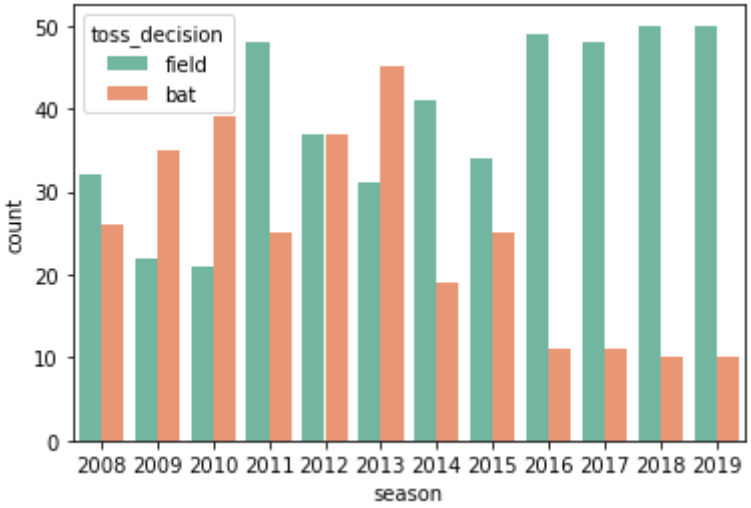
Out[54]:
```
<AxesSubplot:xlabel='season', ylabel='count'>
```



In [55]:
```
# FIELD WIN VS BAT WINS

sns.countplot('season',hue='toss_decision',data=match_data,palette="Set2")
```

Out[55]:
```
<AxesSubplot:xlabel='season', ylabel='count'>
```



FINALS

In [56]:
```
# WINNING TEAMS

final_matches=match_data.drop_duplicates(subset=['season'], keep='last')

final_matches[['season','winner']].reset_index(drop=True).sort_values('season')
```

Out[56]:

	season	winner
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
0	2017	Mumbai Indians
10	2018	Chennai Super Kings
11	2019	Mumbai Indians

In [57]:
```
# IPL FINAL WINNING VENUE AND NUMBER OF TIMES WINNING
final_matches.groupby(['city','winner']).size()
```

Out[57]:
```
city          winner
Bangalore     Kolkata Knight Riders    1
              Sunrisers Hyderabad      1
Chennai       Chennai Super Kings      1
              Kolkata Knight Riders    1
Hyderabad     Mumbai Indians           2
Johannesburg  Deccan Chargers          1
Kolkata       Mumbai Indians           2
Mumbai        Chennai Super Kings      2
              Rajasthan Royals         1
dtype: int64
```

In [58]:
```
final_matches['winner'].value_counts()
```

Out[58]:
```
Mumbai Indians           4
Chennai Super Kings      3
Kolkata Knight Riders    2
Rajasthan Royals         1
Deccan Chargers          1
Sunrisers Hyderabad      1
Name: winner, dtype: int64
```

In [59]:
```
# FINALS MAN OF THE MATCH

final_matches[['winner','player_of_match']].reset_index(drop=True)
```

Out[59]:

	winner	player_of_match
0	Mumbai Indians	KH Pandya
1	Rajasthan Royals	YK Pathan
2	Deccan Chargers	A Kumble
3	Chennai Super Kings	SK Raina
4	Chennai Super Kings	M Vijay
5	Kolkata Knight Riders	MS Bisla
6	Mumbai Indians	KA Pollard
7	Kolkata Knight Riders	MK Pandey
8	Mumbai Indians	RG Sharma
9	Sunrisers Hyderabad	BCJ Cutting
10	Chennai Super Kings	SR Watson
11	Mumbai Indians	JJ Bumrah

PLAYER INFO

In [95]:
```
# USING THE DELIVERIES.CSV DATASET

D_Data
```

Out[95]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler
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

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mills
...	...	...	...	...	...	...	...	...	...
179073	11415	2	Chennai Super Kings	Mumbai Indians	20	2	RA Jadeja	SR Watson	SL Malinga
179074	11415	2	Chennai Super Kings	Mumbai Indians	20	3	SR Watson	RA Jadeja	SL Malinga
179075	11415	2	Chennai Super Kings	Mumbai Indians	20	4	SR Watson	RA Jadeja	SL Malinga
179076	11415	2	Chennai Super Kings	Mumbai Indians	20	5	SN Thakur	RA Jadeja	SL Malinga
179077	11415	2	Chennai Super Kings	Mumbai Indians	20	6	SN Thakur	RA Jadeja	SL Malinga

179078 rows × 21 columns

In [61]:
```
# PLAYER with MAX NUMBER OF TIME MAN OF THE MATCH HOLDER

match_data['player_of_match'].value_counts().head(20)
```

Out[61]:
```
CH Gayle          21
AB de Villiers    20
DA Warner         17
MS Dhoni          17
RG Sharma         17
YK Pathan         16
SR Watson         15
SK Raina          14
G Gambhir         13
MEK Hussey        12
AM Rahane         12
V Kohli           12
AD Russell        11
V Sehwag          11
A Mishra          11
DR Smith          11
JH Kallis         10
KA Pollard        10
SE Marsh           9
AT Rayudu          9
Name: player_of_match, dtype: int64
```

In [104…
```
sixes = D_Data.groupby('batsman')['batsman_runs'].agg(lambda x: (x==6).sum()).reset_
```

In [105…
```
sixes
```

Out[105…

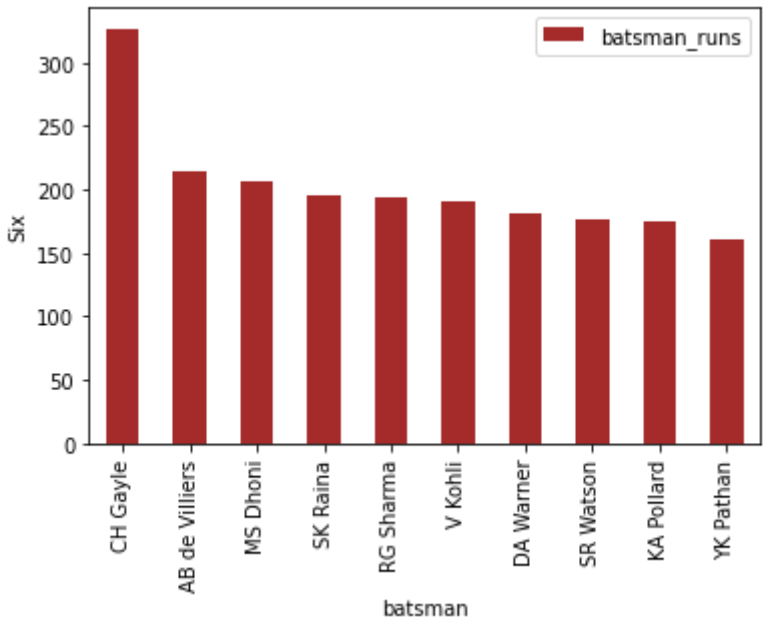
	batsman	batsman_runs
0	CH Gayle	327
1	AB de Villiers	214
2	MS Dhoni	207
3	SK Raina	195

	batsman	batsman_runs
4	RG Sharma	194
...	...	...
511	F Behardien	0
512	DT Patil	0
513	DS Lehmann	0
514	RD Chahar	0
515	M Ntini	0

516 rows × 2 columns

In [123...
```
sixes = D_Data.groupby('batsman')['batsman_runs'].agg(lambda x: (x==6).sum()).reset_
x=sixes.iloc[:10,:].plot('batsman','batsman_runs',kind='bar',color='brown')
plt.ylabel('Six')
```

Out[123... `Text(0, 0.5, 'Six')`



Gayle Storm is at the top of this list with 327 sixes.

In [ ]:
```
```

In [111...
```
fours= D_Data.groupby('batsman')['batsman_runs'].agg(lambda x: (x==4).sum()).reset_i
```

In [125...
```
fours.head(10)
```

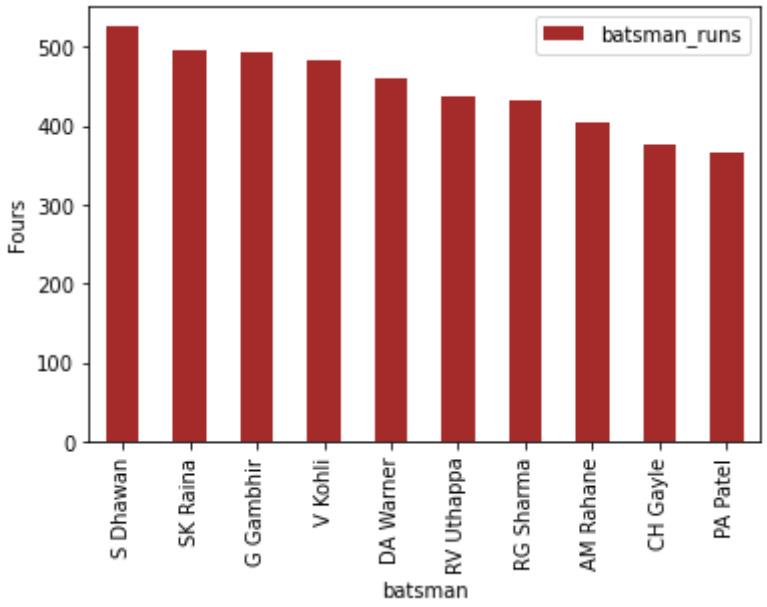
Out[125...

	batsman	batsman_runs
0	S Dhawan	526
1	SK Raina	495
2	G Gambhir	492
3	V Kohli	482
4	DA Warner	459

	batsman	batsman_runs
5	RV Uthappa	436
6	RG Sharma	431
7	AM Rahane	405
8	CH Gayle	376
9	PA Patel	366

In [124...
```
fours= D_Data.groupby('batsman')['batsman_runs'].agg(lambda x: (x==4).sum()).reset_i
x=fours.iloc[:10,:].plot('batsman','batsman_runs',kind='bar',color='brown')
plt.ylabel('Fours')
```

Out[124... `Text(0, 0.5, 'Fours')`



S Dhawan hit maximum fours in IPL

Top 10 Leading Run Scorer in IPL

In [128...
```
batsman_score=D_Data.groupby('batsman')['batsman_runs'].agg(['sum']).reset_index().s
batsman_score=batsman_score.rename(columns={'sum':'batsman_runs'})
print("Top 10 Leading Run Scorer in IPL")
batsman_score.iloc[:10,:]
```

```
Top 10 Leading Run Scorer in IPL
```

Out[128...

	batsman	batsman_runs
0	V Kohli	5434
1	SK Raina	5415
2	RG Sharma	4914
3	DA Warner	4741
4	S Dhawan	4632
5	CH Gayle	4560
6	MS Dhoni	4477

	batsman	batsman_runs
7	RV Uthappa	4446
8	AB de Villiers	4428
9	G Gambhir	4223

IPL's Most Wicket-Taking Bowlers

In [130…
```
wicket_data=D_Data.dropna(subset=['dismissal_kind'])
```

In [132…
```
wicket_data=wicket_data[~wicket_data['dismissal_kind'].isin(['run out','retired hurt
```

In [133…
```
wicket_data.groupby('bowler')['dismissal_kind'].agg(['count']).reset_index().sort_va
```

Out[133…

	bowler	count
0	SL Malinga	170
1	A Mishra	156
2	Harbhajan Singh	150
3	PP Chawla	149
4	DJ Bravo	147
5	B Kumar	133
6	R Ashwin	125
7	SP Narine	122
8	UT Yadav	119
9	RA Jadeja	108

CONCLUSION

1. SUCCESSFUL AND ALL TIME WINNING TEAM IN IPL= MUMBAI INDIANS
2. TEAM WON BY MAX WICKETS= Kolkata Knight Riders'
3. TEAM WON BY MIN RUNS= Mumbai Indians
4. TEAM WON BY MIN WICKETS= Kolkata Knight Riders
5. TOP MAN OF THE MATCH= CH GAYLE followed by AB DE VILLIERS
6. HIGHEST NUMBER OF 4S= S Dhawan hit maximum fours in IPL
7. HIGHEST NUMBER OF 6S= Gayle Storm is at the top of this list with 327 sixes
8. LEADING RUN SCORER= VIRAT KHOLI WITH 5434
9. TOP WICKET TAKING BOWLER= SL MALINGA WITH 170

__ END___

In [ ]: