In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
data = pd.read_csv('ipl 2024 final dataset.csv')
```

In [3]:
```
data
```

Out[3]:

	match_id	season	match_no	date	venue	batting_team	bowling_team	innings	over	over.1	...	bowler	runs_of
0	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.1	0	...	Chahar	
1	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.1	0	...	Chahar	
2	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.2	0	...	Chahar	
3	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.3	0	...	Chahar	
4	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.4	0	...	Chahar	
...	...	...	...	...	...	...	...	...	...	...	...	...	
17048	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	9.5	9	...	Markram	
17049	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	9.6	9	...	Markram	
17050	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	10.1	10	...	Shahbaz Ahmed	
17051	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	10.2	10	...	Shahbaz Ahmed	
17052	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	10.3	10	...	Shahbaz Ahmed	

17053 rows × 23 columns

In [4]:
```
data.columns
```

Out[4]:
```
Index(['match_id', 'season', 'match_no', 'date', 'venue', 'batting_team',
       'bowling_team', 'innings', 'over', 'over.1', 'Over_no', 'Ball_no',
       'striker', 'bowler', 'runs_of_bat', 'extras', 'wide', 'legbyes', 'byes',
       'noballs', 'wicket_type', 'player_dismissed', 'fielder'],
      dtype='object')
```

In [5]:
```
data.head()
```

Out[5]:

	match_id	season	match_no	date	venue	batting_team	bowling_team	innings	over	over.1	...	bowler	runs_of_bat
0	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.1	0	...	Chahar	0
1	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.1	0	...	Chahar	1
2	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.2	0	...	Chahar	0
3	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.3	0	...	Chahar	0
4	202401	2024	1	Mar 22, 2024	MA Chidambaram Stadium, Chennai	RCB	CSK	1	0.4	0	...	Chahar	0

5 rows × 23 columns

In [6]:
```
data.tail()
```

Out[6]:

	match_id	season	match_no	date	venue	batting_team	bowling_team	innings	over	over.1	...	bowler	runs_of
17048	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	9.5	9	...	Markram	
17049	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	9.6	9	...	Markram	
17050	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	10.1	10	...	Shahbaz Ahmed	
17051	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	10.2	10	...	Shahbaz Ahmed	
17052	202474	2024	74	May 26, 2024	MA Chidambaram Stadium, Chennai	KKR	CSK	2	10.3	10	...	Shahbaz Ahmed	

5 rows × 23 columns

In [7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17053 entries, 0 to 17052
Data columns (total 23 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   match_id          17053 non-null  int64  
 1   season            17053 non-null  int64  
 2   match_no          17053 non-null  int64  
 3   date              17053 non-null  object 
 4   venue             17053 non-null  object 
 5   batting_team      17053 non-null  object 
 6   bowling_team      17053 non-null  object 
 7   innings           17053 non-null  int64  
 8   over              17053 non-null  float64
 9   over.1            17053 non-null  int64  
 10  Over_no           17053 non-null  int64  
 11  Ball_no           17053 non-null  int64  
 12  striker           17053 non-null  object 
 13  bowler            17053 non-null  object 
 14  runs_of_bat       17053 non-null  int64  
 15  extras            17053 non-null  int64  
 16  wide              17053 non-null  int64  
 17  legbyes           17053 non-null  int64  
 18  byes              17053 non-null  int64  
 19  noballs           17053 non-null  int64  
 20  wicket_type       883 non-null    object 
 21  player_dismissed  883 non-null    object 
 22  fielder           709 non-null    object 
dtypes: float64(1), int64(13), object(9)
memory usage: 3.0+ MB
```

In [8]:
```
data.isnull().sum()
```

Out[8]:
```
match_id              0
season                0
match_no              0
date                  0
venue                 0
batting_team          0
bowling_team          0
innings               0
over                  0
over.1                0
Over_no               0
Ball_no               0
striker               0
bowler                0
runs_of_bat           0
extras                0
wide                  0
legbyes               0
byes                  0
noballs               0
wicket_type       16170
player_dismissed  16170
fielder           16344
dtype: int64
```

In [9]:
```
# Filling the empty values

wicket_type_mode = data['wicket_type'].mode()[0]
data['wicket_type'] = data['wicket_type'].fillna(wicket_type_mode)

player_dismissed_mode = data['player_dismissed'].mode()[0]
data['player_dismissed'] = data['player_dismissed'].fillna(player_dismissed_mode)

fielder_mode = data['fielder'].mode()[0]
data['fielder'] = data['fielder'].fillna(fielder_mode)
```

In [10]:
```
data.isnull().sum()
```

Out[10]:
```
match_id            0
season              0
match_no            0
date                0
venue               0
batting_team        0
bowling_team        0
innings             0
over                0
over.1              0
Over_no             0
Ball_no             0
striker             0
bowler              0
runs_of_bat         0
extras              0
wide                0
legbyes             0
byes                0
noballs             0
wicket_type         0
player_dismissed    0
fielder             0
dtype: int64
```

In [11]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17053 entries, 0 to 17052
Data columns (total 23 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   match_id          17053 non-null  int64
 1   season            17053 non-null  int64
 2   match_no          17053 non-null  int64
 3   date              17053 non-null  object
 4   venue             17053 non-null  object
 5   batting_team      17053 non-null  object
 6   bowling_team      17053 non-null  object
 7   innings           17053 non-null  int64
 8   over              17053 non-null  float64
 9   over.1            17053 non-null  int64
 10  Over_no           17053 non-null  int64
 11  Ball_no           17053 non-null  int64
 12  striker           17053 non-null  object
 13  bowler            17053 non-null  object
 14  runs_of_bat       17053 non-null  int64
 15  extras            17053 non-null  int64
 16  wide              17053 non-null  int64
 17  legbyes           17053 non-null  int64
 18  byes              17053 non-null  int64
 19  noballs           17053 non-null  int64
 20  wicket_type       17053 non-null  object
 21  player_dismissed  17053 non-null  object
 22  fielder           17053 non-null  object
dtypes: float64(1), int64(13), object(9)
memory usage: 3.0+ MB
```

In [12]:
```
data.shape
```

Out[12]:
```
(17053, 23)
```

In [13]:
```
data.describe()
```

Out[13]:

	match_id	season	match_no	innings	over	over.1	Over_no	Ball_no	runs_of_bat
count	17053.000000	17053.0	17053.000000	17053.000000	17053.000000	17053.000000	17053.000000	17053.000000	17053.000000
mean	202435.902656	2024.0	35.902656	1.480033	9.570474	9.221955	10.221955	3.485193	1.445904
std	20.896995	0.0	20.896995	0.499616	5.674610	5.673515	5.673515	1.707800	1.800340
min	202401.000000	2024.0	1.000000	1.000000	0.100000	0.000000	1.000000	1.000000	0.000000
25%	202418.000000	2024.0	18.000000	1.000000	4.500000	4.000000	5.000000	2.000000	0.000000
50%	202436.000000	2024.0	36.000000	1.000000	9.500000	9.000000	10.000000	3.000000	1.000000
75%	202453.000000	2024.0	53.000000	2.000000	14.400000	14.000000	15.000000	5.000000	2.000000
max	202474.000000	2024.0	74.000000	2.000000	19.600000	19.000000	20.000000	6.000000	6.000000

Total matches played in ipl

In [14]:
```
total_matches = data['match_id'].nunique()
```

In [15]:
```
print(f"Total played matches in IPL are:", total_matches)
```

```
Total played matches in IPL are: 71
```

In [16]:
```
data.columns
```

Out[16]:
```
Index(['match_id', 'season', 'match_no', 'date', 'venue', 'batting_team',
       'bowling_team', 'innings', 'over', 'over.1', 'Over_no', 'Ball_no',
       'striker', 'bowler', 'runs_of_bat', 'extras', 'wide', 'legbyes', 'byes',
       'noballs', 'wicket_type', 'player_dismissed', 'fielder'],
      dtype='object')
```

Total runs scored by each batsman

In [41]:
```
runs_scored = data.groupby('striker')['runs_of_bat'].sum().reset_index()
runs_scored.columns = ['striker', 'runs_scored']
```

In [42]:
```
runs_scored.head()
```

Out[42]:

	striker	runs_scored
0	Abdul Samad	182
1	Abhinav Manohar	9
2	Abhishek Sharma	484
3	Abishek Porel	327
4	Akash Deep	2

Calculate balls faced by each batsman

In [43]:
```
valid_balls_faced = data[(data['wide'] == 0) & (data['noballs'] == 0)]
```

In [44]:
```
balls_faced = valid_balls_faced.groupby('striker').size().reset_index(name='total_balls_faced')
```

In [45]:
```
batting_stats = pd.merge(runs_scored, balls_faced, on='striker')
```

In [46]:
```
batting_stats['strike_rate'] = (batting_stats['runs_scored'] / batting_stats['total_balls_faced'])*100
```

In [48]:
```
batting_stats[['striker', 'runs_scored', 'total_balls_faced', 'strike_rate']].head()
```

Out[48]:

	striker	runs_scored	total_balls_faced	strike_rate
0	Abdul Samad	182	108	168.518519
1	Abhinav Manohar	9	16	56.250000
2	Abhishek Sharma	484	235	205.957447
3	Abishek Porel	327	204	160.294118
4	Akash Deep	2	2	100.000000

In [49]:
```
Fours = data[data['runs_of_bat'] == 4].groupby('striker').size().reset_index(name='Fours')
```

In [50]:
```
Fours.head()
```

Out[50]:

	striker	Fours
0	Abdul Samad	15
1	Abhinav Manohar	1
2	Abhishek Sharma	36
3	Abishek Porel	36
4	Angkrish Raghuvanshi	16

In [51]:
```
Sixes = data[data['runs_of_bat'] == 6].groupby('striker').size().reset_index(name='Sixes')
```

In [52]:
```
Sixes.head()
```

Out[52]:

	striker	Sixes
0	Abdul Samad	11
1	Abhishek Sharma	42
2	Abishek Porel	13
3	Angkrish Raghuvanshi	8
4	Anuj Rawat	3

In [53]:
```
inning_runs = data.groupby(['striker', 'match_id'])['runs_of_bat'].sum().reset_index()
```

In [56]:
```
Centuries = inning_runs[inning_runs['runs_of_bat'] >= 100].groupby('striker').size().reset_index(name='Centurie
```

In [57]:
```
Centuries.head()
```

Out[57]:

	striker	Centuries
0	Bairstow	1
1	Buttler	2
2	Gaikwad	1
3	Head	1
4	Jaiswal	1

In [59]:
```
Fifties = inning_runs[(inning_runs['runs_of_bat'] >= 50) & (inning_runs['runs_of_bat'] < 100)].groupby('striker
```

In [60]:
```
Fifties.head()
```

Out[60]:

	striker	Fifties
0	Abhishek Sharma	3
1	Abishek Porel	2
2	Angkrish Raghuvanshi	1
3	Arshad Khan	1
4	Ashutosh Sharma	1

In [65]:
```
wickets = data[data['wicket_type'].notna()].groupby('bowler').size().reset_index(name='total_wickets')
```

In [66]:
```
valid_balls_bowled = data[(data['wide'] == 0) & (data['noballs'] == 0)]
```

In [67]:
```
balls_bowled = valid_balls_bowled.groupby('bowler').size().reset_index(name='balls_bowled')
```

In [71]:
```
bowling_stats = pd.merge(wickets, balls_bowled, on='bowler')
```

In [74]:
```
bowling_stats['strike_rate'] = bowling_stats['balls_bowled'] / bowling_stats['total_wickets']
```

In [75]:
```
bowling_stats[['bowler', 'balls_bowled', 'total_wickets', 'strike_rate']].head()
```

Out[75]:

	bowler	balls_bowled	total_wickets	strike_rate
0	Abhishek Sharma	42	42	1.000000
1	Akash Deep	21	23	0.913043
2	Akash Madhwal	109	119	0.915966
3	Alzarri Joseph	58	64	0.906250
4	Anshul Kamboj	60	66	0.909091

In [76]:
```
bowling_stats['overs_bowled'] = bowling_stats['balls_bowled'] // 6 + (bowling_stats['balls_bowled'] % 6) / 6
```

In [80]:
```
data['total_runs_conceded'] = data['runs_of_bat'] + data['extras']
```

In [87]:
```
runs_conceded = data.groupby('bowler')['total_runs_conceded'].sum().reset_index(name='total_run_conceded')
```

In [88]:
```
bowling_stats = pd.merge(bowling_stats, runs_conceded, on='bowler')
```

In [89]:
```
bowling_stats['economy_rate'] = bowling_stats['total_run_conceded'] / bowling_stats['overs_bowled']
```

In [91]:
```
bowling_stats[['bowler', 'balls_bowled', 'overs_bowled', 'total_run_conceded', 'total_wickets', 'economy_rate']
```

Out[91]:

	bowler	balls_bowled	overs_bowled	total_run_conceded	total_wickets	economy_rate
0	Abhishek Sharma	42	7.000000	51	42	7.285714
1	Akash Deep	21	3.500000	55	23	15.714286
2	Akash Madhwal	109	18.166667	209	119	11.504587
3	Alzarri Joseph	58	9.666667	123	64	12.724138
4	Anshul Kamboj	60	10.000000	113	66	11.300000

In [93]:
```
maidens = data.groupby(['bowler', 'over'])['runs_of_bat'].sum().reset_index().groupby('bowler').apply(lambda x:
```

In [94]:
```
maidens.head()
```

Out[94]:

	bowler	maidens
0	Abhishek Sharma	8
1	Akash Deep	6
2	Akash Madhwal	20
3	Alzarri Joseph	9
4	Anshul Kamboj	16

In [95]:
```
data['wicket_type'].value_counts()
```

Out[95]:
```
wicket_type
caught                    16814
bowled                      123
runout                       52
lbw                          50
stumped                      13
obstructing the field         1
Name: count, dtype: int64
```

In [96]:
```
catches = data[data['wicket_type'] == 'caught'].groupby('fielder').size().reset_index(name='catches_taken')
```

In [97]:
```
catches.head()
```

Out[97]:

	fielder	catches_taken
0	(sub)Abhinav Manohar	1
1	(sub)Brevis	1
2	(sub)Donovan Ferreira	1
3	(sub)Fraser-McGurk	2
4	(sub)Gowtham	1

In [98]:
```
bowled = data[data['wicket_type'] == 'bowled'].groupby('fielder').size().reset_index(name='bowled')
```

In [99]:
```
bowled.head()
```

Out[99]:

	fielder	bowled
0	Rahul	123

In [100…
```
run_out = data[data['wicket_type'] == 'run_out'].groupby('fielder').size().reset_index(name='run_out')
```

In [101…
```
run_out.head()
```

Out[101…

	fielder	run_out

In [103…
```
runs_by_team = data.groupby('batting_team')['runs_of_bat'].sum().reset_index()
runs_by_team.columns = ['Team', 'Total_Runs']
```

In [104…
```
runs_by_team
```

Out[104…

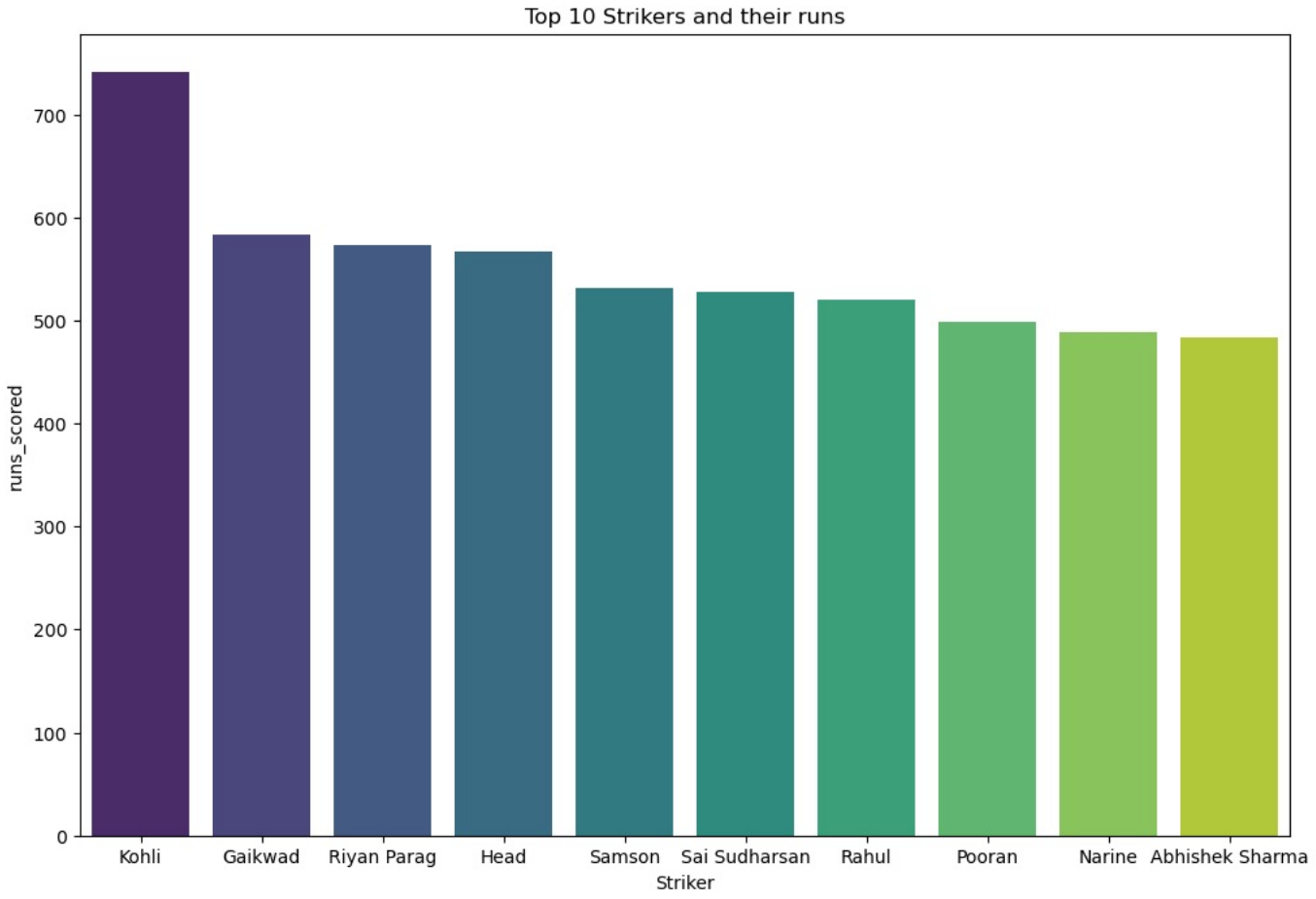
	Team	Total_Runs
0	CSK	2405
1	DC	2465
2	GT	1934
3	KKR	2481
4	LSG	2350
5	MI	2422
6	PBKS	2372
7	RCB	2789
8	RR	2533
9	SRH	2906

Visualizations

In [111…
```
top_run_scorers = batting_stats.sort_values(by='runs_scored', ascending=False).head(10)
```

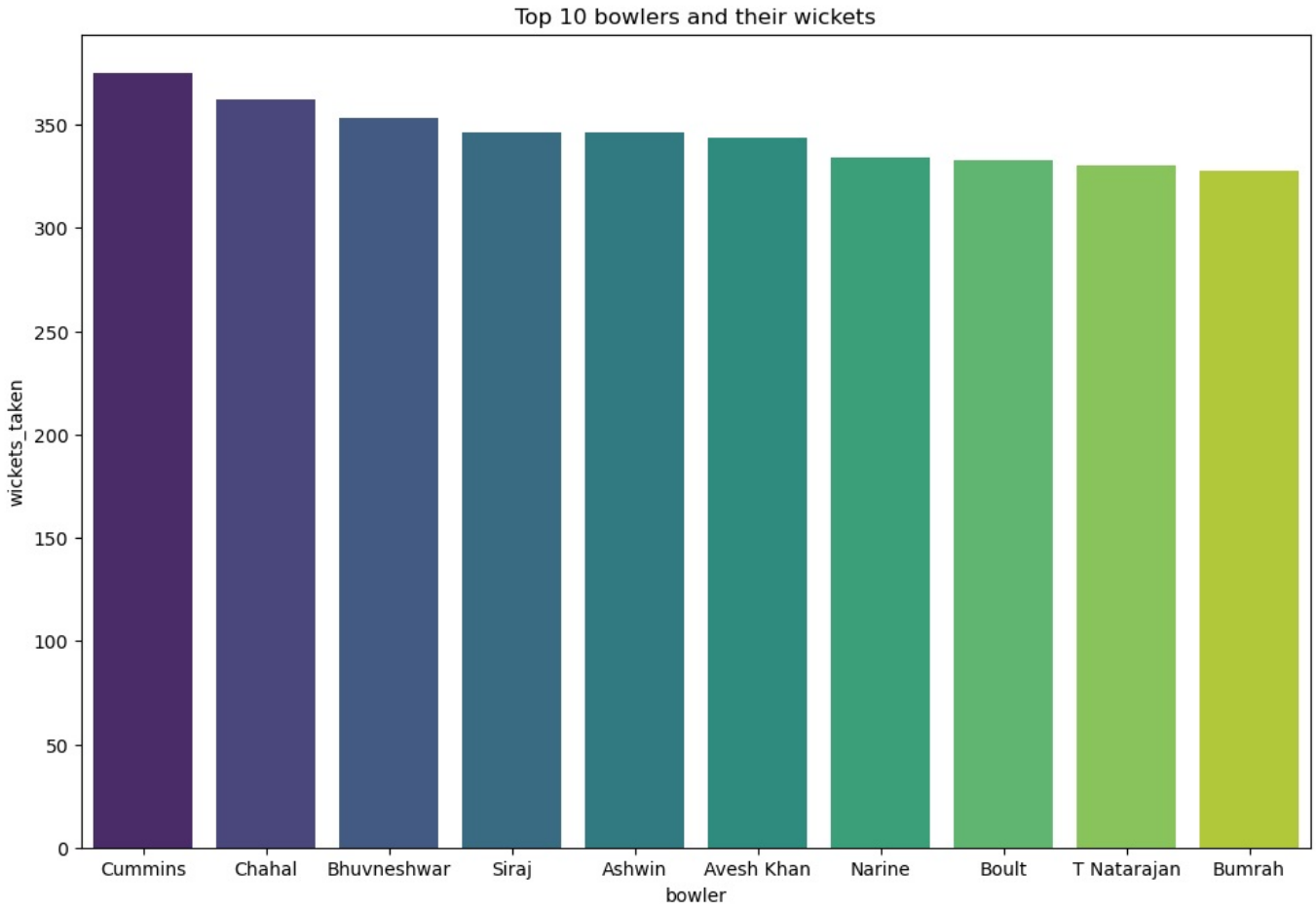
In [112…
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [118…
```
plt.figure(figsize=(12, 8))
sns.barplot(x='striker', y='runs_scored', data=top_run_scorers, palette='viridis')
plt.title("Top 10 Strikers and their runs")
plt.xlabel('Striker')
plt.ylabel('runs_scored')
plt.show()
```

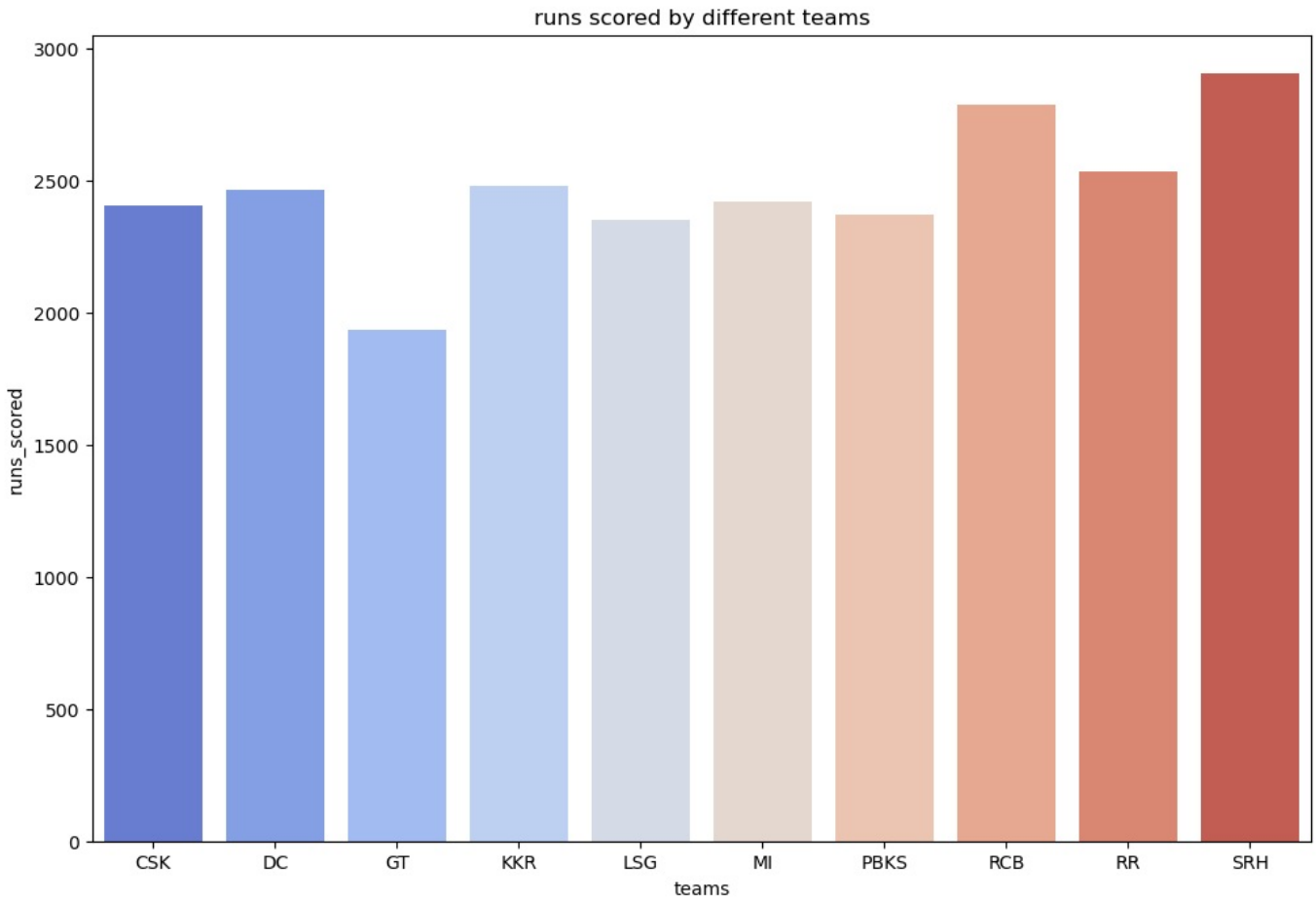


In [119…
```
top_wicket_takers = bowling_stats.sort_values(by='total_wickets', ascending=False).head(10)
```

In [120…
```
plt.figure(figsize=(12, 8))
sns.barplot(x='bowler', y='total_wickets', data=top_wicket_takers, palette='viridis')
plt.title("Top 10 bowlers and their wickets")
plt.xlabel('bowler')
plt.ylabel('wickets_taken')
plt.show()
```

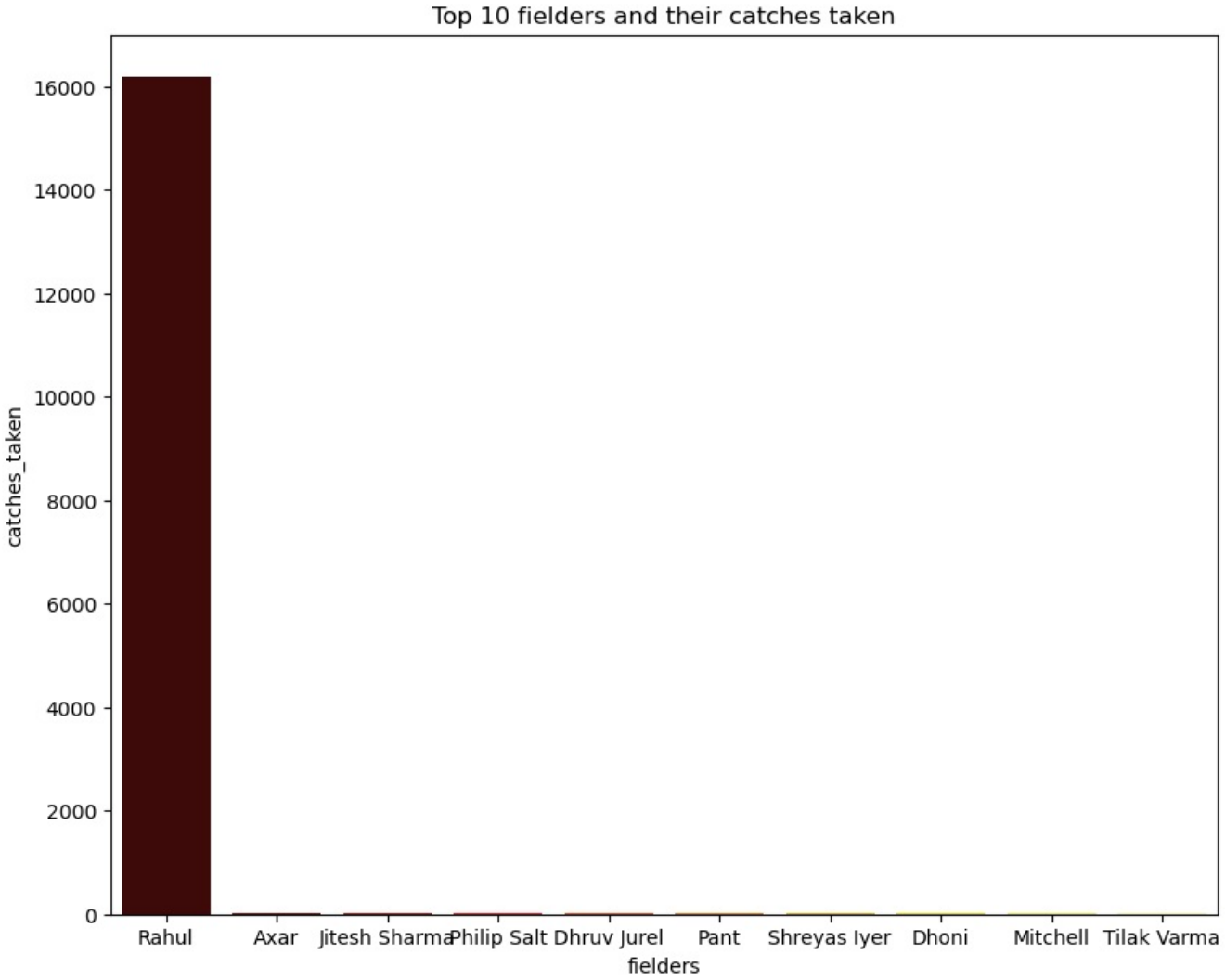


In [121…
```
plt.figure(figsize=(12, 8))
sns.barplot(x='Team', y='Total_Runs', data=runs_by_team, palette='coolwarm')
plt.title("runs scored by different teams")
plt.xlabel('teams')
plt.ylabel('runs_scored')
plt.show()
```



In [122…
```
top_catches_taken = catches.sort_values(by='catches_taken', ascending=False).head(10)
```

In [124…
```
plt.figure(figsize=(10, 8))
sns.barplot(x='fielder', y='catches_taken', data=top_catches_taken, palette='hot')
plt.title("Top 10 fielders and their catches taken")
plt.xlabel('fielders')
plt.ylabel('catches_taken')
plt.show()
```



In [125…
```
top_bowled = bowled.sort_values(by='bowled', ascending=False).head(10)
```

In [126…
```
plt.figure(figsize=(10, 8))
sns.barplot(x='fielder', y='bowled', data=top_bowled, palette='hot')
plt.title("Top 10 fielders and their bowled")
plt.xlabel('fielders')
plt.ylabel('bowled')
plt.show()
```