In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
data = pd.read_csv(r'D:\Career\Udemy\DA\IPL data Analysis/deliveries.csv')
data.head(2)
```

Out[2]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_
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 rows × 21 columns

Indepth Analysis of AB De Villiers

In [3]:

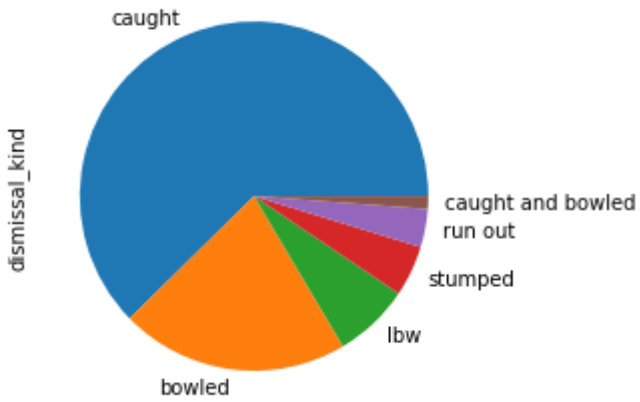
```
filt = data['batsman'] == 'AB de Villiers'
data2 = data[filt]
```

In [4]:

```
data2['dismissal_kind'].value_counts().plot.pie()
```

Out[4]:

```
<AxesSubplot:ylabel='dismissal_kind'>
```



In [5]:

```
len(data2[data2['batsman_runs'] == 4])
```

Out[5]:

```
287
```

In [6]:

```
len(data2[data2['batsman_runs'] == 6])
```

Out[6]:

```
158
```

In [7]:

```
def count(data2,runs):
    return len(data2[data2['batsman_runs'] == runs])*runs
```

In [8]:

```
count(data2,1)
```

Out[8]:

```
980
```

In [9]:

```
count(data2,2)
```

Out[9]:

```
374
```

In [10]:

```
count(data2,3)
```

Out[10]:

```
36
```

In [11]:

```
count(data2,4)
```

Out[11]:

```
1148
```

In [12]:

```
count(data2,6)
```

Out[12]:

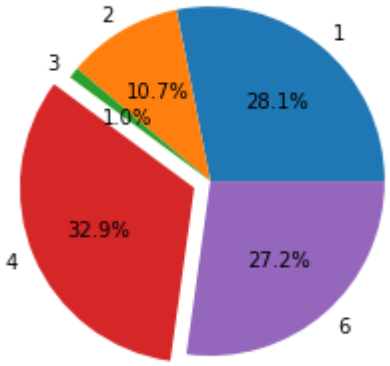
```
948
```

In [13]:

```
slices = [980,374,36,1148,948]
labels = [1,2,3,4,6]
explode = [0,0,0,0.1,0]
plt.pie(slices,labels = labels,autopct = '%1.1f%%',explode = explode )
plt.figure(figsize = (15,15))
```

Out[13]:

```
<Figure size 1080x1080 with 0 Axes>
```




```
<Figure size 1080x1080 with 0 Axes>
```

Score Distribution of Teams

In [14]:

```
data['bowling_team'].unique()
```

Out[14]:

```
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'], dtype=object)
```

In [15]:

```
Teams = {'Royal Challengers Bangalore':'RCB', 'Sunrisers Hyderabad':'SRH',
       'Rising Pune Supergiant':'RPS', 'Mumbai Indians':'MI',
       'Kolkata Knight Riders':'KKR', 'Gujarat Lions':'GL', 'Kings XI Punjab':'KXIP',
       'Delhi Daredevils':'DD', 'Chennai Super Kings':'CSK', 'Rajasthan Royals':'RR',
       'Deccan Chargers':'DC', 'Kochi Tuskers Kerala':'KTK', 'Pune Warriors':'PW',
       'Rising Pune Supergiants':'RPS'}
```

In [16]:

```
data['batting_team'] = data['batting_team'].map(Teams)
data['bowling_team'] = data['bowling_team'].map(Teams)
```

In [17]:

```
runs = data.groupby(['match_id','inning','batting_team'])['total_runs'].sum().reset_index()
runs.drop('match_id',axis = 1,inplace = True)
```

In [18]:

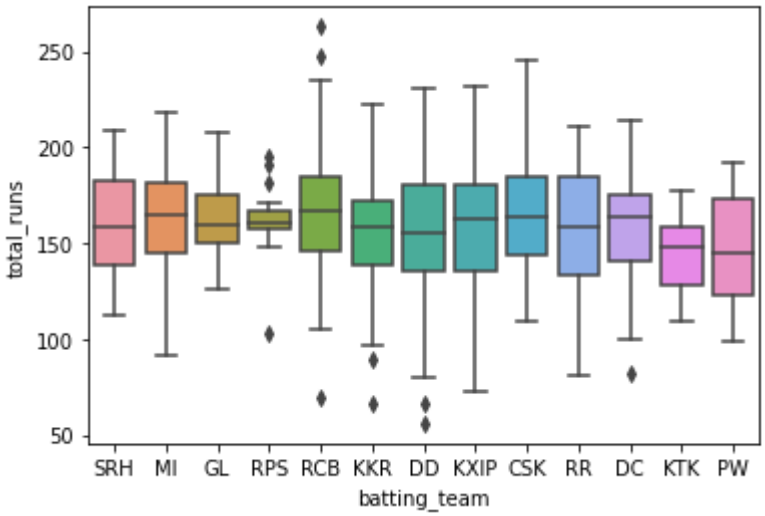
```
inning1 = runs[runs['inning'] == 1]
inning2 = runs[runs['inning'] == 2]
```

In [19]:

```
sns.boxplot(x = 'batting_team',y = 'total_runs',data = inning1)
```

Out[19]:

```
<AxesSubplot:xlabel='batting_team', ylabel='total_runs'>
```

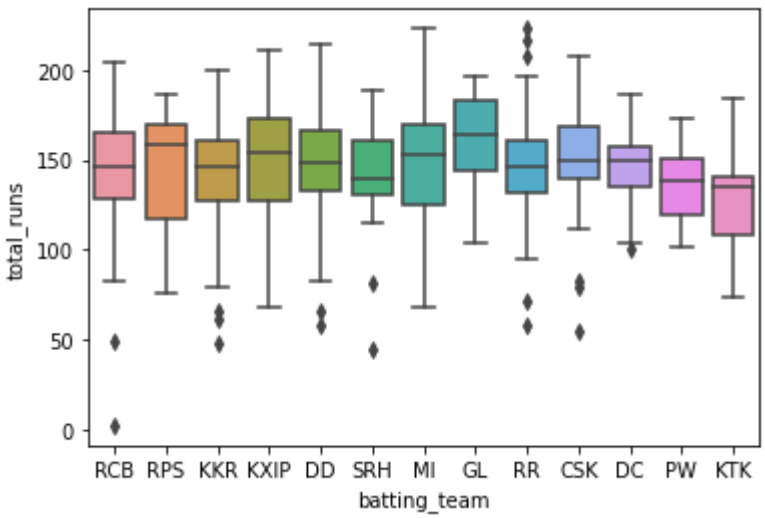


In [20]:

```
sns.boxplot(x = 'batting_team',y = 'total_runs',data = inning2)
```

Out[20]:

```
<AxesSubplot:xlabel='batting_team', ylabel='total_runs'>
```



Teams score more than 200

In [21]:

```
scores = data.groupby(['match_id','inning','batting_team','bowling_team'])['total_runs'].su
```

In [22]:

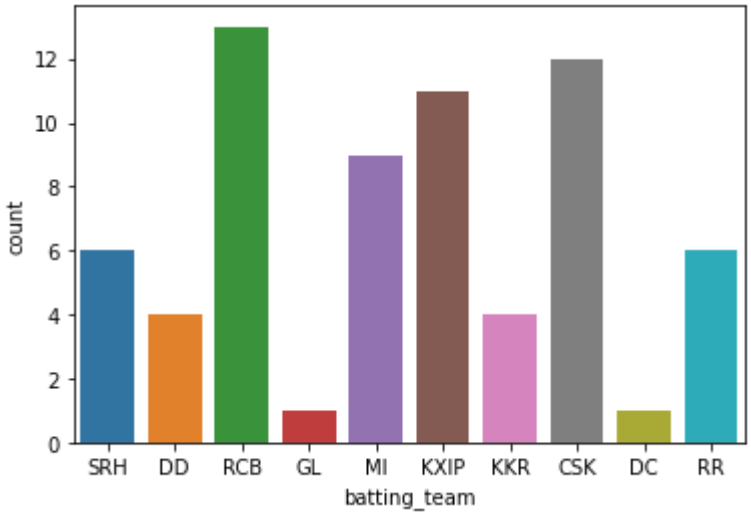
```
s_200 = scores[scores['total_runs'] >= 200]
```

In [23]:

```
sns.countplot(data = s_200,x = 'batting_team')
```

Out[23]:

```
<AxesSubplot:xlabel='batting_team', ylabel='count'>
```

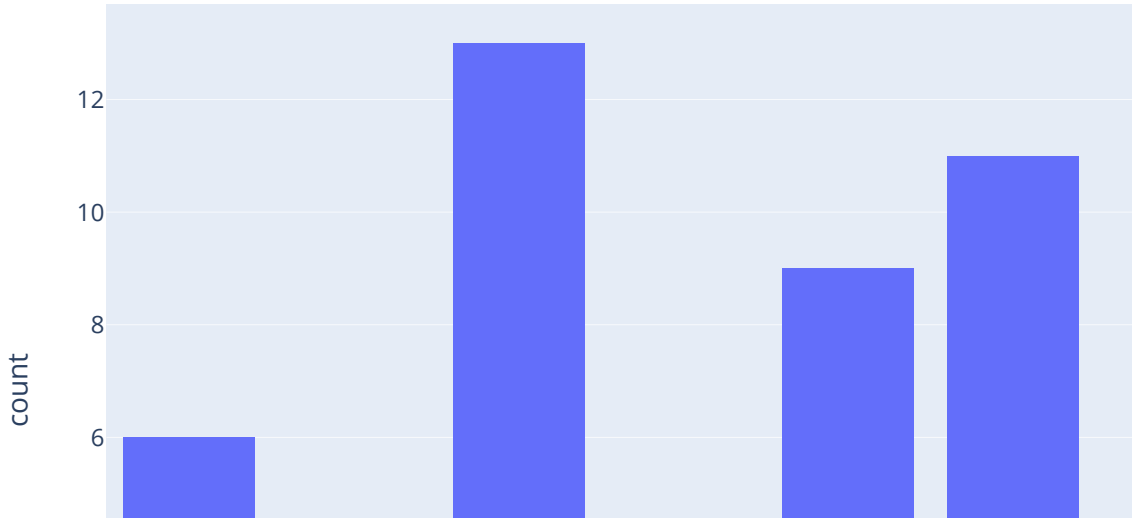


In [24]:

```
import plotly.express as px
```

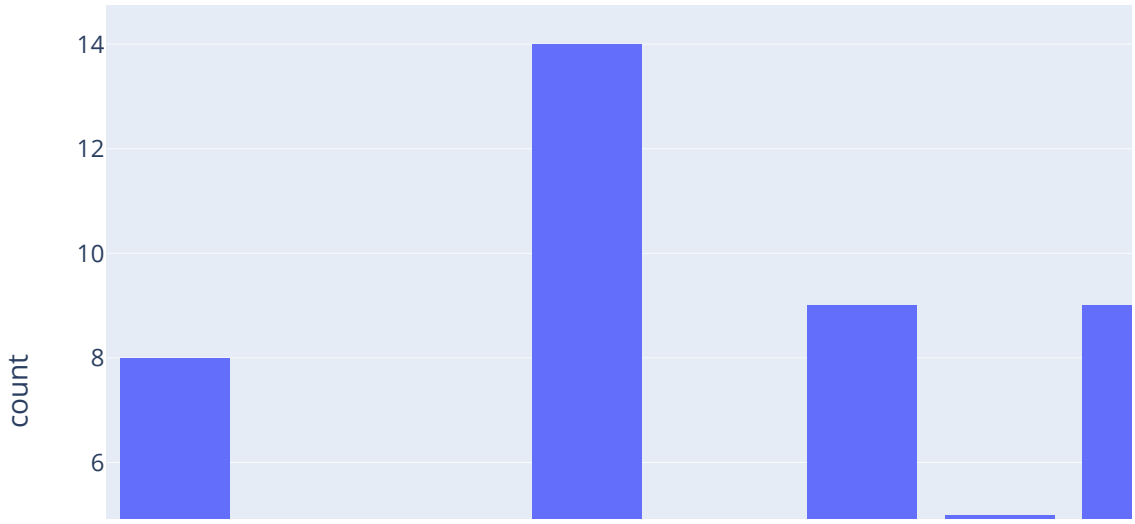
In [25]:

```
px.histogram(s_200,x = 'batting_team')
```



In [26]:

```
px.histogram(s_200,x = 'bowling_team')
```



Top 15 Batsman

In [27]:

```
balls = data.groupby('batsman')['ball'].count().reset_index()
```

In [28]:

```
runs = data.groupby('batsman')['batsman_runs'].sum().reset_index()
```

In [29]:

```
four = data[data['batsman_runs'] == 4]
runs4 = four.groupby('batsman')['batsman_runs'].count().reset_index()
runs4.columns = ['batsman','4s']
```

In [30]:

```
six = data[data['batsman_runs'] == 6]
runs6 = six.groupby('batsman')['batsman_runs'].count().reset_index()
runs6.columns = ['batsman','6s']
```

In [31]:

```
players = pd.merge(pd.merge(pd.merge(runs,balls,on = 'batsman'),runs4,on = 'batsman'),runs6
```

In [32]:

```
players.sort_values(by = 'batsman_runs',ascending = False)[0:15]
```

Out[32]:

	batsman	batsman_runs	ball	4s	6s
237	SK Raina	4548	3369	402	174
272	V Kohli	4423	3494	384	160
208	RG Sharma	4207	3274	354	173
84	G Gambhir	4132	3433	484	58
62	DA Warner	4014	2902	401	160
222	RV Uthappa	3778	2960	377	125
53	CH Gayle	3651	2532	297	266
225	S Dhawan	3561	3005	401	71
168	MS Dhoni	3560	2680	251	156
11	AB de Villiers	3486	2402	287	158
23	AM Rahane	3057	2602	320	60
285	YK Pathan	2922	2076	240	149
122	KD Karthik	2903	2360	286	71
42	BB McCullum	2755	2181	277	124
273	V Sehwag	2728	1833	334	106

In [33]:

```
runs4.sort_values(by = '4s',ascending = False)[0:15]
```

Out[33]:

	batsman	4s
112	G Gambhir	484
303	SK Raina	402
83	DA Warner	401
285	S Dhawan	401
346	V Kohli	384
280	RV Uthappa	377
265	RG Sharma	354
347	V Sehwag	334
34	AM Rahane	320
72	CH Gayle	297
238	PA Patel	297
316	SR Tendulkar	296
19	AB de Villiers	287
161	KD Karthik	286
57	BB McCullum	277

In [34]:

```
runs6.sort_values(by = '6s',ascending = False)[0:15]
```

Out[34]:

	batsman	6s
55	CH Gayle	266
246	SK Raina	174
216	RG Sharma	173
284	V Kohli	160
65	DA Warner	160
11	AB de Villiers	158
175	MS Dhoni	156
297	YK Pathan	149
124	KA Pollard	148
299	Yuvraj Singh	141
230	RV Uthappa	125
44	BB McCullum	124
258	SR Watson	122
80	DR Smith	117
285	V Sehwag	106

In [35]:

```
player = pd.concat([runs,balls.iloc[:,1]],axis = 1)
```

In [36]:

```
top_15 = player.sort_values(by = 'batsman_runs',ascending = False)[0:15]
```

In [37]:

```
top_15['strike_rate'] = top_15['batsman_runs'] / top_15['ball'] * 100
top_15
```

Out[37]:

	batsman	batsman_runs	ball	strike_rate
374	SK Raina	4548	3369	134.995548
431	V Kohli	4423	3494	126.588437
323	RG Sharma	4207	3274	128.497251
137	G Gambhir	4132	3433	120.361200
103	DA Warner	4014	2902	138.318401
340	RV Uthappa	3778	2960	127.635135
85	CH Gayle	3651	2532	144.194313
347	S Dhawan	3561	3005	118.502496
259	MS Dhoni	3560	2680	132.835821
22	AB de Villiers	3486	2402	145.129059
38	AM Rahane	3057	2602	117.486549
454	YK Pathan	2922	2076	140.751445
195	KD Karthik	2903	2360	123.008475
66	BB McCullum	2755	2181	126.318203
433	V Sehwag	2728	1833	148.827059

Highest Score

In [38]:

```
grp = data.groupby(['match_id','batsman','batting_team'])['batsman_runs'].sum().reset_index
```

In [39]:

```
max2 = grp.groupby('batsman')['batsman_runs'].max().reset_index()
max2.columns = ['batsman','max_score']
```

In [40]:

```
max2.sort_values(by = 'max_score',ascending = False)[0:15]
```

Out[40]:

	batsman	max_score
85	CH Gayle	175
66	BB McCullum	158
22	AB de Villiers	133
232	M Vijay	127
103	DA Warner	126
433	V Sehwag	122
291	PC Valthaty	120
11	A Symonds	117
242	MEK Hussey	116
372	SE Marsh	115
446	WP Saha	115
394	ST Jayasuriya	114
249	MK Pandey	114
431	V Kohli	113
123	DPMD Jayawardene	110

Leading Wicket Taker

In [41]:

```
data['dismissal_kind'].unique()
```

Out[41]:

```
array([nan, 'caught', 'bowled', 'run out', 'lbw', 'caught and bowled',
       'stumped', 'retired hurt', 'hit wicket', 'obstructing the field'],
      dtype=object)
```

In [42]:

```
dismissal = ['caught', 'bowled', 'lbw', 'caught and bowled',
       'stumped', 'hit wicket']
```

In [43]:

```
wkt = data[data['dismissal_kind'].isin(dismissal)]
```

In [44]:

```
wkt['bowler'].value_counts()[0:15]
```

Out[44]:

```
SL Malinga         154
A Mishra           134
Harbhajan Singh    127
PP Chawla          126
DJ Bravo           122
B Kumar            111
A Nehra            106
R Vinay Kumar      103
Z Khan             102
R Ashwin           100
SP Narine           95
DW Steyn            92
UT Yadav            91
RP Singh            90
P Kumar             90
Name: bowler, dtype: int64
```

Matches Dataset Analysis

In [45]:

```
data2 = pd.read_csv(r'D:\Career\Udemy\DA\IPL data Analysis/matches.csv')
data2.head(2)
```

Out[45]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result
0	1	2017	Hyderabad	4/5/2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal
1	2	2017	Pune	4/6/2017	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal

In [46]:

```
data2['team1'].unique()
```

Out[46]:

```
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'],
      dtype=object)
```

In [47]:

```
teams = {'Sunrisers Hyderabad':'SRH', 'Mumbai Indians':'MI', 'Gujarat Lions':'GL',
       'Rising Pune Supergiant':'RPS', 'Royal Challengers Bangalore':'RCB',
       'Kolkata Knight Riders':'KKR', 'Delhi Daredevils':'DD', 'Kings XI Punjab':'KXIP',
       'Chennai Super Kings':'CSK', 'Rajasthan Royals':'RR', 'Deccan Chargers':'DC',
       'Kochi Tuskers Kerala':'KTK', 'Pune Warriors':'PW', 'Rising Pune Supergiants':'RPS'}
```

In [48]:

```
data2['team1'] = data2['team1'].map(teams)
data2['team2'] = data2['team2'].map(teams)
data2['toss_winner'] = data2['toss_winner'].map(teams)
data2['winner'] = data2['winner'].map(teams)
```

Total Matches

In [49]:

```
data2.shape[0]
```

Out[49]:

636

Total Venues

In [50]:

```
len(data2['city'].unique())
```

Out[50]:

31

Total Teams

In [51]:

```
len(data2['team1'].unique())
```

Out[51]:

13

Most MOMs

In [52]:

```
data2['player_of_match'].value_counts()[0:15]
```

Out[52]:

```
CH Gayle           18
YK Pathan          16
DA Warner          15
AB de Villiers     15
RG Sharma          14
SK Raina           14
G Gambhir          13
MS Dhoni           13
MEK Hussey         12
AM Rahane          12
V Kohli            11
DR Smith           11
V Sehwag           11
JH Kallis          10
SR Watson          10
Name: player_of_match, dtype: int64
```

Maximum win by runs

In [53]:

```
filt = data2['win_by_runs'].max()
data2[data2['win_by_runs'] == filt]
```

Out[53]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied
43	44	2017	Delhi	5/6/2017	MI	DD	DD	field	normal	0

Maximum win by Wickets

In [54]:

```
filt2 = data2['win_by_wickets'].max()
data2[data2['win_by_wickets'] == filt2]
```

Out[54]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl
2	3	2017	Rajkot	4/7/2017	GL	KKR	KKR	field	normal	
34	35	2017	Chandigarh	4/30/2017	DD	KXIP	KXIP	field	normal	
71	72	2008	Mumbai	4/27/2008	MI	DC	DC	field	normal	
119	120	2009	Cape Town	4/19/2009	KXIP	DD	DD	field	normal	
183	184	2010	Bangalore	3/18/2010	RR	RCB	RCB	field	normal	
298	299	2011	Mumbai	5/20/2011	MI	RR	MI	bat	normal	
376	377	2012	Jaipur	5/20/2012	RR	MI	RR	bat	normal	
390	391	2013	Chandigarh	4/10/2013	KXIP	CSK	CSK	field	normal	
542	543	2015	Delhi	4/26/2015	DD	RCB	RCB	field	normal	
590	591	2016	Rajkot	4/21/2016	GL	SRH	SRH	field	normal	

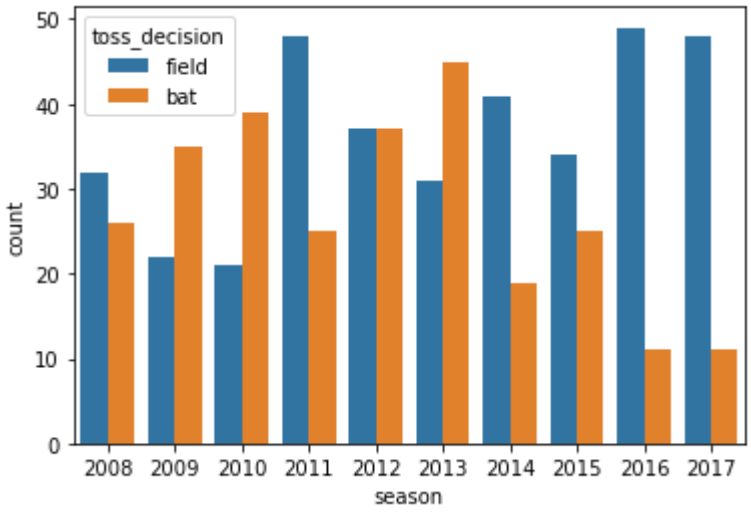
Toss Decisions across seasons

In [55]:

```
sns.countplot(x = 'season',hue = 'toss_decision',data = data2)
```

Out[55]:

```
<AxesSubplot:xlabel='season', ylabel='count'>
```



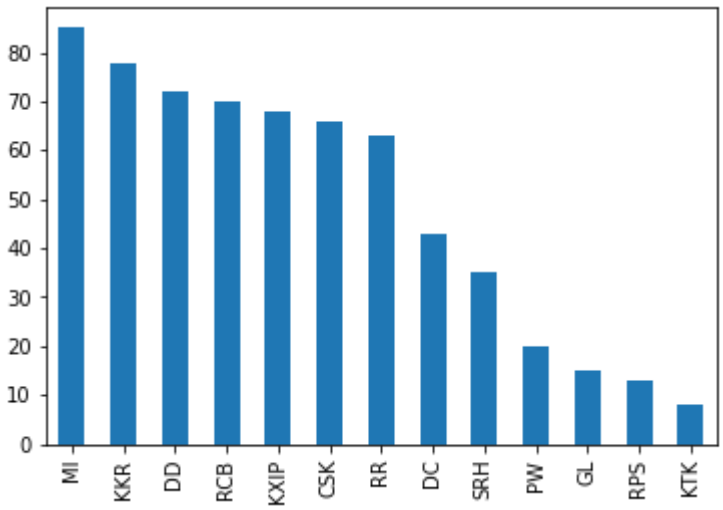
Maximum Toss Winners

In [56]:

```
data2['toss_winner'].value_counts().plot.bar()
```

Out[56]:

```
<AxesSubplot:>
```



Total Matches vs Wins

In [57]:

```
teams = (data2['team1'].value_counts() + data2['team2'].value_counts()).reset_index()
teams.columns = ['Team','Matches']
```

In [58]:

```
wins = data2['winner'].value_counts().reset_index()
wins.columns = ['Team','Wins']
```

In [59]:

```
wins2 = pd.merge(teams,wins,on = 'Team').reset_index()
```

In [60]:

```
wins2['win_%'] = wins2['Wins'] / wins2['Matches'] * 100
```

In [61]:

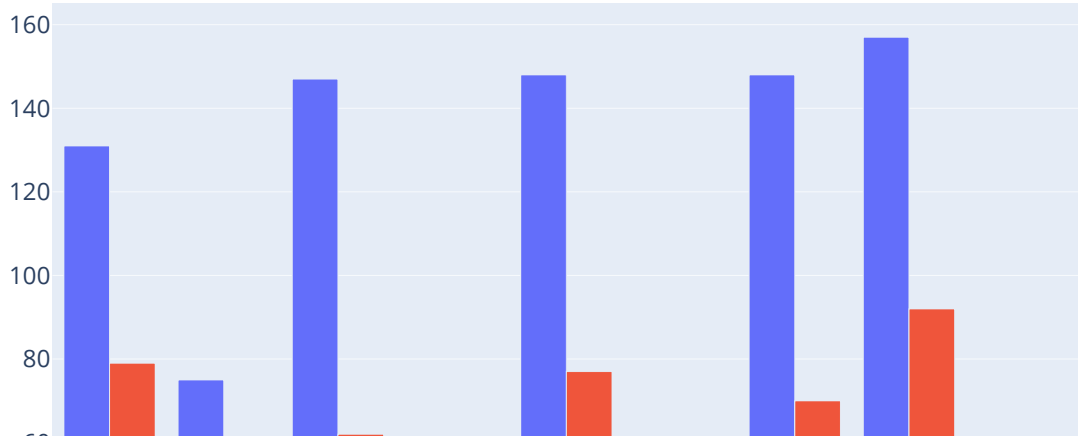
```
import plotly.offline as py
import plotly.graph_objs as go
```

In [62]:

```
trace1 = go.Bar(x = wins2['Team'],y = wins2['Matches'], name = 'Matches_played')
trace2 = go.Bar(x = wins2['Team'],y = wins2['Wins'], name = 'Matches_won')
```

In [63]:

```
data = [trace1,trace2]
py.iplot(data)
```



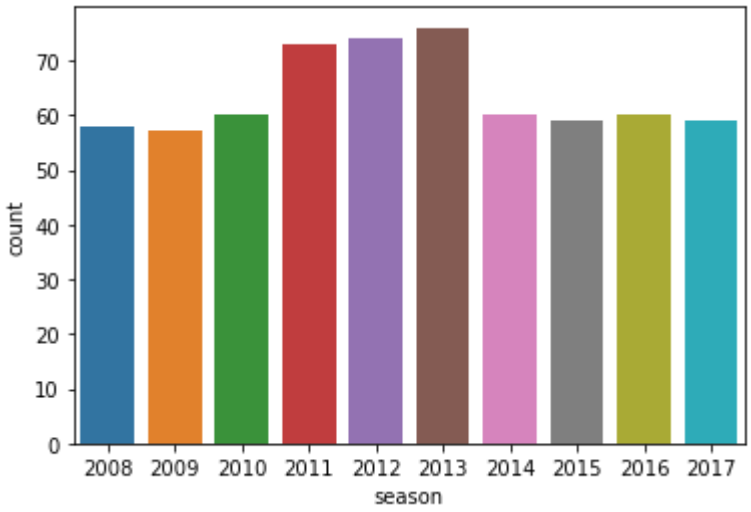
Total Matches in Each season

In [64]:

```
sns.countplot(data = data2,x = 'season')
```

Out[64]:

```
<AxesSubplot:xlabel='season', ylabel='count'>
```



Runs across the seasons

In [65]:

```
data = pd.read_csv(r'D:\Career\Udemy\DA\IPL data Analysis/deliveries.csv')
```

In [66]:

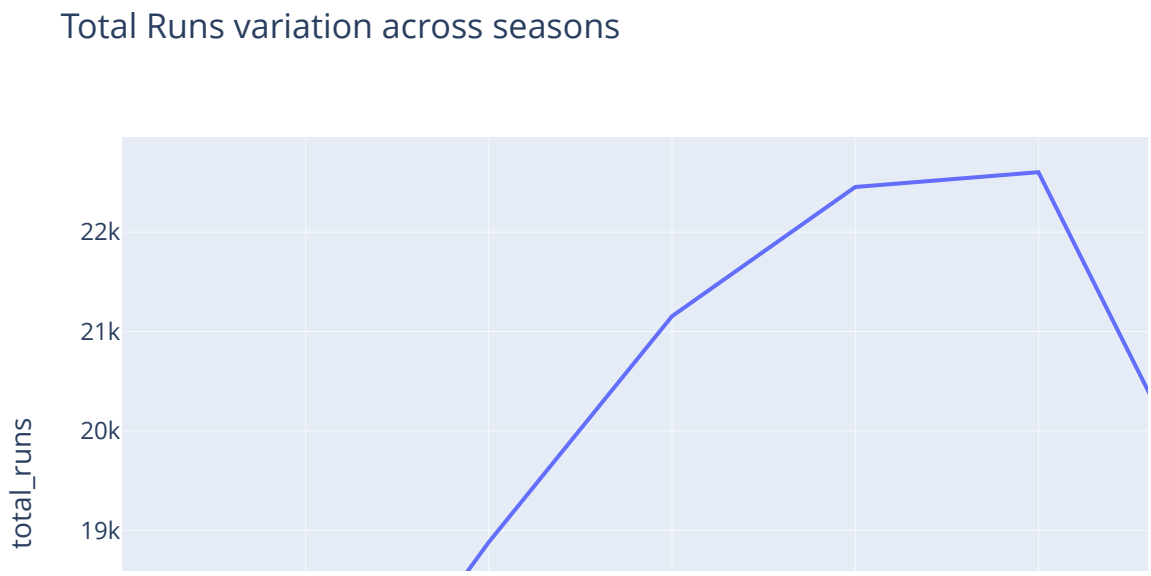
```
season = data2[['id','season']].merge(data,left_on = 'id',right_on = 'match_id',how = 'left
```

In [67]:

```
season = season.groupby('season')['total_runs'].sum().reset_index()
```

In [68]:

```
px.line(season,x = 'season',y = 'total_runs',title = 'Total Runs variation across seasons')
```



Average Runs per Match in Each Season

In [69]:

```
avg_runs = data2.groupby('season')['id'].count().reset_index().rename(columns = {'id':'matc
```

In [70]:

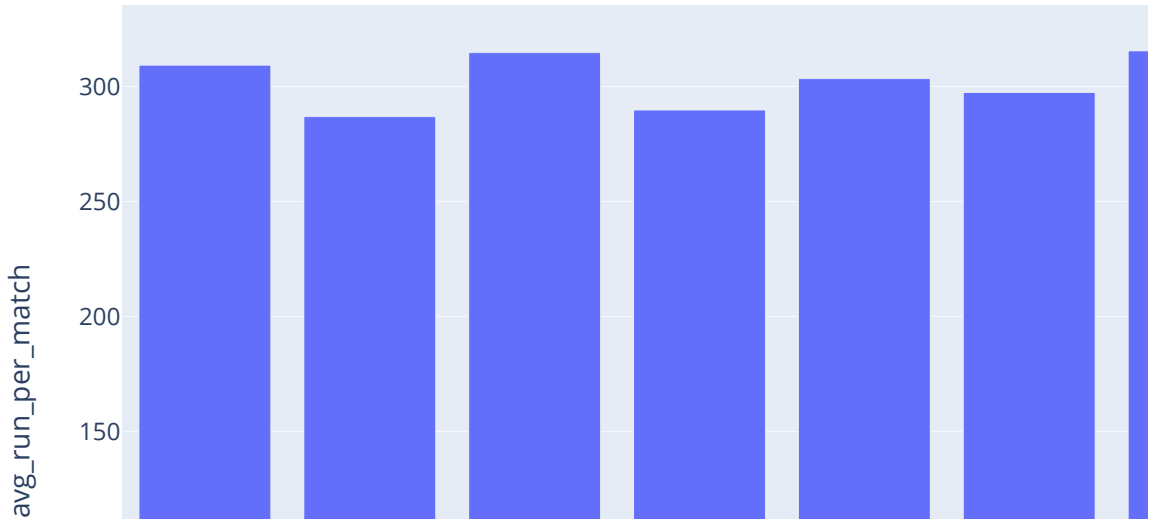
```
season3 = pd.concat([avg_runs,season.iloc[:,1]],axis = 1)
```

In [71]:

```
season3['avg_run_per_match'] = season3['total_runs'] / season3['matches']
```

In [72]:

```
px.bar(season3,x = 'season',y = 'avg_run_per_match')
```

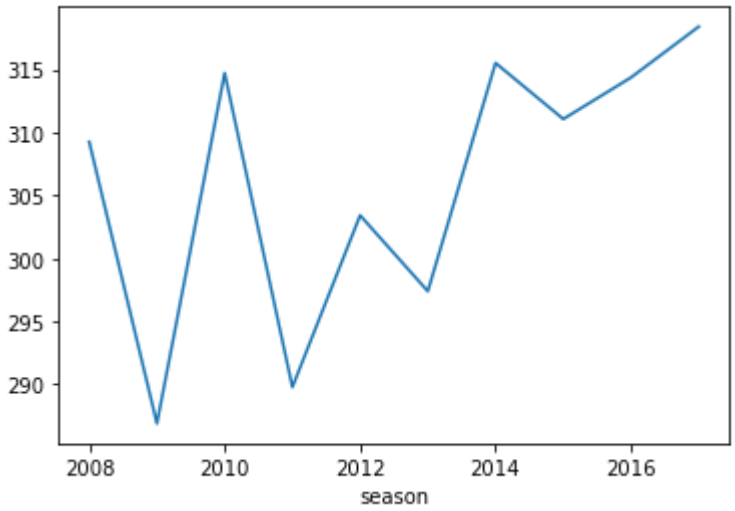


In [73]:

```
season3.set_index('season')['avg_run_per_match'].plot()
```

Out[73]:

```
<AxesSubplot:xlabel='season'>
```



Most Lucky Grounds for Winners

In [74]:

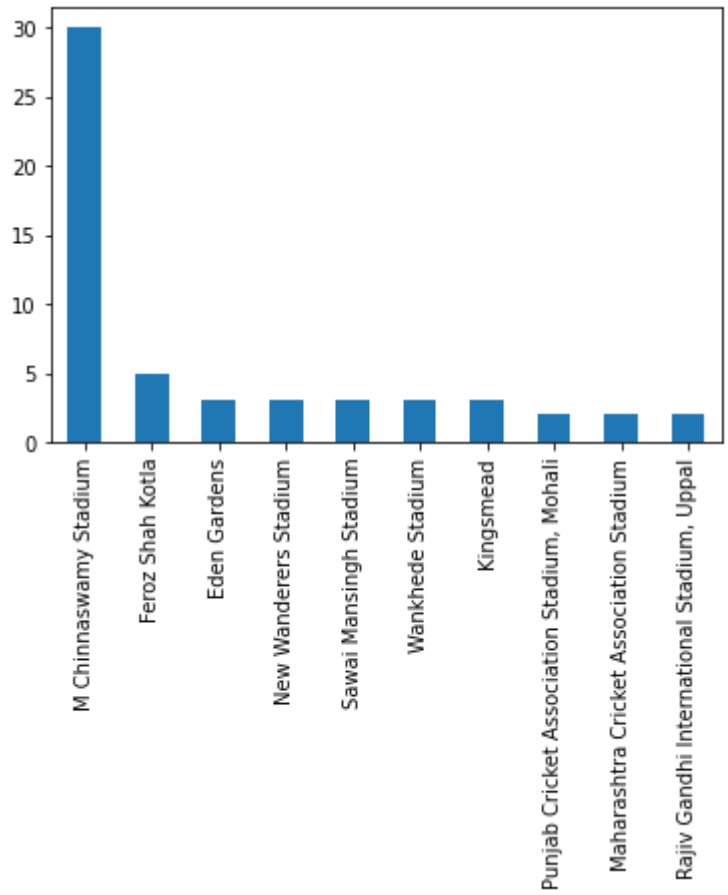
```
def lucky_team(df,team):
    return data2[data2['winner'] == team]['venue'].value_counts().nlargest(10)
```

In [75]:

```
lucky_team(data2,'RCB').plot.bar()
```

Out[75]:

<AxesSubplot:>



Compare 2 teams based on their wins

In [76]:

```
def comparison(team1,team2):
    compare = data2[((data2['team1'] == team1)|(data2['team2'] == team1))&((data2['team1']
    sns.countplot(x = 'season',hue = 'winner',data = compare)
```

In [77]:

```
comparison('RCB','CSK')
```