

IPL 2022 - EDA

February 20, 2024

1 Indian Premier League (IPL)

Indian professional Twenty20 (T20) cricket league established in 2008. The league, which is based on a round-robin group and knockout format, has teams in major Indian cities.

The brainchild of the Board of Control for Cricket in India (BCCI), the IPL has developed into the most lucrative and most popular outlet for the game of cricket. Matches generally begin in late afternoon or evening so that at least a portion of them are played under floodlights at night to maximize the television audience for worldwide broadcasts. Initially, league matches were played on a home-and-away basis between all teams, but, with the planned expansion to 10 clubs (divided into two groups of five) in 2011, that format changed so that matches between some teams would be limited to a single encounter. The top four teams contest three play-off matches, with one losing team being given a second chance to reach the final, a wrinkle aimed at maximizing potential television revenue. The play-off portion of the tournament involves the four teams that finished at the top of the tables in a series of knockout games that allows one team that lost its first-round game a second chance to advance to the final match.

1.0.1 Import Libraries

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

1.0.2 Read Dataset into Pandas DataFrame

```
[2]: df=pd.read_csv("Book_ip122_ver_33.csv")
print(df.shape)
```

(74, 20)

```
[3]: df.head(10)
```

```
[3]:   match_id      date      venue
0         1  March 26,2022  Wankhede Stadium, Mumbai \
1         2  March 27,2022  Brabourne Stadium, Mumbai
2         3  March 27,2022  Dr DY Patil Sports Academy, Mumbai
3         4  March 28,2022  Wankhede Stadium, Mumbai
```

4	5	March 29,2022	Maharashtra Cricket Association Stadium,Pune
5	6	March 30,2022	Dr DY Patil Sports Academy, Mumbai
6	7	March 31,2022	Brabourne Stadium, Mumbai
7	8	April 1,2022	Wankhede Stadium, Mumbai
8	9	April 2,2022	Dr DY Patil Sports Academy, Mumbai
9	10	April 2,2022	Maharashtra Cricket Association Stadium,Pune

	team1	team2	stage	toss_winner	toss_decision	first_ings_score	
0	Chennai	Kolkata	Group	Kolkata	Field	131	\
1	Delhi	Mumbai	Group	Delhi	Field	177	
2	Bangalore	Punjab	Group	Punjab	Field	205	
3	Gujarat	Lucknow	Group	Gujarat	Field	158	
4	Hyderabad	Rajasthan	Group	Hyderabad	Field	210	
5	Bangalore	Kolkata	Group	Bangalore	Field	128	
6	Chennai	Lucknow	Group	Lucknow	Field	210	
7	Kolkata	Punjab	Group	Kolkata	Field	137	
8	Mumbai	Rajasthan	Group	Mumbai	Field	193	
9	Delhi	Gujarat	Group	Delhi	Field	171	

	first_ings_wkts	second_ings_score	second_ings_wkts	match_winner	won_by	
0	5	133	4	Kolkata	Wickets	\
1	5	179	6	Delhi	Wickets	
2	2	208	5	Punjab	Wickets	
3	6	161	5	Gujarat	Wickets	
4	6	149	7	Rajasthan	Runs	
5	10	132	7	Bangalore	Wickets	
6	7	211	4	Lucknow	Wickets	
7	10	141	4	Kolkata	Wickets	
8	8	170	8	Rajasthan	Runs	
9	6	157	9	Gujarat	Runs	

	margin	player_of_the_match	top_scorer	highscore	
0	6	Umesh Yadav	MS Dhoni	50	\
1	4	Kuldeep Yadav	Ishan Kishan	81	
2	5	Odean Smith	Faf du Plessis	88	
3	5	Mohammed Shami	Deepak Hooda	55	
4	61	Sanju Samson	Aiden Markram	57	
5	3	Wanindu Hasaranga	Sherfane Rutherford	28	
6	6	Evin Lewis	Quinton de Kock	61	
7	6	Umesh Yadav	Andre Russell	70	
8	23	Jos Buttler	Jos Buttler	100	
9	14	Lockie Ferguson	Shubman Gill	84	

	best_bowling	best_bowling_figure
0	Dwayne Bravo	3--20
1	Kuldeep Yadav	3--18
2	Mohammed Siraj	2--59

3	Mohammed Shami	3--25
4	Yuzvendra Chahal	3--22
5	Wanindu Hasaranga	4--20
6	Ravi Bishnoi	2--24
7	Umesh Yadav	4--23
8	Jasprit Bumrah	3--17
9	Lockie Ferguson	4--28

```
[4]: df.tail(10)
```

```
[4]:
```

	match_id	date	venue	team1
64	65	May 17,2022	Wankhede Stadium, Mumbai	Hyderabad \
65	66	May 18,2022	Dr DY Patil Sports Academy, Mumbai	Kolkata
66	67	May 19,2022	Wankhede Stadium, Mumbai	Banglore
67	68	May 20,2022	Brabourne Stadium, Mumbai	Chennai
68	69	May 21,2022	Wankhede Stadium, Mumbai	Delhi
69	70	May 22,2022	Wankhede Stadium, Mumbai	Hyderabad
70	71	May 24,2022	Eden Gardens, Kolkata	Gujarat
71	72	May 25,2022	Eden Gardens, Kolkata	Banglore
72	73	May 27,2022	Narendra Modi Stadium, Ahmedabad	Banglore
73	74	May 29,2022	Narendra Modi Stadium, Ahmedabad	Gujarat

	team2	stage	toss_winner	toss_decision	first_ings_score
64	Mumbai	Group	Mumbai	Field	193 \
65	Lucknow	Group	Lucknow	Bat	210
66	Gujarat	Group	Gujarat	Bat	168
67	Rajasthan	Group	Chennai	Bat	150
68	Mumbai	Group	Mumbai	Field	159
69	Punjab	Group	Hyderabad	Bat	157
70	Rajasthan	Playoff	Gujarat	Field	188
71	Lucknow	Playoff	Lucknow	Field	207
72	Rajasthan	Playoff	Rajasthan	Field	157
73	Rajasthan	Final	Rajasthan	Bat	130

	first_ings_wkts	second_ings_score	second_ings_wkts	match_winner
64	6	190	7	Hyderabad \
65	0	208	8	Lucknow
66	5	170	2	Banglore
67	6	151	5	Rajasthan
68	7	160	5	Mumbai
69	8	160	5	Punjab
70	6	191	3	Gujarat
71	4	193	6	Banglore
72	8	161	3	Rajasthan
73	9	133	3	Gujarat

won_by	margin	player_of_the_match	top_scorer	highscore
--------	--------	---------------------	------------	-----------

64	Runs	3	Rahul Tripathi	Rahul Tripathi	76 \
65	Runs	2	Quinton de Kock	Quinton de Kock	140
66	Wickets	8	Virat Kohli	Virat Kohli	73
67	Wickets	5	R Aswin	Moeen Ali	93
68	Wickets	5	Jasprit Bumrah	Ishan Kishan	48
69	Wickets	5	Harpreet Brar	Liam Livingstone	49
70	Wickets	7	David Miller	Jos Buttler	89
71	Runs	14	Rajat Patidar	Rajat Patidar	112
72	Wickets	7	Jos Buttler	Jos Buttler	106
73	Wickets	7	Hardik Pandya	Shubman Gill	45

	best_bowling	best_bowling_figure
64	Ramandeep Singh	3--20
65	Mohsin Khan	3--20
66	Rashid Khan	2--32
67	Prashant Solanki	2--20
68	Jasprit Bumrah	3--25
69	Harpreet Brar	3--26
70	Hardik Pandya	1--14
71	Josh Hazlewood	3--43
72	Prasidh Krishna	3--22
73	Hardik Pandya	3--17

1.0.3 Print Information about the DataFrame

```
[5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74 entries, 0 to 73
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   match_id              74 non-null    int64
1   date                  74 non-null    object
2   venue                 74 non-null    object
3   team1                 74 non-null    object
4   team2                 74 non-null    object
5   stage                 74 non-null    object
6   toss_winner           74 non-null    object
7   toss_decision         74 non-null    object
8   first_ings_score      74 non-null    int64
9   first_ings_wkts       74 non-null    int64
10  second_ings_score     74 non-null    int64
11  second_ings_wkts      74 non-null    int64
12  match_winner          74 non-null    object
13  won_by                74 non-null    object
14  margin                74 non-null    int64
15  player_of_the_match   74 non-null    object
```

```

16 top_scorer          74 non-null    object
17 highscore           74 non-null    int64
18 best_bowling        74 non-null    object
19 best_bowling_figure 74 non-null    object
dtypes: int64(7), object(13)
memory usage: 11.7+ KB

```

1.0.4 Check Description of the data in the DataFrame

```
[6]: df.describe(include="all")
```

```

[6]:      match_id      date      venue      team1      team2
count  74.000000      74      74      74      74 \
unique    NaN      62      6      9      9
top      NaN  May 1,2022  Wankhede Stadium, Mumbai  Bangalore  Rajasthan
freq      NaN      2      21      16      17
mean   37.500000    NaN    NaN    NaN    NaN
std    21.505813    NaN    NaN    NaN    NaN
min     1.000000    NaN    NaN    NaN    NaN
25%    19.250000    NaN    NaN    NaN    NaN
50%    37.500000    NaN    NaN    NaN    NaN
75%    55.750000    NaN    NaN    NaN    NaN
max     74.000000    NaN    NaN    NaN    NaN

      stage toss_winner toss_decision first_ings_score first_ings_wkts
count      74      74      74      74.000000      74.000000 \
unique      3      10      2      NaN      NaN
top    Group    Gujarat    Field      NaN      NaN
freq      70      10      59      NaN      NaN
mean      NaN      NaN      NaN    171.121622    6.135135
std      NaN      NaN      NaN    29.048355    2.222699
min      NaN      NaN      NaN    68.000000    0.000000
25%      NaN      NaN      NaN    154.250000    5.000000
50%      NaN      NaN      NaN    169.500000    6.000000
75%      NaN      NaN      NaN    192.750000    8.000000
max      NaN      NaN      NaN    222.000000    10.000000

      second_ings_score second_ings_wkts match_winner won_by      margin
count      74.000000      74.000000      74      74  74.000000 \
unique      NaN      NaN      10      2      NaN
top      NaN      NaN      Gujarat  Wickets      NaN
freq      NaN      NaN      12      37      NaN
mean    158.540541      6.175676      NaN      NaN  16.972973
std      29.299207      2.639832      NaN      NaN  19.651047
min      72.000000      1.000000      NaN      NaN   2.000000
25%    142.750000      4.000000      NaN      NaN   5.250000
50%    160.000000      6.000000      NaN      NaN   8.000000

```

75%	176.000000	8.000000	NaN	NaN	18.000000
max	211.000000	10.000000	NaN	NaN	91.000000

	player_of_the_match	top_scorer	highscore	best_bowling
count	74	74	74.000000	74 \
unique	56	37	NaN	42
top	Kuldeep Yadav	Jos Buttler	NaN	Yuzvendra Chahal
freq	4	7	NaN	5
mean	NaN	NaN	71.716216	NaN
std	NaN	NaN	20.705052	NaN
min	NaN	NaN	28.000000	NaN
25%	NaN	NaN	57.000000	NaN
50%	NaN	NaN	68.000000	NaN
75%	NaN	NaN	87.750000	NaN
max	NaN	NaN	140.000000	NaN

	best_bowling_figure
count	74
unique	51
top	3--22
freq	5
mean	NaN
std	NaN
min	NaN
25%	NaN
50%	NaN
75%	NaN
max	NaN

1.0.5 Cleaning the Dataset

Cleaning a dataset is indeed a crucial step in any data analysis (EDA stands for Exploratory Data Analysis). Ensuring the data is clean and free of null or missing values is one of the fundamental tasks in data preprocessing.

```
[7]: df.isnull()
```

```
[7]:
```

	match_id	date	venue	team1	team2	stage	toss_winner	toss_decision
0	False	False	False	False	False	False	False	False \
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
..
69	False	False	False	False	False	False	False	False
70	False	False	False	False	False	False	False	False
71	False	False	False	False	False	False	False	False
72	False	False	False	False	False	False	False	False

73	False	False	False	False	False	False	False	False
----	-------	-------	-------	-------	-------	-------	-------	-------

	first_ings_score	first_ings_wkts	second_ings_score	second_ings_wkts
0	False	False	False	False \
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
..
69	False	False	False	False
70	False	False	False	False
71	False	False	False	False
72	False	False	False	False
73	False	False	False	False

	match_winner	won_by	margin	player_of_the_match	top_scorer	highscore
0	False	False	False	False	False	False \
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
..
69	False	False	False	False	False	False
70	False	False	False	False	False	False
71	False	False	False	False	False	False
72	False	False	False	False	False	False
73	False	False	False	False	False	False

	best_bowling	best_bowling_figure
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
..
69	False	False
70	False	False
71	False	False
72	False	False
73	False	False

[74 rows x 20 columns]

```
[8]: df.isnull().sum()
```

```
[8]: match_id      0
     date          0
```

```
venue          0
team1          0
team2          0
stage          0
toss_winner    0
toss_decision  0
first_ings_score 0
first_ings_wkts 0
second_ings_score 0
second_ings_wkts 0
match_winner   0
won_by         0
margin         0
player_of_the_match 0
top_scorer     0
highscore      0
best_bowling   0
best_bowling_figure 0
dtype: int64
```

1.0.6 Highscore

```
[9]: df["highscore"].mean()
```

```
[9]: 71.71621621621621
```

```
[10]: plt.figure(figsize=(8, 6))

# Plot histogram with seaborn
sns.histplot(x=df["highscore"], data=df, bins=25, kde=True, color='skyblue',
             edgecolor='black')

# Add vertical line for mean
mean_highscore = df["highscore"].mean()
plt.axvline(mean_highscore, color='red', linestyle='--', label=f'Mean Highscore:
             {mean_highscore:.2f}')

# Add title and labels with larger font sizes
plt.title("Distribution of Highscores in IPL 2022", fontsize=16)
plt.xlabel("Highscore", fontsize=14)
plt.ylabel("Frequency", fontsize=14)

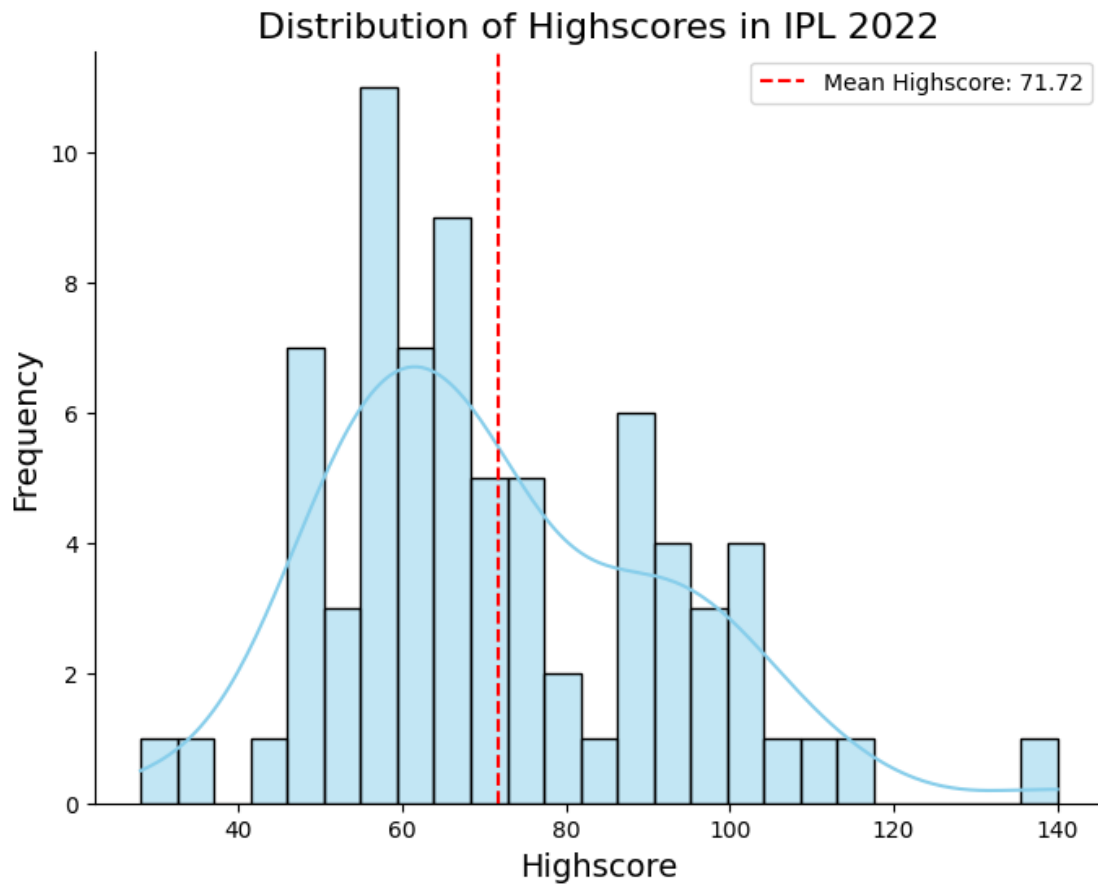
# Add legend
plt.legend()

# Remove top and right spines
sns.despine()
```



```
# Save plot
plt.savefig('distribution_of_highscores_in_ipl_2022.png', bbox_inches='tight')

# Show plot
plt.show()
```



The median highscore in IPL 2022 matches was 68.0. The histogram visualization provides an overview of the distribution of highscores, showing the frequency of different highscore ranges in the tournament. This can be helpful in understanding the typical level of runs scored in matches during the IPL 2022 season and identifying any patterns or outliers in the data.

1.0.7 Preferred Toss Decisions

```
[11]: df['toss_decision'].value_counts()
```

```
[11]: toss_decision
Field    59
Bat      15
```

Name: count, dtype: int64

```
[12]: plt.figure(figsize=(8, 6))

# Plot countplot with seaborn
ax = sns.countplot(x='toss_decision', data=df, palette='Set2')

# Add title and labels with larger font sizes
plt.title("Preferred Toss Decisions", fontsize=16)
plt.xlabel("Toss Decision", fontsize=14)
plt.ylabel("Count", fontsize=14)

# Add value counts on top of each bar
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.
    ↪get_height()), ha='center', va='center', fontsize=12, xytext=(0, 5),
    ↪textcoords='offset points')

# Remove top and right spines
sns.despine()

# Save plot
plt.savefig('preferred_toss_decisions.png', bbox_inches='tight')

# Show plot
plt.show()
```



In the IPL 2022 dataset, teams have predominantly chosen to field (bowl) first after winning the toss, with 59 instances, compared to only 15 instances where they chose to bat first. This suggests that in the majority of matches, teams opted to chase a target or exploit favorable bowling conditions rather than setting a target while batting first. The countplot visualization helps illustrate this preference for fielding as the preferred toss decision in the tournament.

1.0.8 Match Winner

```
[13]: df['match_winner'].value_counts()
```

```
[13]: match_winner
      Gujarat      12
      Rajasthan   10
      Banglore     9
      Lucknow      9
      Delhi        7
      Punjab       7
      Kolkata       6
      Hyderabad    6
```

```
Chennai      4
Mumbai       4
Name: count, dtype: int64
```

```
[14]: plt.figure(figsize=(12, 8)) # Increase figure size for better visualization

# Plot countplot with seaborn, ordering by the count of matches won
ax = sns.countplot(x='match_winner', data=df, order=df['match_winner'].
    ↪value_counts().index, palette='Set3')

# Add title and labels with larger font sizes
plt.title("Number of Matches Won by Each Team in IPL 2022", fontsize=16)
plt.xlabel("Team", fontsize=14)
plt.ylabel("Count", fontsize=14)

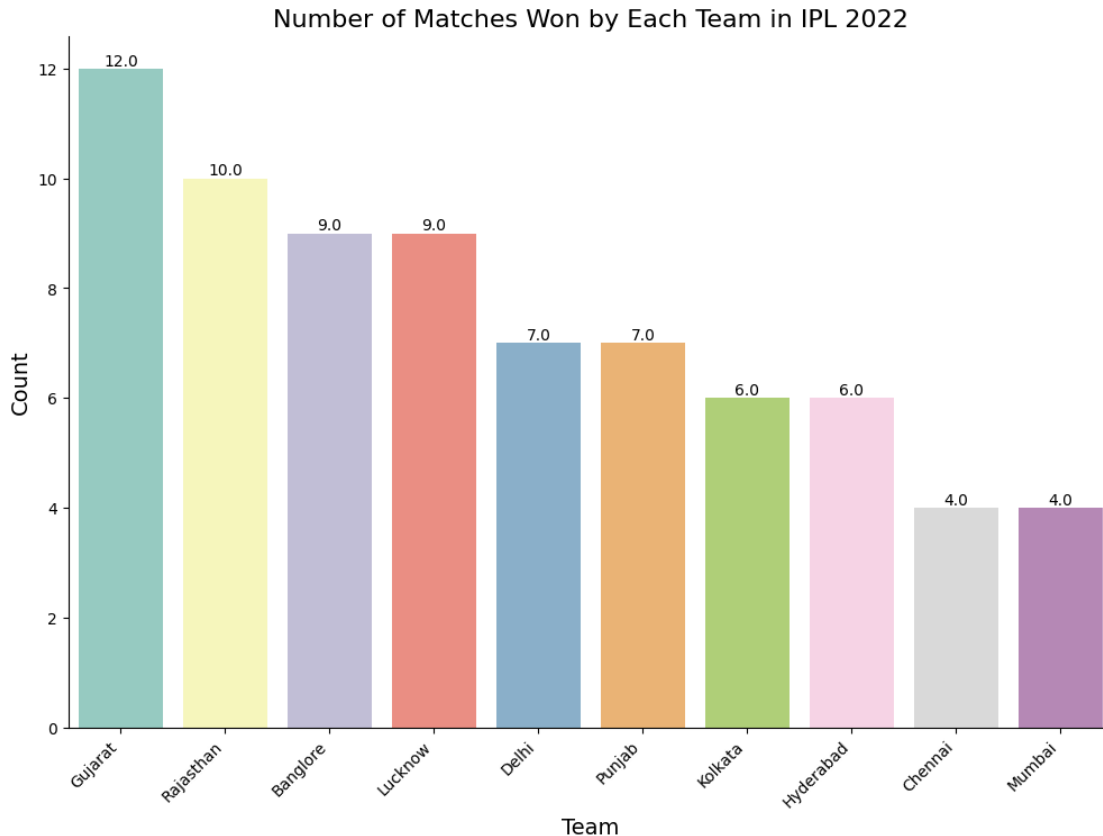
# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Add value counts on top of each bar
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.
    ↪get_height()), ha='center', va='center', fontsize=10, xytext=(0, 5), ↪
    ↪textcoords='offset points')

# Remove top and right spines
sns.despine()

# Save plot
plt.savefig('number_of_matches_won_by_each_team_in_ipl_2022.png', ↪
    ↪bbox_inches='tight')

# Show plot
plt.show()
```



In the IPL 2022 season, the team “Gujarat” won the highest number of matches, with 12 victories, making them the most successful team in terms of match wins. “Rajasthan” follows closely with 10 wins, while “Banglore” and “Lucknow” both secured 9 wins each. On the other hand, “Chennai” and “Mumbai” had the lowest number of wins, each with 4 victories.

The countplot visualization effectively displays the distribution of match wins among the teams in the tournament, highlighting the varying levels of success for different franchises in IPL 2022.

1.0.9 Top Scorers

```
[15]: df[["highscore" , "top_scorer"]].sort_values(by='highscore', ascending=False)
```

```
[15]:
```

	highscore	top_scorer
65	140	Quinton de Kock
33	116	Jos Buttler
71	112	Rajat Patidar
72	106	Jos Buttler
29	103	Jos Buttler
..
35	47	Abhishek Sharma
41	46	Quinton de Kock

73	45	Shubman Gill
58	36	MS Dhoni
5	28	Sherfane Rutherford

[74 rows x 2 columns]

```
[16]: plt.figure(figsize=(12, 8)) # Increase figure size for better visualization

# Plot barplot with seaborn using 'top_scorer' as x-axis and 'highscore' as
# y-axis
sns.barplot(x='top_scorer', y='highscore', data=df, palette='viridis')

# Add title and labels with larger font sizes
plt.title("Top Scorers and Their High Scores in IPL 2022", fontsize=16)
plt.xlabel("Top Scorers", fontsize=14)
plt.ylabel("High Score", fontsize=14)

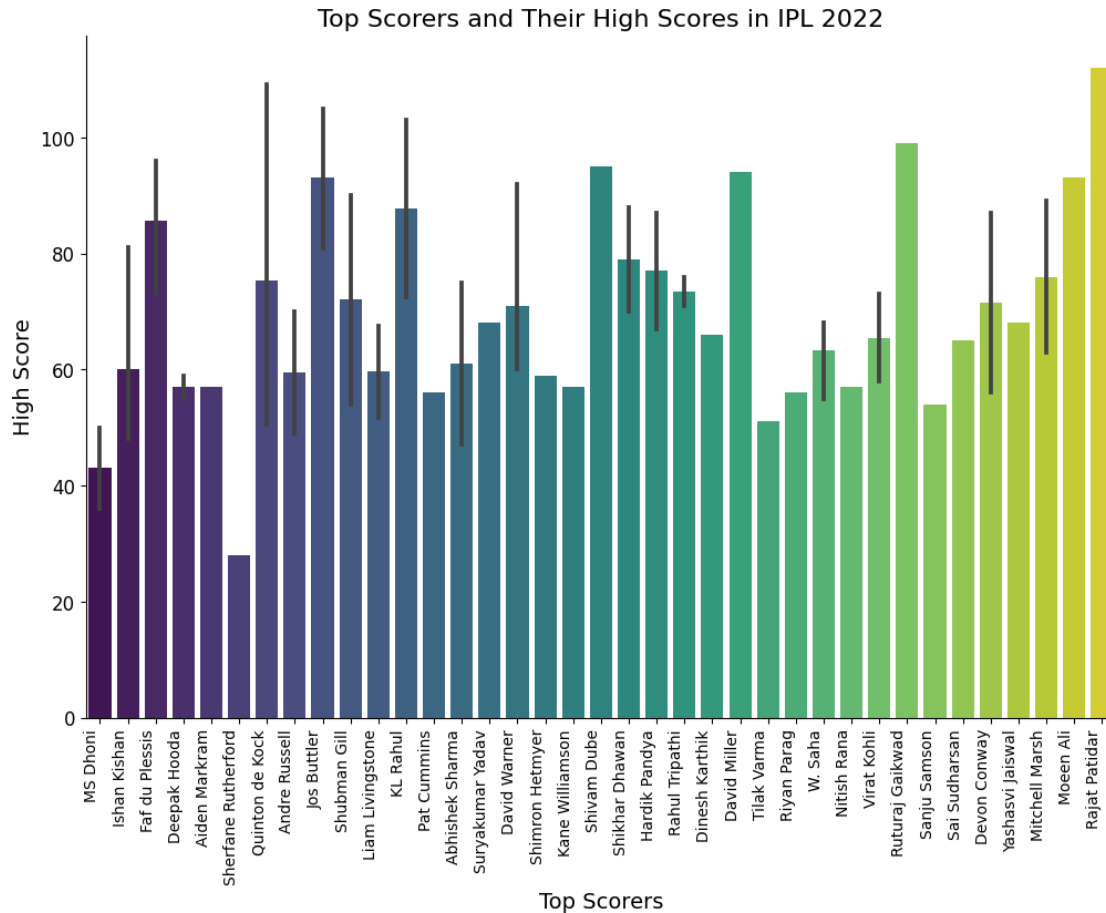
# Rotate x-axis labels for better readability
plt.xticks(rotation=90, ha='right')

# Increase font size of y-axis ticks for better readability
plt.yticks(fontsize=12)

# Remove top and right spines
sns.despine()

# Save plot
plt.savefig('top_scorers_and_their_highscores_in_ipl_2022.png',
           bbox_inches='tight')

# Show plot
plt.show()
```



In IPL 2022, Quinton de Kock achieved the highest individual highscore of 140 runs, followed by Jos Buttler with 116 runs and Rajat Patidar with 112 runs. This indicates that Quinton de Kock delivered an outstanding performance with the bat, leading the list of top scorers for the season.

The bar chart visualization further illustrates the highscores of the top scorers in IPL 2022. It shows the relationship between the players (on the x-axis) and their respective highscores (on the y-axis), with each bar's color indicating the highscore achieved.

1.0.10 Top Bowlers

```
[17]: df['best_bowling'].value_counts()
```

```
[17]: best_bowling
Yuzvendra Chahal      5
Rashid Khan           4
T Natarajan           3
Kagiso Rabada         3
Jasprit Bumrah        3
Kuldeep Yadav         3
```

Avesh Khan	3
Josh Hazlewood	3
Dwayne Bravo	2
Mohsin Khan	2
Andre Russell	2
Prasidh Krishna	2
Daniel Sams	2
Umrان Malik	2
Maheesh Theekshana	2
Hardik Pandya	2
Harshal Patel	2
Lockie Ferguson	2
Ravi Bishnoi	2
Wanindu Hasaranga	2
Mohammed Shami	2
Mukesh Choudhary	1
Harpreet Brar	1
Prashant Solanki	1
Ramandeep Singh	1
Shardul Takur	1
Trent Boult	1
Chetan Sakariya	1
Moeen Ali	1
Khaleel Ahmed	1
Tim Southee	1
Rilley Meredith	1
Mohammed Siraj	1
Washington Sundar	1
Pradeep Sangwan	1
Kuldeep Sen	1
Krunal Pandya	1
Umesh Yadav	1
Rahul Chahar	1
Murugan Ashwin	1
Odean Smith	1
Axar Patel	1

Name: count, dtype: int64

```
[18]: # Get the top 6 bowlers with the most occurrences of best bowling figures
best_bowlers = df['best_bowling'].value_counts().nlargest(6)

plt.figure(figsize=(12, 8)) # Increase figure size for better visualization

# Plot barplot with seaborn using 'best_bowling' as y-axis and 'count' as x-axis
sns.barplot(y=best_bowlers.values, x=best_bowlers.index, palette='viridis')

# Add title and labels with larger font sizes
```



```

plt.title("Top 6 Bowlers in IPL 2022", fontsize=16)
plt.xlabel("Best Bowling Figures", fontsize=14)
plt.ylabel("Count", fontsize=14)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

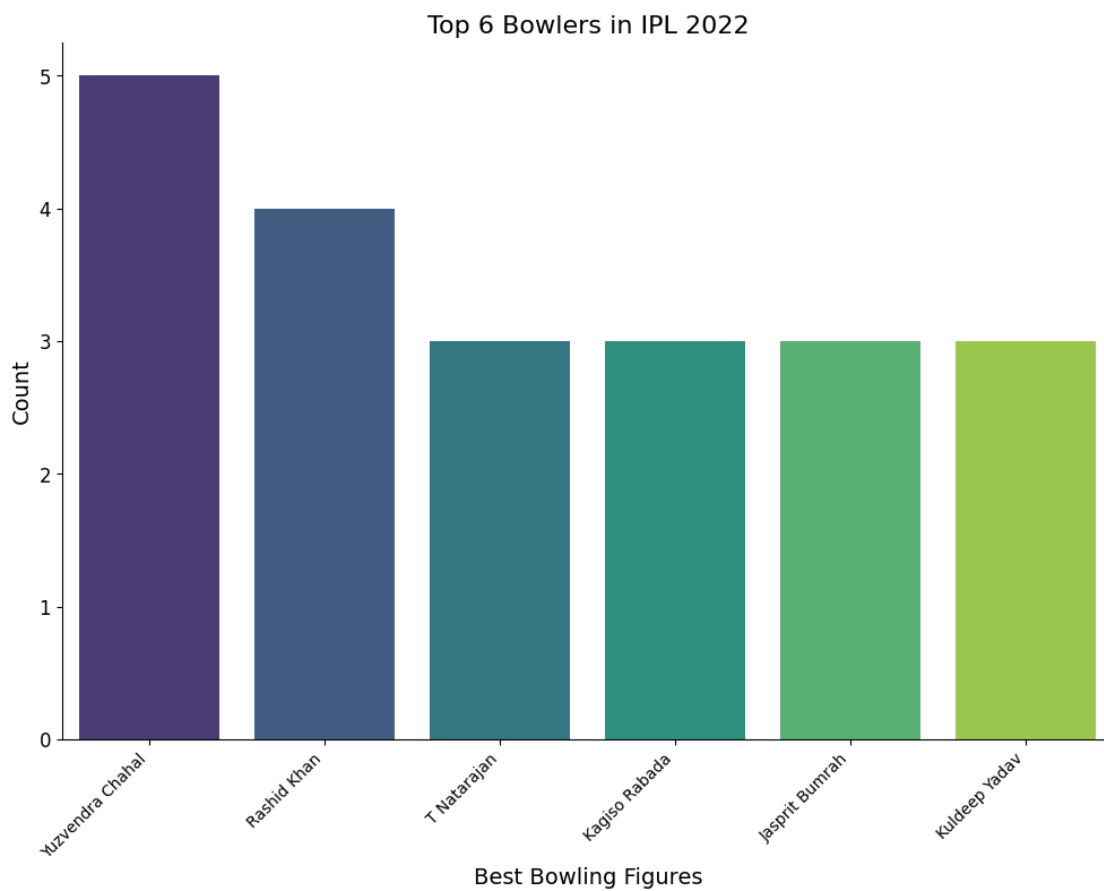
# Increase font size of y-axis ticks for better readability
plt.yticks(fontsize=12)

# Remove top and right spines
sns.despine()

# Save plot
plt.savefig('top_6_bowlers_in_ipl_2022.png', bbox_inches='tight')

# Show plot
plt.show()

```



In IPL 2022, Yuzvendra Chahal emerged as the top bowler with the most “best bowling” performances, achieving this distinction 5 times during the tournament. Following him, several bowlers including Rashid Khan, T Natarajan, Kagiso Rabada, Jasprit Bumrah, Kuldeep Yadav, Avesh Khan, Josh Hazlewood, and others also had notable “best bowling” performances, with each of them achieving this feat 3 or more times.

The bar chart visualization highlights the top 6 bowlers in terms of their “best bowling” performances. It showcases the frequency with which these bowlers delivered outstanding performances during the IPL 2022 season.

1.0.11 Player of the match

```
[19]: df['player_of_the_match'].value_counts()
```

```
[19]: player_of_the_match
```

Kuldeep Yadav	4
Jos Buttler	3
Umesh Yadav	2
Quinton de Kock	2
David Miller	2
Umrans Malik	2
K L Rahul	2
Rahul Tripathi	2
Hardik Pandya	2
Jasprit Bumrah	2
Shubman Gill	2
Yuzvendra Chahal	2
Wanindu Hasaranga	2
Avesh Khan	2
Dinesh Karthik	2
Virat Kohli	1
Rinku Singh	1
Kagiso Rabada	1
Harshal Patel	1
David Warner	1
Harpreet Brar	1
Tim David	1
Yashasvi Jaiswal	1
R Ashwin	1
Andre Russell	1
Shardul Thakur	1
Trent Boult	1
Devon Conway	1
W. Saha	1
Mitchell Marsh	1
Mohsin Khan	1
Daniel Sams	1
Jonny Bairstow	1

Ruturaj Gaikwad	1
Rashid Khan	1
Suruakumar Yadav	1
Anuj Rawat	1
Odean Smith	1
Mohammed Shami	1
Sanju Samson	1
Evin Lewis	1
Lockie Ferguson	1
Liam Livingstone	1
Pat Cummins	1
Abhishek Sharma	1
Kane Williamson	1
Rahul Tewatia	1
Shivam Dube	1
Mayank Agarwal	1
Faf du Plessis	1
Mukesh Choudhary	1
Marco Jansen	1
Shikhar Dhawan	1
Riyan Parag	1
Krunal Pandya	1
Rajat Patidar	1

Name: count, dtype: int64

```
[20]: # Get the top 6 players with the most occurrences of player of the match awards
best_players = df['player_of_the_match'].value_counts().nlargest(6)

plt.figure(figsize=(12, 8)) # Increase figure size for better visualization

# Plot barplot with seaborn using 'player_of_the_match' as y-axis and 'count'
# as x-axis
sns.barplot(y=best_players.values, x=best_players.index, palette='viridis')

# Add title and labels with larger font sizes
plt.title("Top 6 Man of the Match in IPL 2022", fontsize=16)
plt.xlabel("Players", fontsize=14)
plt.ylabel("Count", fontsize=14)

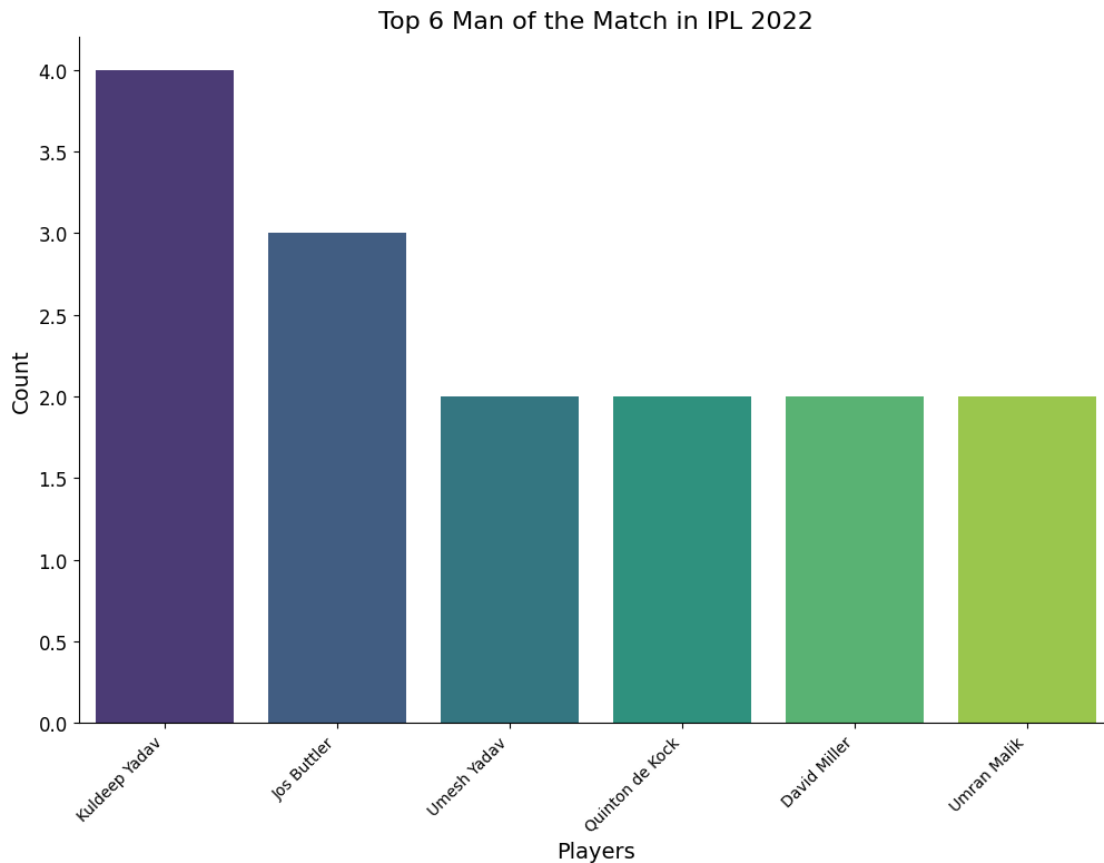
# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Increase font size of y-axis ticks for better readability
plt.yticks(fontsize=12)

# Remove top and right spines
sns.despine()
```

```
# Save plot
plt.savefig('top_6_man_of_the_match_in_ipl_2022.png', bbox_inches='tight')

# Show plot
plt.show()
```



In IPL 2022, Kuldeep Yadav was the player who received the “Man of the Match” award the most times, achieving it on 4 occasions. Several other players, including Jos Buttler, Umesh Yadav, Quinton de Kock, David Miller, Umran Malik, K L Rahul, Rahul Tripathi, Hardik Pandya, Jasprit Bumrah, Shubman Gill, Yuzvendra Chahal, Wanindu Hasaranga, Avesh Khan, and Dinesh Karthik, received the award multiple times as well.

The bar chart visualization highlights the top 6 players who received the “Man of the Match” award most frequently during IPL 2022.

1.0.12 Maximum Scored match in First Innings

```
[21]: df.loc[df['first_ings_score'].idxmax()]
```

```
[21]: match_id          34
      date              April 22,2022
      venue             Wankhede Stadium, Mumbai
      team1             Delhi
      team2             Rajasthan
      stage            Group
      toss_winner       Delhi
      toss_decision     Field
      first_ings_score  222
      first_ings_wkts   2
      second_ings_score 207
      second_ings_wkts  8
      match_winner     Rajasthan
      won_by            Runs
      margin            15
      player_of_the_match Jos Buttler
      top_scorer        Jos Buttler
      highscore         116
      best_bowling      Prasidh Krishna
      best_bowling_figure 3--22
      Name: 33, dtype: object
```

In match 34 of IPL 2022, held on April 22, 2022, at the Wankhede Stadium in Mumbai, the following key information stands out:

- The match was played between Delhi and Rajasthan in the Group stage.
- Delhi won the toss and chose to field.
- Rajasthan, batting first, scored an impressive 222 runs for the loss of 2 wickets.
- Delhi's second innings saw them score 207 runs but lose 8 wickets in the process.
- Rajasthan won the match by a margin of 15 runs.
- Jos Buttler was named the “Player of the Match” for his outstanding performance, scoring 116 runs, which was a significant contribution to Rajasthan’s total.

This information highlights a high-scoring and competitive match in which Rajasthan emerged victorious, largely thanks to Jos Buttler’s exceptional batting performance.

1.0.13 Minimum Scored match in First Innings

```
[22]: df.loc[df['first_ings_score'].idxmin()]
```

```
[22]: match_id          36
      date              April 23,2022
      venue             Brabourne Stadium, Mumbai
      team1             Bangalore
      team2             Hyderabad
      stage            Group
      toss_winner       Hyderabad
      toss_decision     Field
      first_ings_score  68
```

```

first_ings_wkts          10
second_ings_score        72
second_ings_wkts         1
match_winner              Hyderabad
won_by                    Wickets
margin                    9
player_of_the_match       Marco Jansen
top_scorer                Abhishek Sharma
highscore                 47
best_bowling              T Natarajan
best_bowling_figure       3--10
Name: 35, dtype: object

```

In match 36 of IPL 2022, which took place on April 23, 2022, at the Brabourne Stadium in Mumbai, the following key information stands out:

- The match was contested between Bangalore and Hyderabad in the Group stage.
- Hyderabad won the toss and elected to field.
- Bangalore's first innings yielded a very low score of 68 runs, with all their wickets (10) falling.
- In response, Hyderabad chased down the target of 68 runs by losing just 1 wicket and scored 72 runs.
- Hyderabad won the match by 9 wickets, demonstrating a dominant performance in both batting and bowling.
- Marco Jansen was named the "Player of the Match," and T Natarajan delivered the best bowling performance with figures of 3 wickets for 10 runs.

This information highlights a one-sided match where Hyderabad secured a comfortable victory over Bangalore, primarily due to their strong bowling performance and efficient chase.

```

[23]: from wordcloud import WordCloud

man_of_the_match_players = df['player_of_the_match']

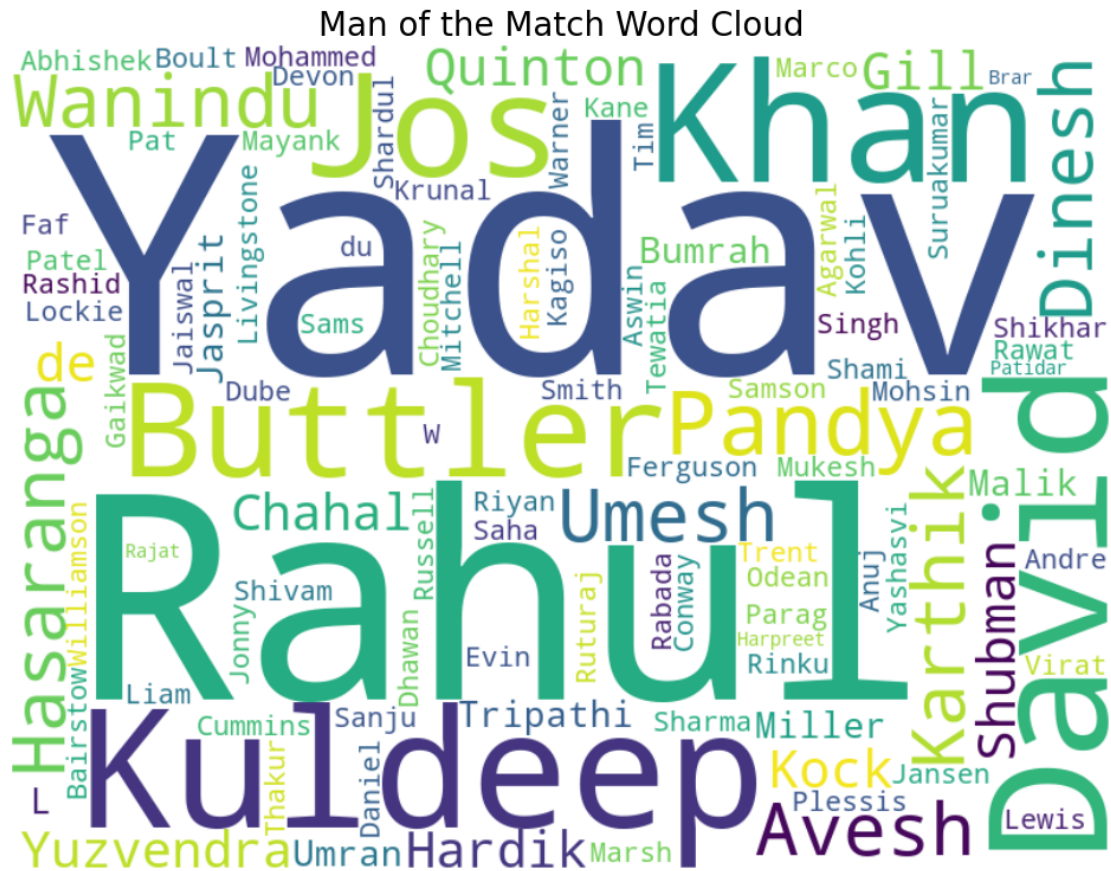
wordcloud = WordCloud(width=800, height=600, background_color='white',
    ↳ colormap='viridis', max_words=500, contour_color='steelblue').generate(' '.
    ↳ join(man_of_the_match_players))

plt.figure(figsize=(12, 10)) # Increase figure size for better visualization
plt.imshow(wordcloud, interpolation='bilinear')
plt.title('Man of the Match Word Cloud', fontsize=20)
plt.axis('off')

# Save plot
plt.savefig('man_of_the_match_wordcloud.png', bbox_inches='tight')

plt.show()

```



The “**Man of the Match Word Cloud**” visualization presents the names of players who received the “Man of the Match” award in IPL 2022. The size of each player’s name in the word cloud corresponds to the frequency with which they were awarded this honor.

This visualization can be used to quickly identify the most frequently recognized and standout players in the tournament, showcasing the individuals who made significant contributions and had outstanding performances throughout IPL 2022.

[]: