1.Loding Data

In [ ]:
```
!pip install pandas
!pip install numpy
!pip install matplotlib
!pip install seaborn
```

In [9]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [10]:
```
mat=pd.read_csv("matches.csv")
```

In [11]:
```
mat
```

Out[11]:

	id	season	city	date	match_type	player_of_match	
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinı
1	335983	2007/08	Chandigarh	2008-04-19	League	MEK Hussey	As
2	335984	2007/08	Delhi	2008-04-19	League	MF Maharoof	Fe
3	335985	2007/08	Mumbai	2008-04-20	League	MV Boucher	W
4	335986	2007/08	Kolkata	2008-04-20	League	DJ Hussey	Eden
...	...	...	...	...	...	...	
1090	1426307	2024	Hyderabad	2024-05-19	League	Abhishek Sharma	Raji Inte Upp
1091	1426309	2024	Ahmedabad	2024-05-21	Qualifier 1	MA Starc	Ahı
1092	1426310	2024	Ahmedabad	2024-05-22	Eliminator	R Ashwin	Ahı
1093	1426311	2024	Chennai	2024-05-24	Qualifier 2	Shahbaz Ahmed	Chida
1094	1426312	2024	Chennai	2024-05-26	Final	MA Starc	Chida

1095 rows × 20 columns

In [12]:
```
dev=pd.read_csv("deliveries.csv")
```

In [13]:
```
dev
```

Out[13]:

	match_id	inning	batting_team	bowling_team	over	ball	batter
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum
2	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	3	BB McCullum
3	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	4	BB McCullum
4	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	5	BB McCullum
...	...	...	...	...	...	...	...
260915	1426312	2	Kolkata Knight Riders	Sunrisers Hyderabad	9	5	SS Iyer
260916	1426312	2	Kolkata Knight Riders	Sunrisers Hyderabad	9	6	VR Iyer
260917	1426312	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	1	VR Iyer
260918	1426312	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	2	SS Iyer
260919	1426312	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	3	VR Iyer

260920 rows × 17 columns

2.analyze data set

In [14]:
```
mat.head()
```

Out[14]:

	id	season	city	date	match_type	player_of_match	ven
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnaswa Stadi
1	335983	2007/08	Chandigarh	2008-04-19	League	MEK Hussey	Pun Cric Associat Stadiu Moh
2	335984	2007/08	Delhi	2008-04-19	League	MF Maharoof	Feroz Sh Ko
3	335985	2007/08	Mumbai	2008-04-20	League	MV Boucher	Wankhe Stadi
4	335986	2007/08	Kolkata	2008-04-20	League	DJ Hussey	Ec Garde

In [15]:
```
mat.shape
```

Out[15]: (1095, 20)

In [16]:
```
mat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1095 entries, 0 to 1094
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   id               1095 non-null   int64  
 1   season           1095 non-null   object 
 2   city             1044 non-null   object 
 3   date             1095 non-null   object 
 4   match_type       1095 non-null   object 
 5   player_of_match  1090 non-null   object 
 6   venue            1095 non-null   object 
 7   team1            1095 non-null   object 
 8   team2            1095 non-null   object 
 9   toss_winner      1095 non-null   object 
 10  toss_decision    1095 non-null   object 
 11  winner           1090 non-null   object 
 12  result           1095 non-null   object 
 13  result_margin    1076 non-null   float64
 14  target_runs      1092 non-null   float64
 15  target_overs     1092 non-null   float64
 16  super_over       1095 non-null   object 
 17  method           21 non-null     object 
 18  umpire1          1095 non-null   object 
 19  umpire2          1095 non-null   object 
dtypes: float64(3), int64(1), object(16)
memory usage: 171.2+ KB
```

In [17]:
```
mat.describe()
```

Out[17]:

	id	result_margin	target_runs	target_overs
count	1.095000e+03	1076.000000	1092.000000	1092.000000
mean	9.048283e+05	17.259294	165.684066	19.759341
std	3.677402e+05	21.787444	33.427048	1.581108
min	3.359820e+05	1.000000	43.000000	5.000000
25%	5.483315e+05	6.000000	146.000000	20.000000
50%	9.809610e+05	8.000000	166.000000	20.000000
75%	1.254062e+06	20.000000	187.000000	20.000000
max	1.426312e+06	146.000000	288.000000	20.000000

In [18]:
```
# city with most matches won
mat.groupby(["city"]).agg({"winner":["count"]}).max()
```

Out[18]:
```
winner  count    173
dtype: int64
```

In [19]:
```
mat.groupby(["city"]).agg({"winner":["count"]}).sort_values(ascending=False,
```

Out[19]:

	winner
	count
city	
Mumbai	173

In [20]:
```
# team that won most matches
mat["winner"].value_counts().head(1)
```

Out[20]:
```
winner
Mumbai Indians    144
Name: count, dtype: int64
```

In [21]:
```
# player who won most of man of the match awards
mat["player_of_match"].value_counts().head(1)
```

Out[21]:
```
player_of_match
AB de Villiers    25
Name: count, dtype: int64
```

In [22]:
```
# most frequent umpire 1
mat["umpire1"].value_counts().head(1)
```

Out[22]:
```
umpire1
AK Chaudhary    115
Name: count, dtype: int64
```

In [23]:
```
# most frequent umpire 2

mat["umpire2"].value_counts().head(1)
```

Out[23]:
```
umpire2
S Ravi    83
Name: count, dtype: int64
```

In [24]:
```
mat.describe().T
```

Out[24]:

	count	mean	std	min	25%	5
id	1095.0	904828.319635	367740.242299	335982.0	548331.5	98096
result_margin	1076.0	17.259294	21.787444	1.0	6.0	
target_runs	1092.0	165.684066	33.427048	43.0	146.0	16
target_overs	1092.0	19.759341	1.581108	5.0	20.0	2

In [25]:
```
dev.head()
```

Out[25]:

	match_id	inning	batting_team	bowling_team	over	ball	batter	bowle
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly	P Kuma
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum	P Kuma
2	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	3	BB McCullum	P Kuma
3	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	4	BB McCullum	P Kuma
4	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	5	BB McCullum	P Kuma

In [26]:
```
dev.shape
```

Out[26]:
```
(260920, 17)
```

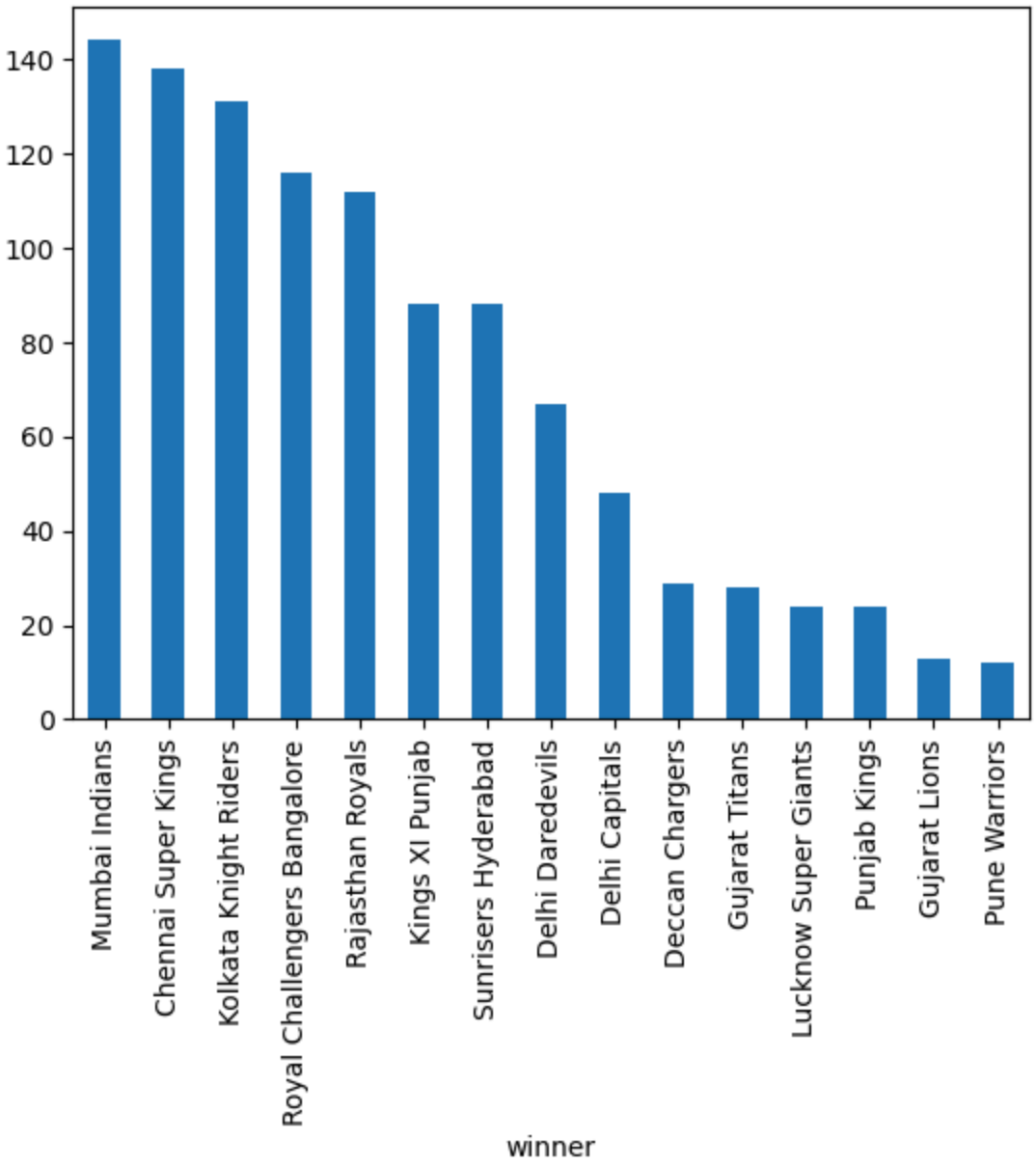
In [27]:
```
dev.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 260920 entries, 0 to 260919
Data columns (total 17 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   match_id          260920 non-null  int64 
 1   inning            260920 non-null  int64 
 2   batting_team      260920 non-null  object
 3   bowling_team      260920 non-null  object
 4   over              260920 non-null  int64 
 5   ball              260920 non-null  int64 
 6   batter            260920 non-null  object
 7   bowler            260920 non-null  object
 8   non_striker       260920 non-null  object
 9   batsman_runs      260920 non-null  int64 
 10  extra_runs        260920 non-null  int64 
 11  total_runs        260920 non-null  int64 
 12  extras_type       14125 non-null   object
 13  is_wicket         260920 non-null  int64 
 14  player_dismissed  12950 non-null   object
 15  dismissal_kind    12950 non-null   object
 16  fielder           9354 non-null    object
dtypes: int64(8), object(9)
memory usage: 33.8+ MB
```

3. visualizing the data

In [28]:
```
mat["winner"].value_counts().head(15).plot(kind="bar")
```

Out[28]:
```
<Axes: xlabel='winner'>
```

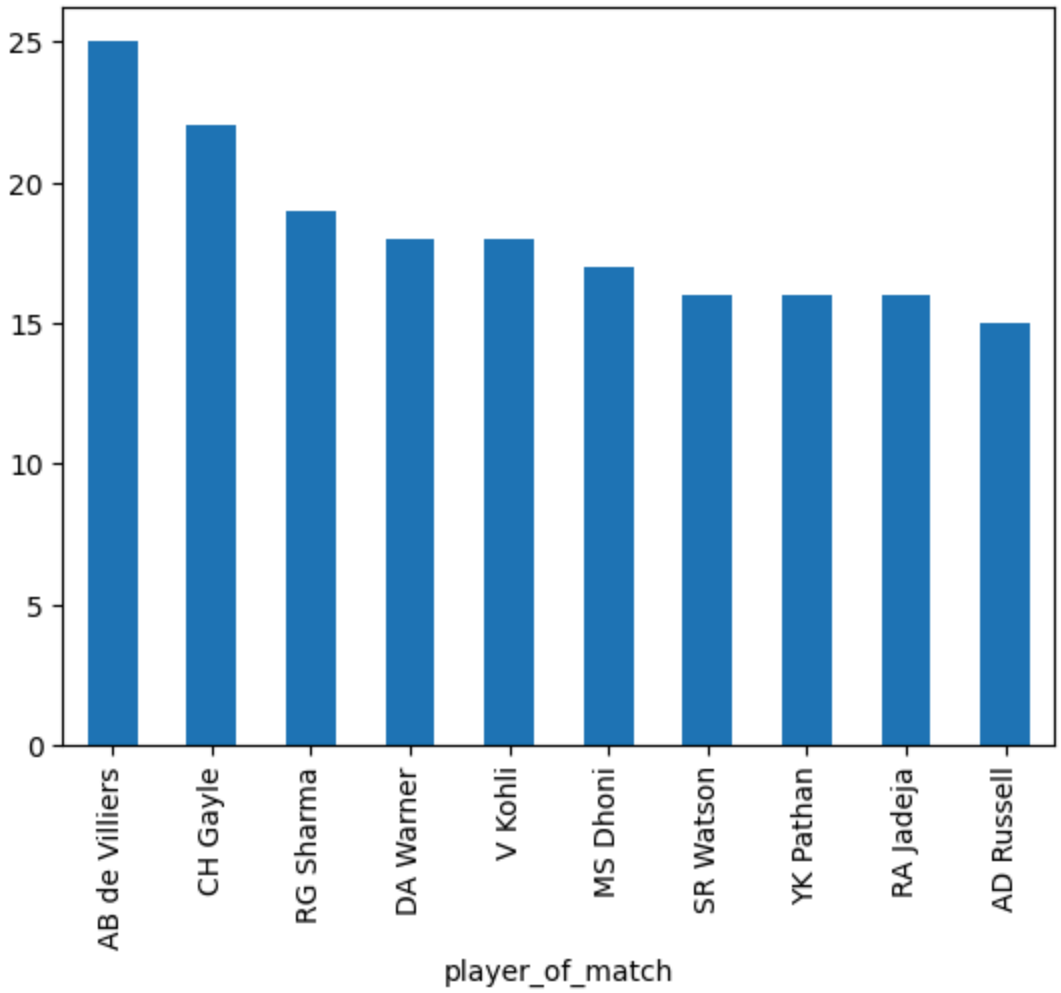


In [29]:
```
mat.winner.unique()
```

Out[29]:
```
array(['Kolkata Knight Riders', 'Chennai Super Kings', 'Delhi Daredevils',
       'Royal Challengers Bangalore', 'Rajasthan Royals',
       'Kings XI Punjab', 'Deccan Chargers', 'Mumbai Indians',
       'Pune Warriors', 'Kochi Tuskers Kerala', nan,
       'Sunrisers Hyderabad', 'Rising Pune Supergiants', 'Gujarat Lions',
       'Rising Pune Supergiant', 'Delhi Capitals', 'Punjab Kings',
       'Gujarat Titans', 'Lucknow Super Giants',
       'Royal Challengers Bengaluru'], dtype=object)
```

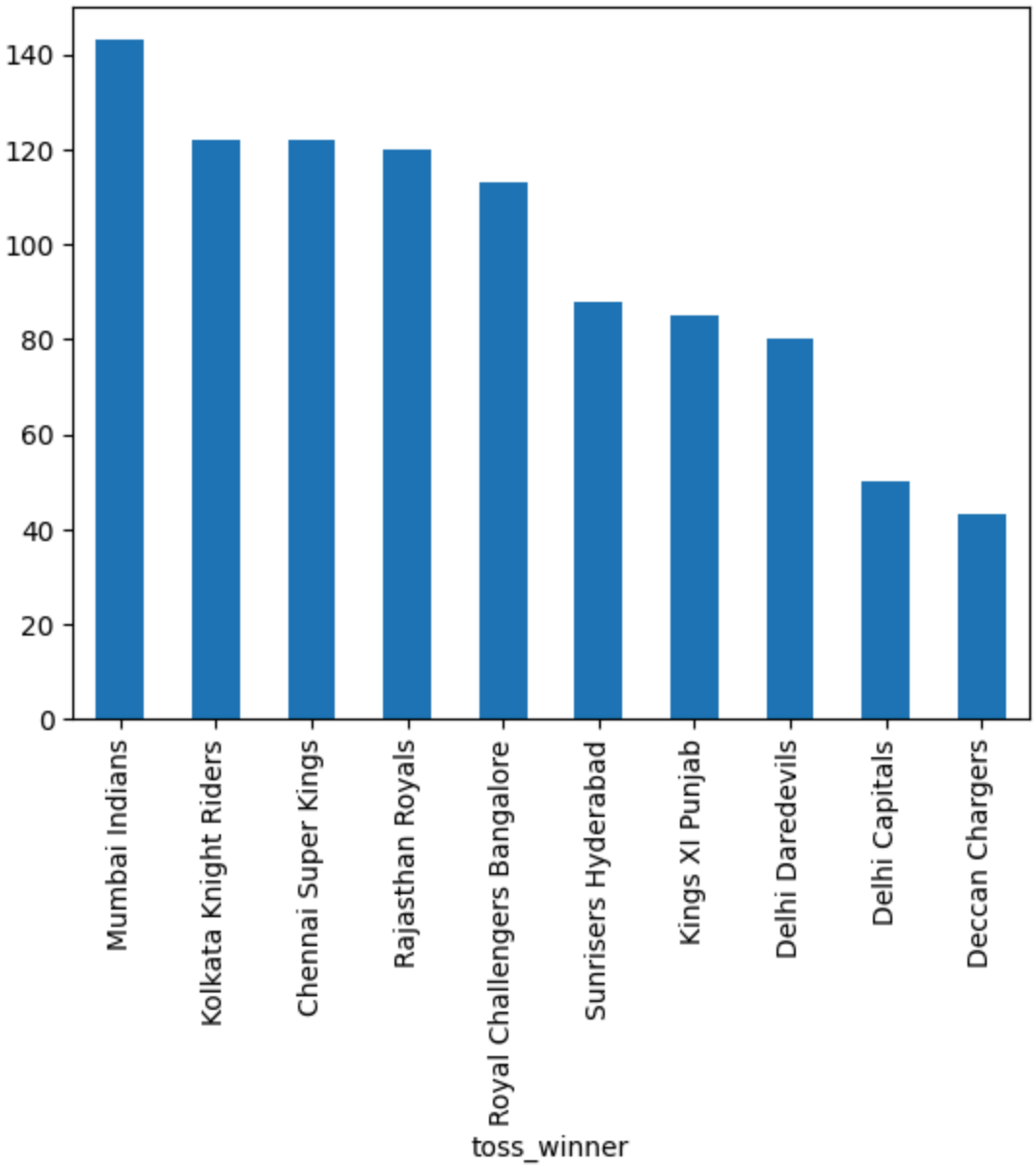
In [30]:
```
mat["player_of_match"].value_counts().head(10).plot(kind="bar")
```

Out[30]:
```
<Axes: xlabel='player_of_match'>
```



In [31]:
```
mat["toss_winner"].value_counts().head(10).plot(kind="bar")
```

Out[31]:
```
<Axes: xlabel='toss_winner'>
```



In [32]:
```
mat.groupby(["toss_winner"]).agg({"winner":["count"]}).max()
```

Out[32]:
```
winner  count    143
dtype: int64
```

In [33]:
```
mat.groupby(["toss_winner"]).agg({"winner":["count"]}).sort_values(ascending
```

Out[33]:

	winner
	count
toss_winner	
Mumbai Indians	143

In [34]:
```
mat=mat.rename(columns={"id":"match_id"})
```

In [35]:
```
mat
```

Out[35]:

	match_id	season	city	date	match_type	player_of_match	
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chi
1	335983	2007/08	Chandigarh	2008-04-19	League	MEK Hussey	A
2	335984	2007/08	Delhi	2008-04-19	League	MF Maharoof	F
3	335985	2007/08	Mumbai	2008-04-20	League	MV Boucher	
4	335986	2007/08	Kolkata	2008-04-20	League	DJ Hussey	Ede
...	...	...	...	...	...	...	
1090	1426307	2024	Hyderabad	2024-05-19	League	Abhishek Sharma	Ra Int Up
1091	1426309	2024	Ahmedabad	2024-05-21	Qualifier 1	MA Starc	Al
1092	1426310	2024	Ahmedabad	2024-05-22	Eliminator	R Ashwin	Al
1093	1426311	2024	Chennai	2024-05-24	Qualifier 2	Shahbaz Ahmed	Chi
1094	1426312	2024	Chennai	2024-05-26	Final	MA Starc	Chi

1095 rows × 20 columns

In [36]:
```
dev.head(250)
```

Out[36]:

	match_id	inning	batting_team	bowling_team	over	ball	batter	b
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly	P I
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum	P I
2	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	3	BB McCullum	P I
3	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	4	BB McCullum	P I
4	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	5	BB McCullum	P I
...	...	...	...	...	...	...	...	
245	335983	1	Chennai Super Kings	Kings XI Punjab	3	1	MEK Hussey	Sree
246	335983	1	Chennai Super Kings	Kings XI Punjab	3	2	MEK Hussey	Sree
247	335983	1	Chennai Super Kings	Kings XI Punjab	3	3	MEK Hussey	Sree
248	335983	1	Chennai Super Kings	Kings XI Punjab	3	4	ML Hayden	Sree
249	335983	1	Chennai Super Kings	Kings XI Punjab	3	5	MEK Hussey	Sree

250 rows × 17 columns

In [37]:
```
df=pd.merge(mat,dev,on="match_id",how="left")
```

In [38]:
```
df.columns
```

Out[38]:
```
Index(['match_id', 'season', 'city', 'date', 'match_type', 'player_of_matc
h',
       'venue', 'team1', 'team2', 'toss_winner', 'toss_decision', 'winner',
       'result', 'result_margin', 'target_runs', 'target_overs', 'super_ove
r',
       'method', 'umpire1', 'umpire2', 'inning', 'batting_team',
       'bowling_team', 'over', 'ball', 'batter', 'bowler', 'non_striker',
       'batsman_runs', 'extra_runs', 'total_runs', 'extras_type', 'is_wicke
t',
       'player_dismissed', 'dismissal_kind', 'fielder'],
      dtype='object')
```

In [39]:
```
df.isnull().sum()
```

Out[39]:
```
match_id               0
season                 0
city               12397
date                   0
match_type             0
player_of_match      490
venue                  0
team1                  0
team2                  0
toss_winner            0
toss_decision          0
winner               490
result                 0
result_margin       4124
target_runs          309
target_overs         309
super_over             0
method            257274
umpire1                0
umpire2                0
inning                 0
batting_team           0
bowling_team           0
over                   0
ball                   0
batter                 0
bowler                 0
non_striker            0
batsman_runs           0
extra_runs             0
total_runs             0
extras_type       246795
is_wicket              0
player_dismissed  247970
dismissal_kind    247970
fielder           251566
dtype: int64
```

In [40]:
```
df.shape
```

Out[40]:
```
(260920, 36)
```

In [41]:
```
df.dropna(how="all",axis=1).head()
```

Out[41]:

	match_id	season	city	date	match_type	player_of_match	ve
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnasw Stad
1	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnasw Stad
2	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnasw Stad
3	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnasw Stad
4	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnasw Stad

5 rows × 36 columns

In [42]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 260920 entries, 0 to 260919
Data columns (total 36 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   match_id          260920 non-null  int64  
 1   season            260920 non-null  object 
 2   city              248523 non-null  object 
 3   date              260920 non-null  object 
 4   match_type        260920 non-null  object 
 5   player_of_match   260430 non-null  object 
 6   venue             260920 non-null  object 
 7   team1             260920 non-null  object 
 8   team2             260920 non-null  object 
 9   toss_winner       260920 non-null  object 
 10  toss_decision     260920 non-null  object 
 11  winner            260430 non-null  object 
 12  result            260920 non-null  object 
 13  result_margin     256796 non-null  float64
 14  target_runs       260611 non-null  float64
 15  target_overs      260611 non-null  float64
 16  super_over        260920 non-null  object 
 17  method            3646 non-null    object 
 18  umpire1           260920 non-null  object 
 19  umpire2           260920 non-null  object 
 20  inning            260920 non-null  int64  
 21  batting_team      260920 non-null  object 
 22  bowling_team      260920 non-null  object 
 23  over              260920 non-null  int64  
 24  ball              260920 non-null  int64  
 25  batter            260920 non-null  object 
 26  bowler            260920 non-null  object 
 27  non_striker       260920 non-null  object 
 28  batsman_runs      260920 non-null  int64  
 29  extra_runs        260920 non-null  int64  
 30  total_runs        260920 non-null  int64  
 31  extras_type       14125 non-null   object 
 32  is_wicket         260920 non-null  int64  
 33  player_dismissed  12950 non-null   object 
 34  dismissal_kind    12950 non-null   object 
 35  fielder           9354 non-null    object 
dtypes: float64(3), int64(8), object(25)
memory usage: 71.7+ MB
```

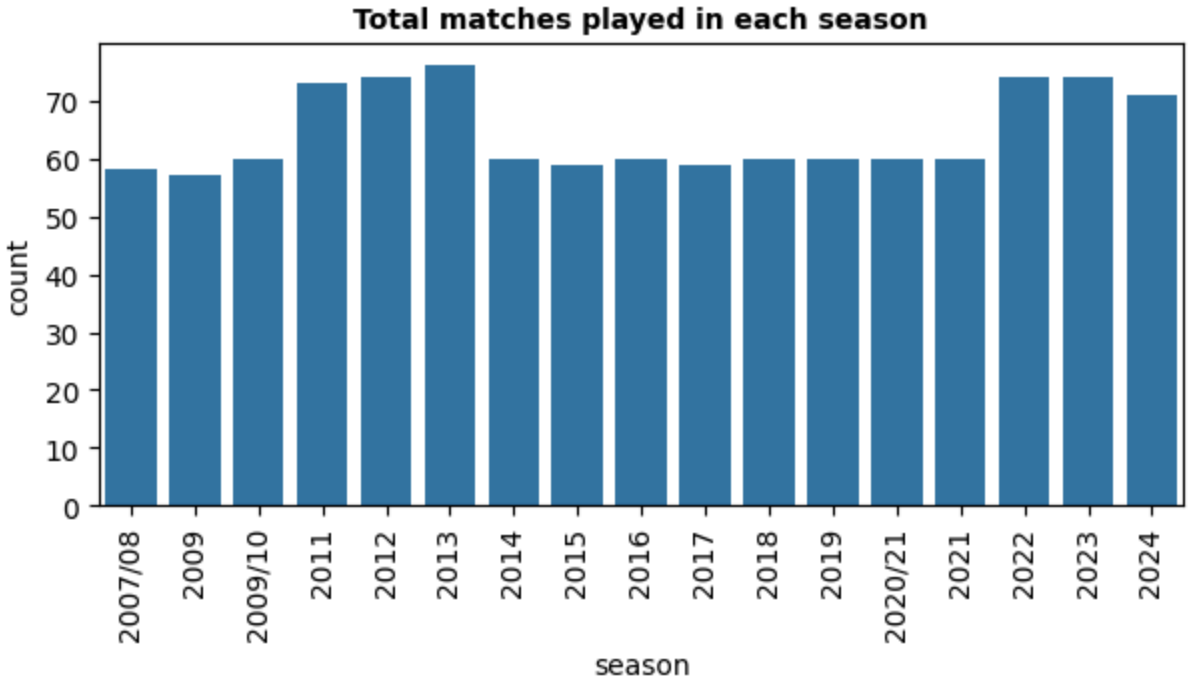
In [43]:
```
mat.groupby(["season"]).agg({"match_id":"count"}).rename(columns={'match_id'
```

Out[43]:

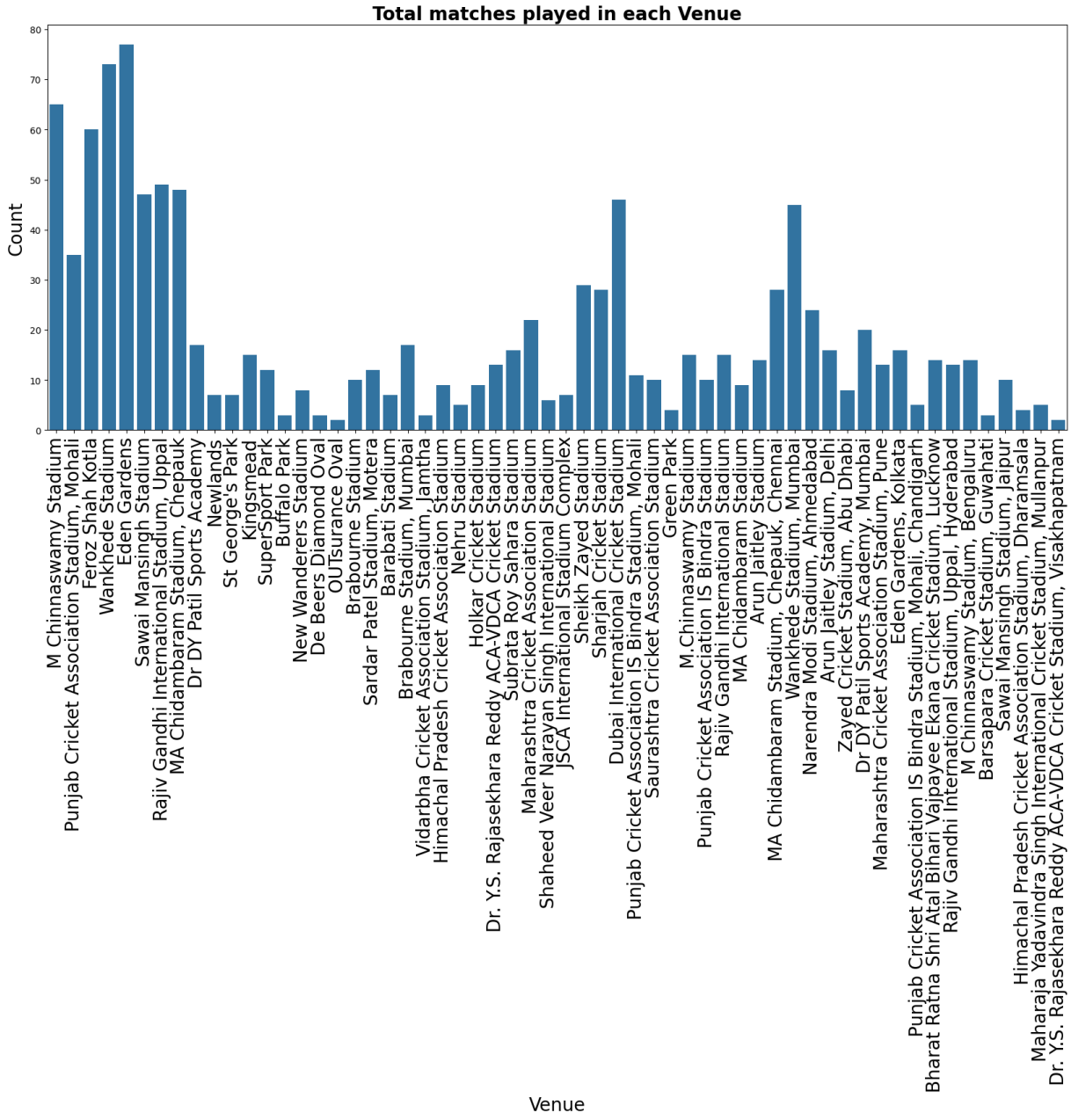
season	no. of matches
2007/08	58
2009	57
2009/10	60
2011	73
2012	74
2013	76
2014	60
2015	59
2016	60
2017	59
2018	60
2019	60
2020/21	60
2021	60
2022	74
2023	74
2024	71

In [44]:

```
plt.subplots(figsize=(7, 3))
sns.countplot(x="season",data=mat)
plt.xticks(rotation=90)
plt.title('Total matches played in each season', fontsize = 10, fontweight =
plt.show()
```



In [45]:
```
#Number of matches playes in each stadium
# mat.venue.value_counts().head(15).plot(kind="bar",figsize=(16,10))

plt.subplots(figsize=(20, 8))
sns.countplot(x="venue",data=mat)
plt.xticks(rotation=90, fontsize=20)
plt.yticks(fontsize=10)
plt.xlabel('Venue', fontsize=20)
plt.ylabel('Count', fontsize=20)
plt.title('Total matches played in each Venue', fontsize = 20, fontweight =
plt.show()
```



In [46]:
```python
def bat_first(x):
    if 'toss_winning_team'=='team1':
        if 'toss_decition'=='bat':
            return 'team1'
        else:
            return 'team2'
    elif 'toss_winning_team'=='team2':
        if 'toss_decition'=='bat':
            return 'team2'
        else:
            return 'team1'
```

In [47]:
```python
dev.head(2)
```

Out[47]:

	match_id	inning	batting_team	bowling_team	over	ball	batter	bowle
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly	I Kuma
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum	I Kuma

In [48]:
```
df.columns
```

Out[48]:
```
Index(['match_id', 'season', 'city', 'date', 'match_type', 'player_of_matc
h',
       'venue', 'team1', 'team2', 'toss_winner', 'toss_decision', 'winner',
       'result', 'result_margin', 'target_runs', 'target_overs', 'super_ove
r',
       'method', 'umpire1', 'umpire2', 'inning', 'batting_team',
       'bowling_team', 'over', 'ball', 'batter', 'bowler', 'non_striker',
       'batsman_runs', 'extra_runs', 'total_runs', 'extras_type', 'is_wicke
t',
       'player_dismissed', 'dismissal_kind', 'fielder'],
      dtype='object')
```

In [49]:
```
filter=["team1","toss_winner"]
```

In [50]:
```
tab=df[filter]
tab.groupby(["team1"]).agg("count")
```

Out[50]:

team1	toss_winner
Chennai Super Kings	31138
Deccan Chargers	9448
Delhi Capitals	10082
Delhi Daredevils	19753
Gujarat Lions	3784
Gujarat Titans	4954
Kings XI Punjab	21848
Kochi Tuskers Kerala	1563
Kolkata Knight Riders	28560
Lucknow Super Giants	5484
Mumbai Indians	29612
Pune Warriors	5483
Punjab Kings	7463
Rajasthan Royals	24167
Rising Pune Supergiant	1617
Rising Pune Supergiants	1677
Royal Challengers Bangalore	31649
Royal Challengers Bengaluru	2171
Sunrisers Hyderabad	20467

In [51]:
```
mat.groupby(["team1"]).agg({"match_id":"count"})
```

Out[51]:

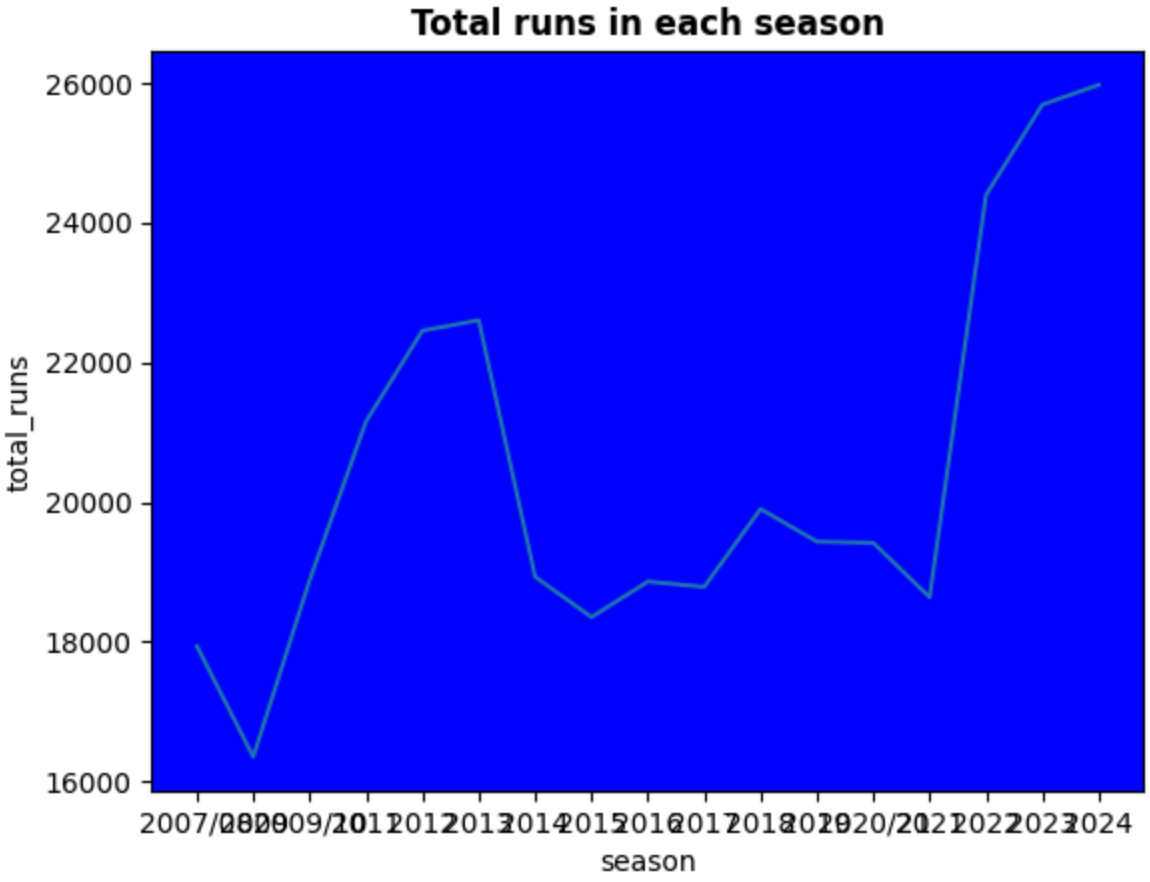
team1	match_id
Chennai Super Kings	128
Deccan Chargers	39
Delhi Capitals	41
Delhi Daredevils	85
Gujarat Lions	16
Gujarat Titans	21
Kings XI Punjab	92
Kochi Tuskers Kerala	7
Kolkata Knight Riders	121
Lucknow Super Giants	23
Mumbai Indians	123
Pune Warriors	23
Punjab Kings	31
Rajasthan Royals	101
Rising Pune Supergiant	7
Rising Pune Supergiants	7
Royal Challengers Bangalore	135
Royal Challengers Bengaluru	9
Sunrisers Hyderabad	86

In [52]:
```
season=df.groupby(['season'])['total_runs'].sum()
season
```

Out[52]:
```
season
2007/08    17937
2009       16353
2009/10    18883
2011       21154
2012       22453
2013       22602
2014       18931
2015       18353
2016       18862
2017       18786
2018       19901
2019       19434
2020/21    19416
2021       18637
2022       24395
2023       25688
2024       25971
Name: total_runs, dtype: int64
```

In [53]:
```
# season=df.groupby(['season'])['total_runs'].sum()
ax = plt.axes()
ax.set(facecolor = "blue")
sns.lineplot(data=season,palette="magma")
plt.title('Total runs in each season',fontsize=12,fontweight="bold")
plt.show()
```

```
C:\Users\91799\AppData\Local\Temp\ipykernel_10384\1493792037.py:4: UserWarni
ng: Ignoring `palette` because no `hue` variable has been assigned.
  sns.lineplot(data=season,palette="magma")
```



In [54]:
```
x=dev.groupby(['batting_team'])['total_runs'].sum().reset_index().sort_value
y=x.reset_index(drop=True,inplace=True)
y
x
```

Out[54]:

	batting_team	total_runs
0	Mumbai Indians	42176
1	Kolkata Knight Riders	39331
2	Chennai Super Kings	38629
3	Royal Challengers Bangalore	37692
4	Rajasthan Royals	34747
5	Kings XI Punjab	30064
6	Sunrisers Hyderabad	29071
7	Delhi Daredevils	24296
8	Delhi Capitals	14900
9	Deccan Chargers	11463
10	Punjab Kings	9536
11	Gujarat Titans	7757
12	Lucknow Super Giants	7510
13	Pune Warriors	6358
14	Gujarat Lions	4862
15	Royal Challengers Bengaluru	2930
16	Rising Pune Supergiant	2470
17	Rising Pune Supergiants	2063
18	Kochi Tuskers Kerala	1901

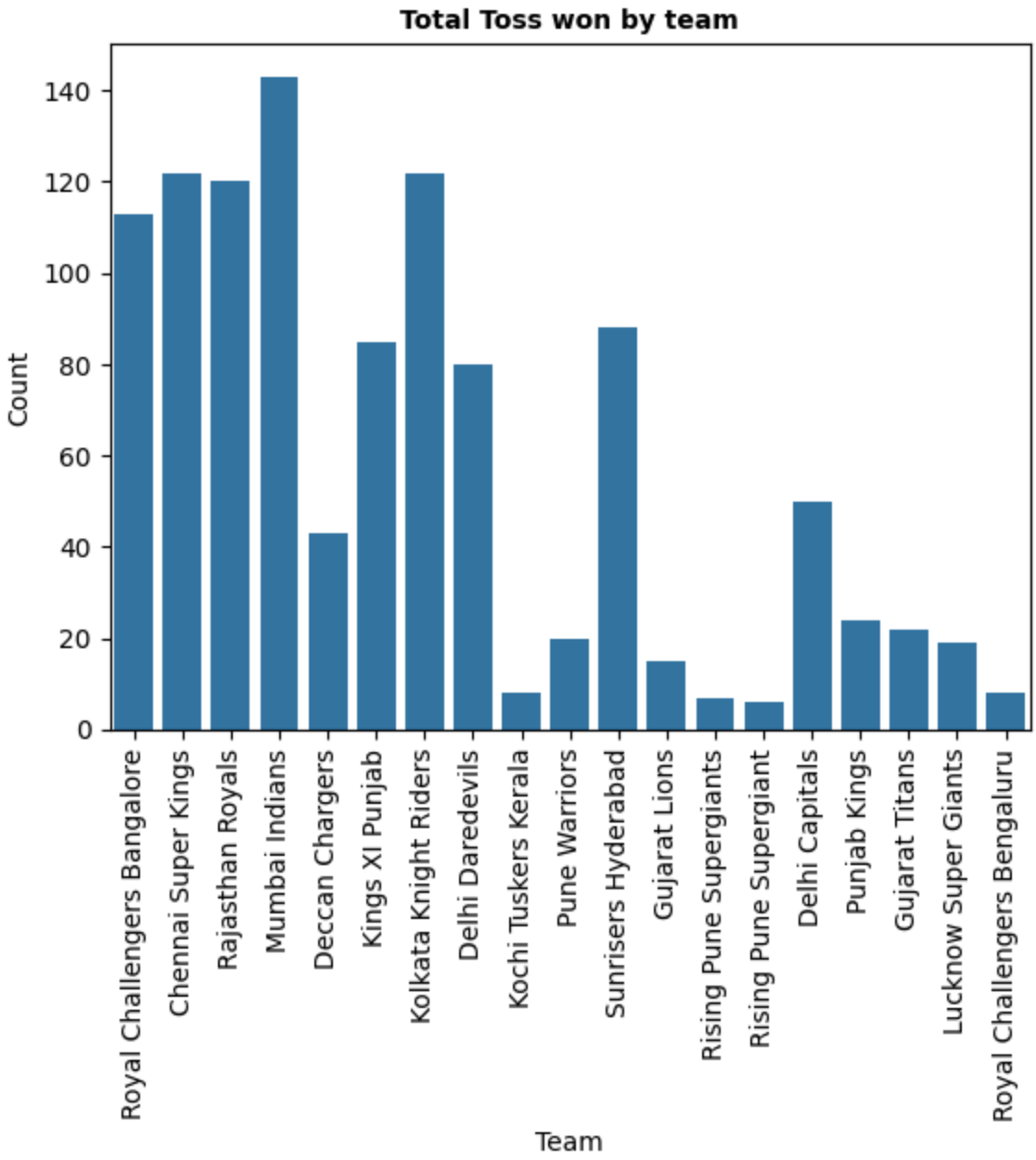
maximum toss won

In [55]:
```
mat["toss_winner"].value_counts().head(10)
```

Out[55]:
```
toss_winner
Mumbai Indians                 143
Kolkata Knight Riders          122
Chennai Super Kings            122
Rajasthan Royals               120
Royal Challengers Bangalore    113
Sunrisers Hyderabad             88
Kings XI Punjab                 85
Delhi Daredevils                80
Delhi Capitals                  50
Deccan Chargers                 43
Name: count, dtype: int64
```

In [56]:
```
sns.countplot(x="toss_winner",data=mat)
plt.xticks(rotation=90, fontsize=10)
```

```
plt.yticks(fontsize=10)
plt.xlabel('Team', fontsize=10)
plt.ylabel('Count', fontsize=10)
plt.title('Total Toss won by team', fontsize = 10, fontweight = "bold")
plt.show()
```

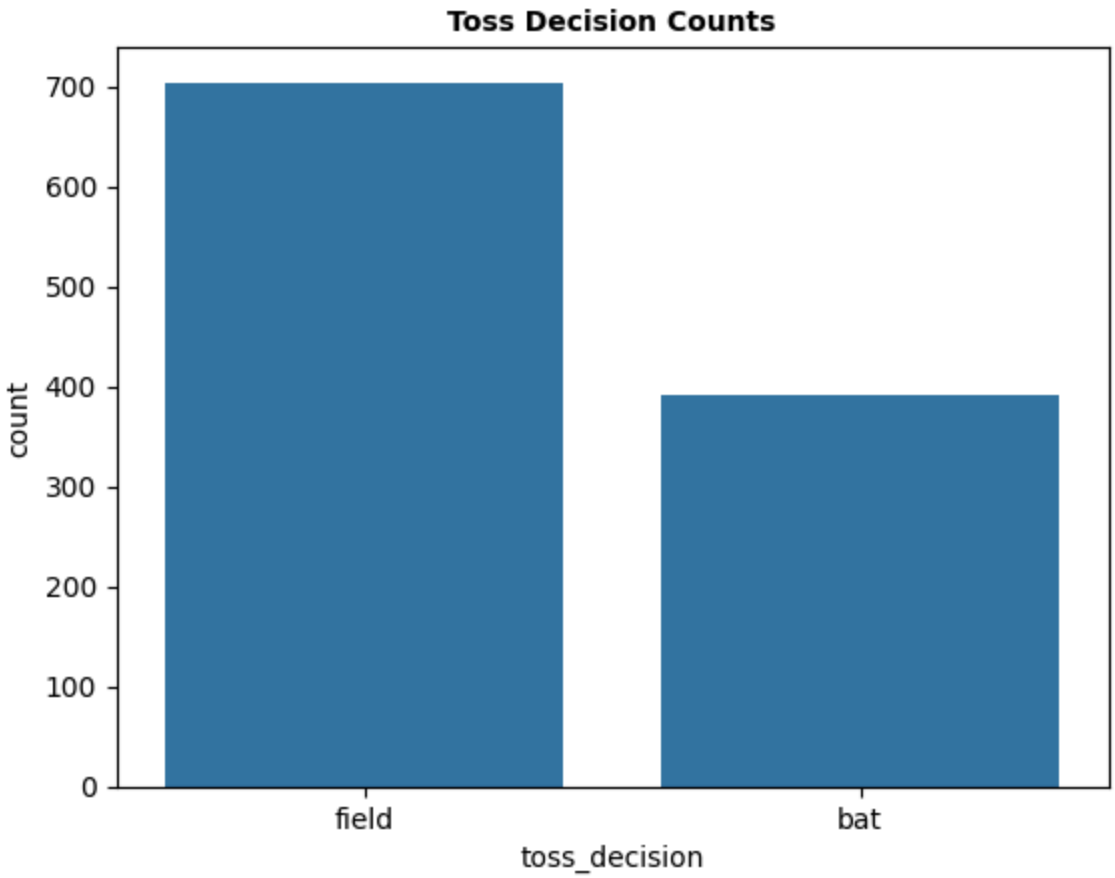


won the toss and win the match

In [57]:
```
k=mat.toss_decision[ mat.toss_winner==mat.winner]
k
```

Out[57]:
```
1         bat
8       field
10      field
12      field
14        bat
        ...
1072    field
1073      bat
1075    field
1078    field
1092    field
Name: toss_decision, Length: 554, dtype: object
```

In [58]:
```
deliveries_df = pd.read_csv("deliveries.csv")
matches_df = pd.read_csv("matches.csv")

# Assuming you want to see the count of toss decisions
sns.countplot(x="toss_decision", data=matches_df)
plt.title("Toss Decision Counts", fontsize=10, fontweight="bold")
plt.show()
```

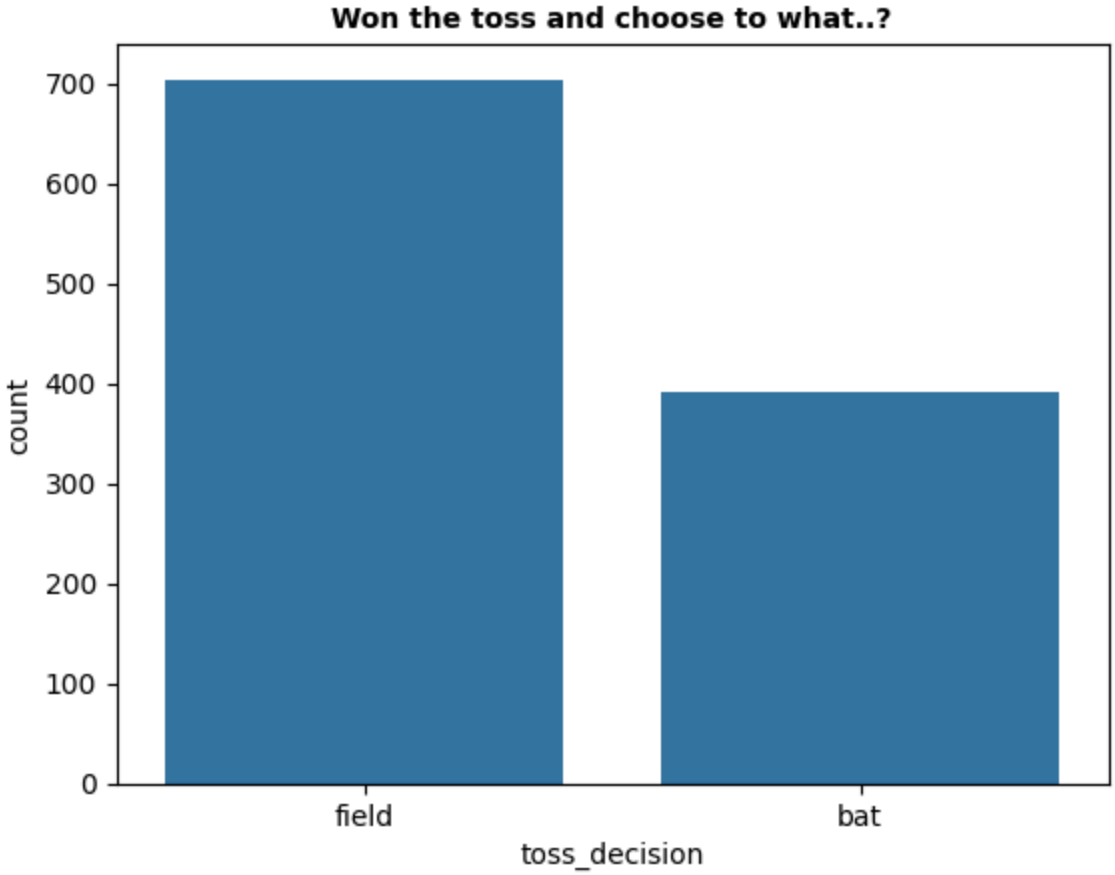


In [59]:
```
mat.head(3)
```

Out[59]:

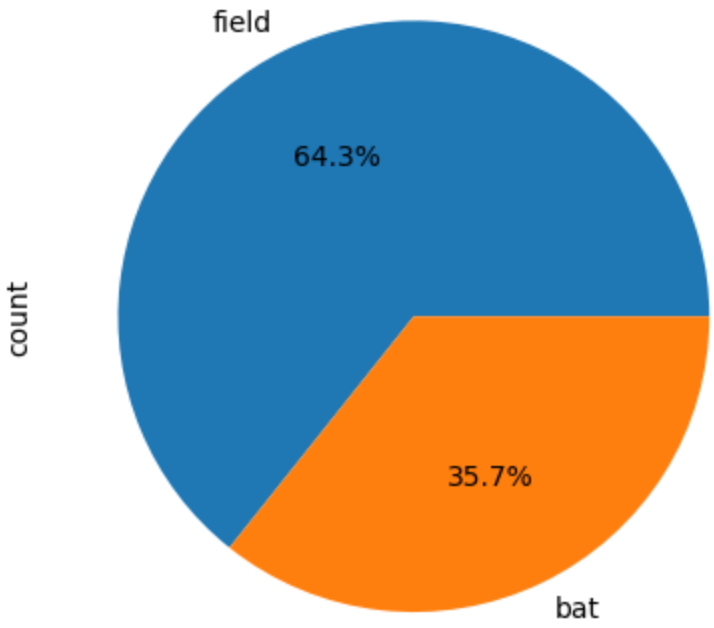
	match_id	season	city	date	match_type	player_of_match	v
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnasv Sta
1	335983	2007/08	Chandigarh	2008-04-19	League	MEK Hussey	P C Assoc Sta M
2	335984	2007/08	Delhi	2008-04-19	League	MF Maharoof	Feroz

In [60]:
```
sns.countplot(x="toss_decision",data=mat)
plt.title("Won the toss and choose to what..?", fontsize = 10, fontweight =
plt.show()
```



In [61]:
```
mat.toss_decision.value_counts().plot(kind="pie", autopct='%1.1f%%')
```

Out[61]:
```
<Axes: ylabel='count'>
```



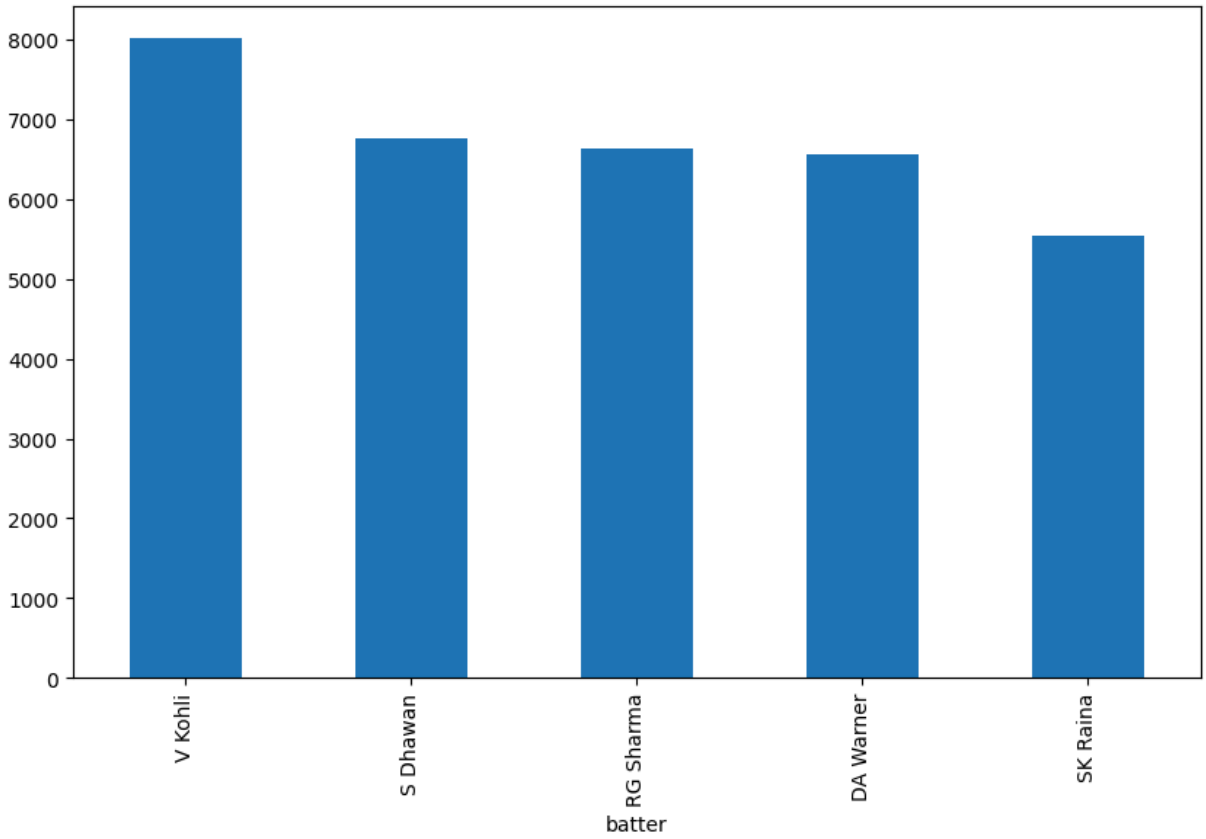
batsman overview

In [62]:
```
dev.columns
```

Out[62]:
```
Index(['match_id', 'inning', 'batting_team', 'bowling_team', 'over', 'bal
l',
       'batter', 'bowler', 'non_striker', 'batsman_runs', 'extra_runs',
       'total_runs', 'extras_type', 'is_wicket', 'player_dismissed',
       'dismissal_kind', 'fielder'],
      dtype='object')
```

In [63]:
```
dev.groupby(["batter"])["batsman_runs"].sum().sort_values(ascending=False).h
```

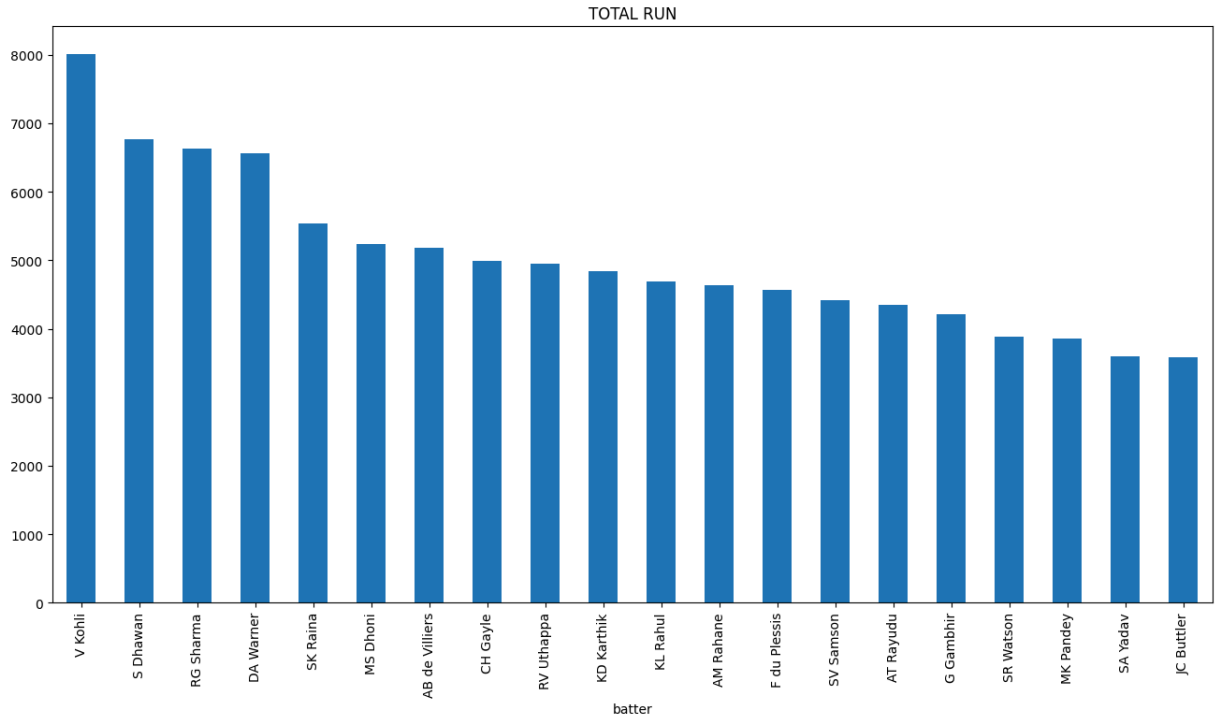
Out[63]:
```
<Axes: xlabel='batter'>
```



In [64]:
```
deliveries_df = pd.read_csv("deliveries.csv")
matches_df = pd.read_csv("matches.csv")

# Define the function to get the total runs scored by a player
def get_total_runs(deliveries_df, player_name):
    # Filter the dataframe for the specific player
    player_deliveries = deliveries_df[deliveries_df['batter'] == player_name

    # Calculate the actual total runs scored by the player
    total_runs = player_deliveries['batsman_runs'].sum()

    return total_runs

# Specify the player name (you can change this to any player's name)
player_name = "S Dhawan"

# Get the total runs scored by the player
total_runs = get_total_runs(deliveries_df, player_name)

# Print the result
print(f"Total runs scored by {player_name}: {total_runs}")
```

```
Total runs scored by S Dhawan: 6769
```

In [65]:
```
dev.groupby(["batter"])["batsman_runs"].sum().sort_values(ascending=False).h
plt.title("TOTAL RUN")
plt.show()
```



In [66]:
```
strike_rate=dev.groupby(["batter"]).agg({"ball":"count","batsman_runs":"sum"
```

In [67]:
```
strike_rate["strike_rate"]=strike_rate.batsman_runs/strike_rate.ball*100
```

In [68]:
```
strike_rate.head(10)
```

Out[68]:

batter	ball	batsman_runs	strike_rate
V Kohli	6236	8014	128.511867
S Dhawan	5483	6769	123.454313
RG Sharma	5183	6630	127.918194
DA Warner	4849	6567	135.429986
SK Raina	4177	5536	132.535312
MS Dhoni	3947	5243	132.835065
AB de Villiers	3487	5181	148.580442
CH Gayle	3516	4997	142.121729
RV Uthappa	3927	4954	126.152279
KD Karthik	3687	4843	131.353404

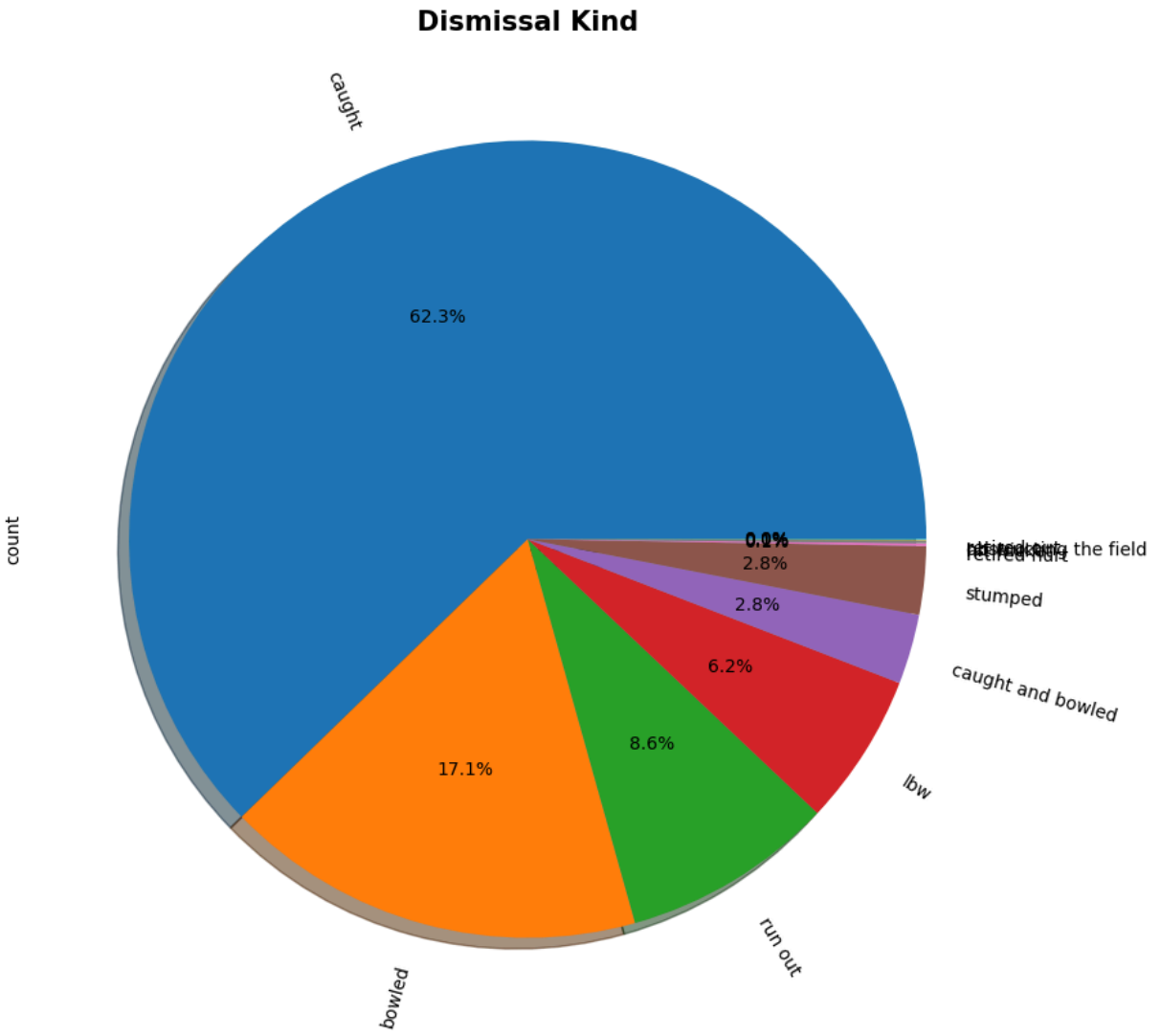
In [69]:
```
df.groupby(["batter","season"])["batsman_runs"].sum().sort_values(ascending=
```

Out[69]:

batter	season	
V Kohli	2016	973
Shubman Gill	2023	890
JC Buttler	2022	863
DA Warner	2016	848
V Kohli	2024	741
KS Williamson	2018	735
MEK Hussey	2013	733
CH Gayle	2012	733
F du Plessis	2023	730
CH Gayle	2013	720

Name: batsman_runs, dtype: int64

In [70]:
```
plt.subplots(figsize=(10, 18))
dev['dismissal_kind'].value_counts().plot.pie(autopct='%1.1f%%',shadow=True,
plt.title("Dismissal Kind",fontweight="bold",fontsize=15)
plt.show()
```



In [71]:
```
dev.dismissal_kind.unique()
```

Out[71]:
```
array([nan, 'caught', 'bowled', 'run out', 'lbw', 'retired hurt',
       'stumped', 'caught and bowled', 'hit wicket',
       'obstructing the field', 'retired out'], dtype=object)
```

BOLLING OVERVIEW

In [72]:
```
dev.head(2)
```

Out[72]:

	match_id	inning	batting_team	bowling_team	over	ball	batter	bowle
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly	P Kuma
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum	P Kuma

In [73]:
```
eco=dev.groupby("bowler").agg({"batsman_runs":"sum","ball":"count"}).sort_va
```

In [74]:
```
eco["economy"]=eco["batsman_runs"]/(eco["ball"]/6)
```

In [75]:
```
eco.head(10)
```

Out[75]:

bowler	batsman_runs	ball	economy
R Ashwin	5178	4679	6.639880
SP Narine	4492	4146	6.500724
B Kumar	4744	4060	7.010837
PP Chawla	5027	3895	7.743774
RA Jadeja	4777	3895	7.358665
YS Chahal	4478	3628	7.405733
Harbhajan Singh	3928	3496	6.741419
A Mishra	4065	3444	7.081882
DJ Bravo	4178	3296	7.605583
UT Yadav	4185	3190	7.871473

In [76]:
```
dev.head()
```

Out[76]:

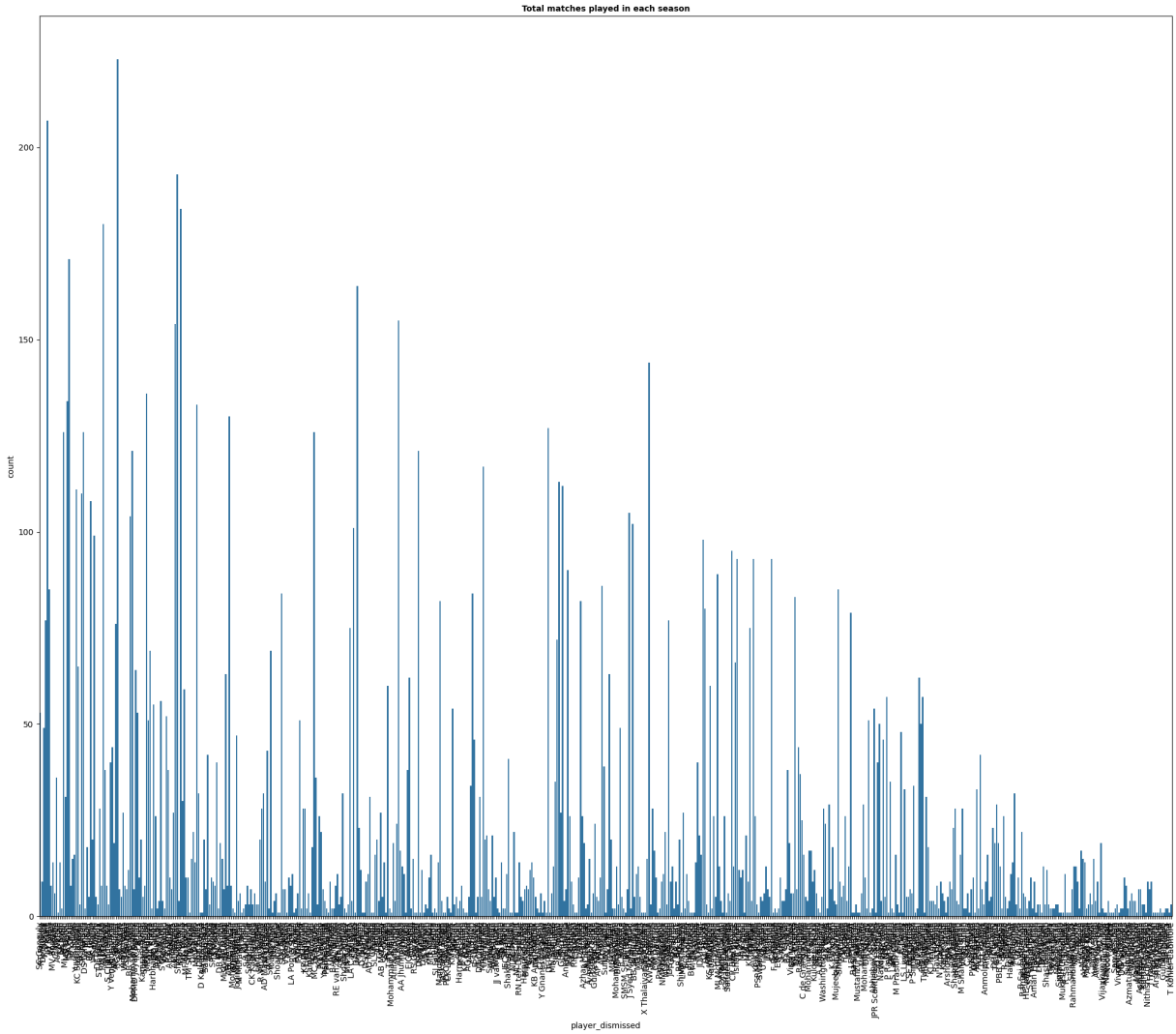
	match_id	inning	batting_team	bowling_team	over	ball	batter	bowle
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly	I Kuma
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum	I Kuma
2	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	3	BB McCullum	I Kuma
3	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	4	BB McCullum	I Kuma
4	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	5	BB McCullum	I Kuma

In [77]:
```
df.groupby('bowler').agg({'total_runs':'sum','ball':'count','player_dismisse
```

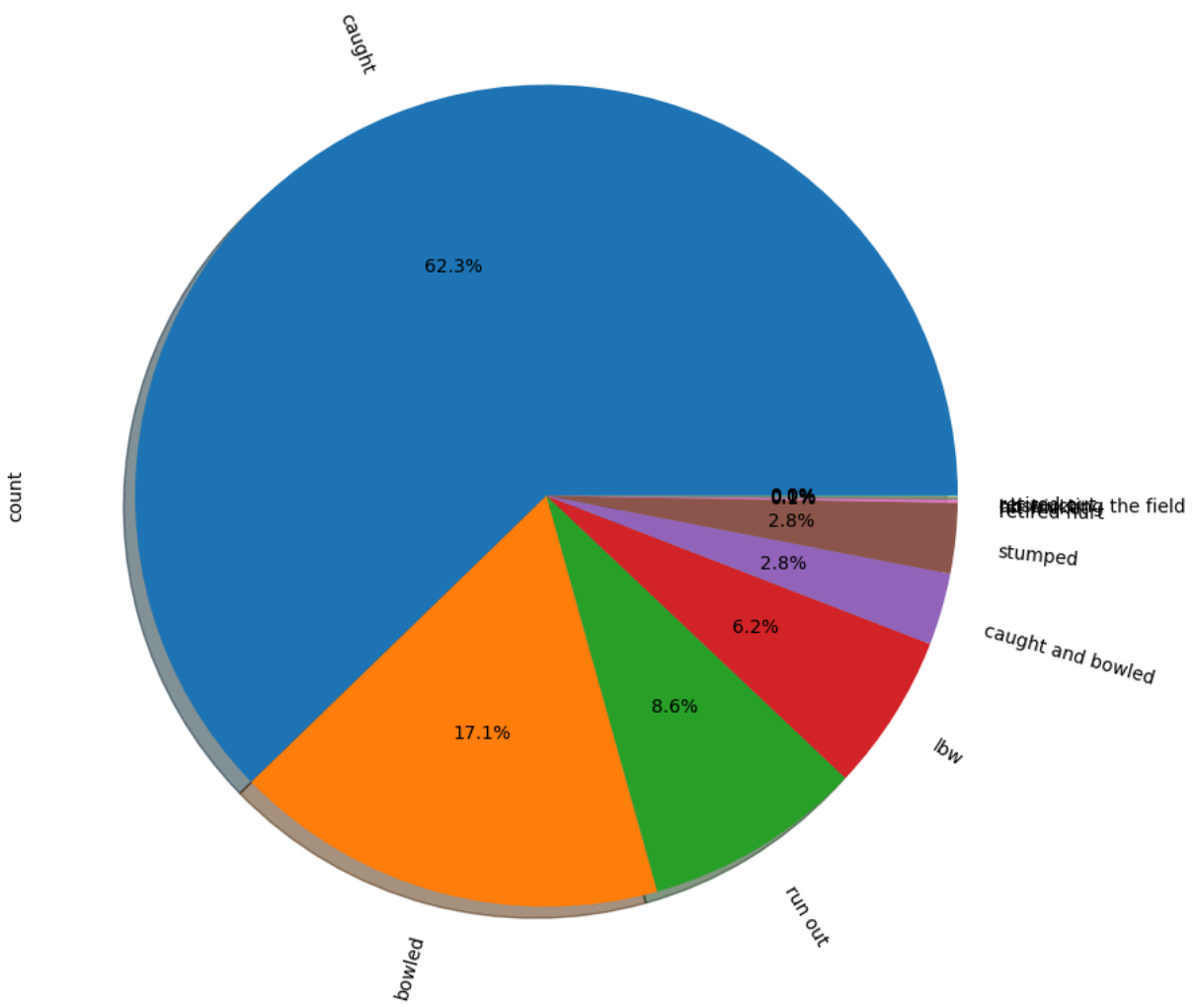
Out[77]:

bowler	total_runs	ball	player_dismissed
R Ashwin	5435	4679	198
PP Chawla	5179	3895	201
B Kumar	5051	4060	195
RA Jadeja	4917	3895	169
YS Chahal	4681	3628	213
SP Narine	4672	4146	200
UT Yadav	4442	3190	163
DJ Bravo	4436	3296	207
A Mishra	4193	3444	183
Harbhajan Singh	4101	3496	161

In [78]:
```
plt.subplots(figsize=(25, 20))
sns.countplot(x="player_dismissed",data=dev)
plt.xticks(rotation=90)
plt.title('Total matches played in each season', fontsize = 10, fontweight =
plt.show()
```



In [79]:
```
plt.subplots(figsize=(10, 18))
dev['dismissal_kind'].value_counts().plot.pie(autopct='%1.1f%%',shadow=True,
plt.title("Dismissal Kind",fontweight="bold",fontsize=15)
plt.show()
```



In [ ]:

In [ ]: