Data Analysis & Visualisation
On
IPL-Indian Premier League

Submitted by:

Aishwarya Patel

3rd Semester, B.Sc(H) Computer Science

Department of Computer Science

P.G.D.A.V. College

University of Delhi

Date of Submission: December 10, 2023

IPL

The Indian Premier League (IPL) is a professional Twenty20 cricket league, contested by 8 teams based out of 8 different Indian cities. The league was founded by the Board of Control for Cricket in India (BCCI) in 2007. It is usually held between March and May of every year and has an exclusive window in the ICC Future Tours Programme.

Aim of this Project

To analyse and visualise our dataset(containing data of IPL matches played in the period of 2016 - 2019) and then draw some conclusion from it.

Source

In this data Analysis We will be using IPL Player Stats-2016 till 2019.csv which has been dowmloaded from *data.world*

LINK - https://data.world/cclayford/cricinfo-statsguru-data

Section 1- Installing the required libraries

In this data Analysis We will be using various Python Libraries such as pandas, Numpy, Seaborn & Matplotlib,so first we need to install all these libraries ,if we don't have in our system and import them here .

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline
```

Reading data of IPL Player Stats - 2016 till 2019.csv file using pandas library

```
ipl_df = pd.read_csv("C:\sem 3\DAV\project\IPL Player Stats - 2016
till 2019.csv")

ipl_df
```

- **To know about number of rows and columns of dataset we will use the .shape method**

```
a,b=ipl_df.shape

'Total rows=%s and Total columns=%s'%(a,b)

Out:'Total rows=631 and Total columns=29'
```

- **Now we will use .columns to check the names of all columns present in our dataset**

```
ipl_df.columns

Out: Index(['Team', 'Player', 'Tournament', 'Matches', 'Batting Innings
', 'Not Out',
```

```
       'Runs Scored', 'Highest Score', 'Batting Average', 'Balls Faced'
,
       'Batting Strike Rate', '100', '50', '0', '4s', '6s', 'Bowling In
nings',
       'Overs Bowled', 'Maidens Bowled', 'Runs Conceded', 'Wickets Take
n',
       'Best Bowling Figures', 'Bowling Average', 'Bowling Economy Rate
',
       'Bowling Strike Rate', '4+ Innings Wickets', '5+ Innings Wickets
',
       'Catches Taken', 'Stumpings Made'],
      dtype='object')
```

Section 2 - Data Cleaning and Data Preparation

In this project,we would not be using all the columns so, we will create a new data frame without making any changes to the initial data frame which will have the following columns-

- Team
- Player
- Tournament
- Matches
- Batting Innings
- Not Out
- Runs Scored
- Batting Average
- Balls Faced
- Batting Strike Rate
- Wickets Taken
- Bowling Average
- Bowling Economy Rate
- Bowling Strike Rate
- Catches Taken

```
ipl_df2 = pd.DataFrame(ipl_df,columns =
['Team','Player','Tournament','Matches','Batting Innings','Not
Out','Runs Scored','Batting Average','Balls Faced','Batting Strike
Rate','Wickets Taken','Bowling Average','Bowling Economy
Rate','Bowling Strike Rate','Catches Taken'])
```

```
ipl_df2
```

 # Now we again check name of all columns to test the above functionality.

```
ipl_df2.columns
```

 # Now let us see all Tournaments in which IPL is played.

#.unique() is use to see the unique values in a particular column.

```
ipl_df2["Tournament"].unique()
```

```
Out: array(['IPL 2016', 'IPL 2017', 'IPL 2018', 'IPL 2019'], dtype=obje
ct)
```

In "Tournament" column,no data cleaning is required,because it contain all tournament that had been played betweem 2016-2019

 # Now Let us see all the teams that have played in tournaments held between 2016-2019

```
ipl_df.Team.unique()
```

```
Out: array(['Delhi Daredevils', 'Kings XI Punjab', 'Kolkata Knight Ride
rs',
       'Mumbai Indians', 'Royal Challengers Bangalore',
       'Sunrisers Hyderabad', 'Gujarat Lions', 'Rising Pune Supergiants
',
       'Rising Pune Supergiant', 'Chennai Super Kings',
       'Rajasthan Royals', 'Delhi Capitals'], dtype=object)
```

 # Now Let us see all the players that have played in tournaments held between 2016-2019

```
ipl_df.Player.unique()
```

From above observations ,some Data cleaning is required

#Pune was represented by two Team Names as 'Rising Pune Supergiant' & 'Rising Pune Supergiants' so as a convinience we will changes these with the recent team representing Pune 'Rising Pune Supergiant' in all columns involving this name i.e 'Team' column.

#Similarly 2nd Change is in team name of Delhi ,Earlier the team name for delhi was 'Delhi Daredevils' but later it was changed to 'Delhi Capitals' so we will replace the "delhi Daredevils' with 'Delhi Capitals'.

```
ipl_df2.Team.replace({'Rising Pune Supergiants' : 'Rising Pune
Supergiant', 'Delhi Daredevils':'Delhi Capitals'},inplace=True)
```

Now check the changes in column Team.

```
ipl_df2.Team.unique()
```

So We had successfully cleaned our messy or misspelled data.

#Use of .isna() is to set NaN values of each column to True and use of .sum() is to calculate total NaN present

#in each column

```
ipl_df2.isna().sum()
```

In this data we don't have any NaN value but we have dash(-) present in some columns, to find the names of columns which contain dash we will use

```
columns_with_dashes = ipl_df2.columns[(ipl_df2.applymap(lambda x:
isinstance(x, str) and '-' in x)).any()]

columns_with_dashes
```

Now we will use .replace() to replace dash(-) with zero for our convenience

```
ipl_df2.replace('-',0,inplace=True)
```

#To examine the above change use .head()

```
ipl_df2.head(50)
```

#Here we will use .info() to get the Data type of each columns

```
ipl_df2.info()
```

```
Out: <class 'pandas.core.frame.DataFrame'>
RangeIndex: 631 entries, 0 to 630
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Team                  631 non-null    object
 1   Player                631 non-null    object
 2   Tournament            631 non-null    object
 3   Matches               631 non-null    int64
 4   Batting Innings       631 non-null    object
 5   Not Out               631 non-null    object
 6   Runs Scored           631 non-null    object
 7   Batting Average       631 non-null    object
 8   Balls Faced           631 non-null    object
 9   Batting Strike Rate   631 non-null    object
 10  Wickets Taken         631 non-null    object
 11  Bowling Average       631 non-null    object
 12  Bowling Economy Rate  631 non-null    object
 13  Bowling Strike Rate   631 non-null    object
 14  Catches Taken         631 non-null    int64
dtypes: int64(2), object(13)
memory usage: 74.1+ KB
```

#Here the Dtype of columns – 'Batting','Innings','Runs Scored','Batting Average','Balls Faced','Batting Strike Rate','Wickets Taken','Bowling Average','Bowling Economy Rate','Bowling Strike Rate' is [object]so,we need to convert it to [int64] type for our further calculation in this project so we will use .astype() and .round() to convert it upto 2 decimal places.

```
columns =['Matches', 'Batting Innings', 'Not Out',
       'Runs Scored','Batting Average', 'Balls Faced',
'Batting Strike Rate',
       'Wickets Taken', 'Bowling Average','Bowling Economy
Rate',
       'Bowling Strike Rate', 'Catches Taken']
ipl_df2[columns] =
ipl_df2[columns].astype(float).round().astype(int)
```

#Now we will see our changes

```
ipl_df2.head(40)
```

Now it is clear that no empty value is left in our data set So we are done with our data cleaning and data preparation part and can move to next step.

Section 3 - Exploratory Analysis

Now We Will Analyse the Data For Different types of Querie

#Here first of all we will do hierarchical indexing of column 'Tournament' and 'Team' to see data more clearly and for our better understanding by using se_index()

```
frame = ipl_df2.set_index(['Tournament','Team'],inplace=True)
```

#check changes

```
frame.head()
```

#We will see some statistics calculations on each row like max,min,mean,standard deviation etc within a column using describe()

```
ipl_df2.describe()
```

Section 4 - Asking Interesting Questions on data

As part of this data analysis, it is very crucial to raise question and find answer to them. Here we will try to find out some of the most essential questions, that will help us in drawing a major conclusion from our dataset.

I will be asking following Questions:

Q1). What is the total no of runs scored by each team in the IPL2017 tournament?Which Team has scored maximun run in the IPL2017 tournament?

Q2).What is the average batting strike rate for each player across all tournaments? Who has the lowest batting strike rate give name and value.

Q3). What is the overall bowling economy rate for each team?

Q4).What is the correlation between Batting Average and Bowling Average?Who has the highest batting average in IPL 2019?

Q5).Create a heatmap of the correlation matrix for numerical columns.

Q6).Create subplots showing the distribution of Runs Scored and Bowling Average.

Q7).Create a scatter plot between Wickets Taken and Bowling Economy Rate.

Q8). How many players have taken more than 20 wickets in a single match?

Q9).Create a bar chart showing the total number of catches taken by each team.

Q10).Calculate and display the top 3 players with the highest number of Not Out innings.

Q1). What is the total no of runs scored by each team in the IPL2017 tournament? Which Team has scored maximun run in the IPL2017 tournament?

#To find the run scores of teamns in IPL 2017 we will do .groupby Team and use .sum()

```
run_by_team = ipl_df2[ipl_df2['Tournament'] == 'IPL 2017'].groupby('Team')['Runs Scored'].sum()

run_by_team
```

#To know which team has scored maximum run in IPL 2017 we will use .idxmax() which will return the index i.e. Team

```
run_by_team.idxmax()
```

Q2).What is the average batting strike rate for each player across all tournaments?Who has the lowest batting strike rate give name and value.

#We will be using same as previous question groupby() and .mean() to get the average

```
avg_strike_rate = ipl_df2.groupby('Player')['Batting Strike Rate'].mean()
avg_strike_rate
```

#To know who has lowest batting strike rate we will be using .idxmin() and find its value by .min()

```
a = avg_strike_rate.idxmin()
b = avg_strike_rate.min()
'Lowest Batting striker is %s and his strike value is %s'%(a,b)
```

Q3). What is the overall bowling economy rate for each team?

Here we are using groupby() and mean to find it

```
bowling_economy_by_team = ipl_df2.groupby('Team')['Bowling Economy Rate'].mean()
bowling_economy_by_team
```

Q4).What is the correlation between Batting Average and Bowling Average? Who has the highest batting average in IPL 2019?

#Here we will use .corr() to find the relation between Batting Average and Bowling Average.

```
correlation_avg = ipl_df2['Batting
Average'].corr(ipl_df2['Bowling Average'])

correlation_avg
```

#nlargest() is used to retrieve the top N largest values from a particular column or a combination of columns.we will used this to find the player having highest batting average in IPL 2019

```
highest_avg_odi = ipl_df2[ipl_df2['Tournament'] == 'IPL
2019'].nlargest(1, 'Batting Average')

highest_avg_odi
```

Q5).Create a heatmap of the correlation matrix for numerical columns.

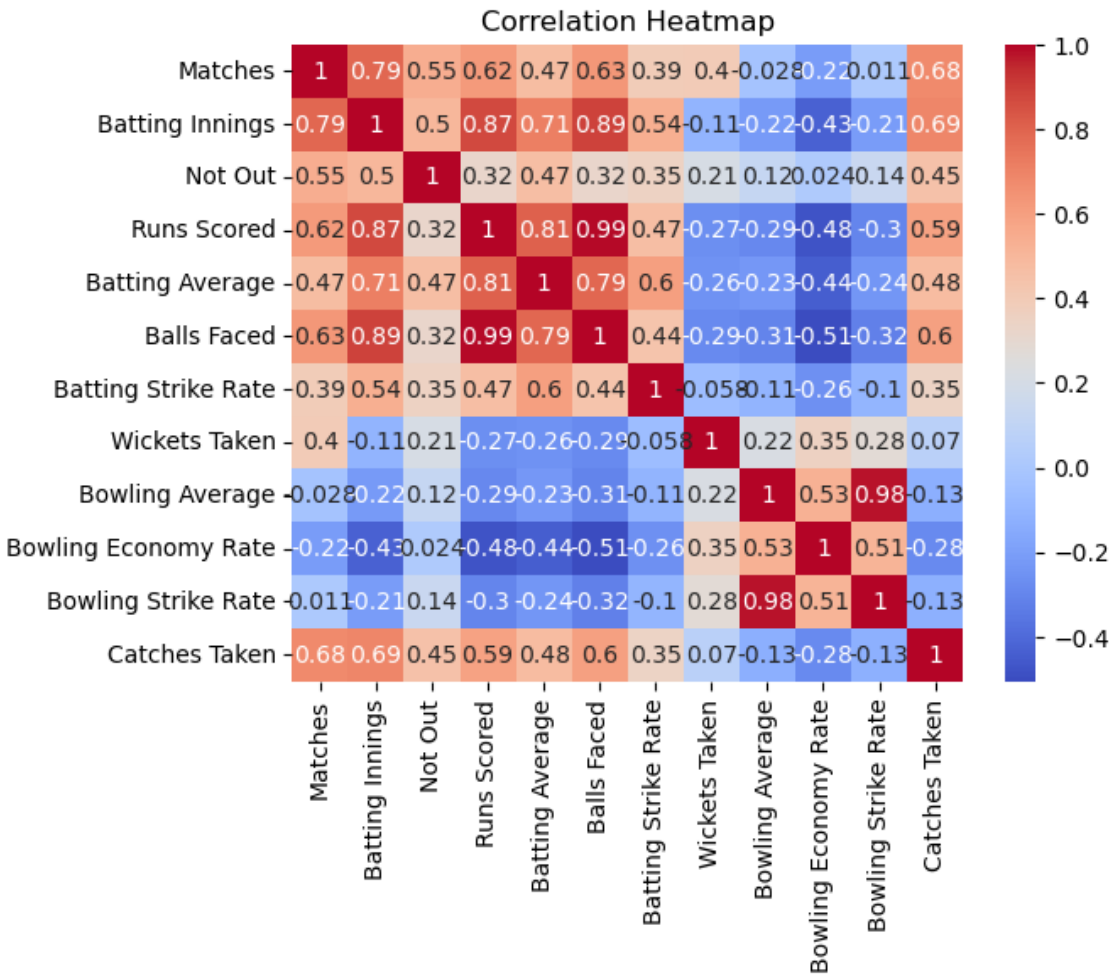
#Taking the use of matplotlib and seaborn to create a heatmap and giving it tille Correlation using .title() and .heatmap()

```
correlation_matrix = ipl_df2.corr()

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Heatmap')

plt.tight_layout()
```



Q6).Create subplots showing the distribution of Runs Scored and Bowling Average.

#using .subplot () to create a subplot in which we are using .histplot() for graph and setting their title by .set_title() , x lable by .set_xlabel() and y lable by .set_ylabel()

```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))


sns.histplot(ipl_df2['Runs Scored'], bins=20, kde=True,
ax=axes[0])
axes[0].set_title('Distribution of Runs Scored')
axes[0].set_xlabel('Runs Scored')
axes[0].set_ylabel('Frequency')
```

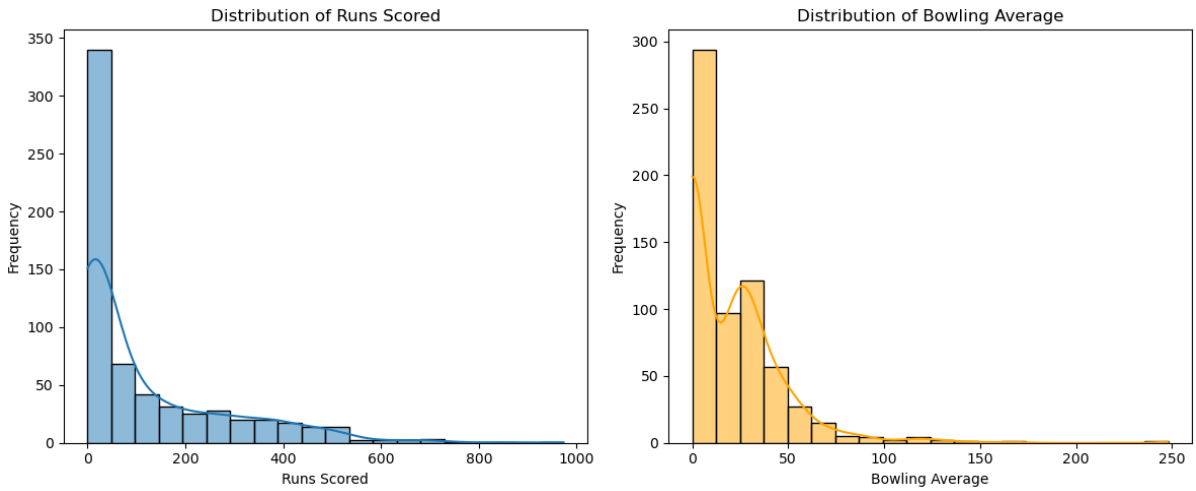
```
sns.histplot(ipl_df2['Bowling Average'], bins=20, kde=True,
ax=axes[1], color='orange')

axes[1].set_title('Distribution of Bowling Average')

axes[1].set_xlabel('Bowling Average')

axes[1].set_ylabel('Frequency')


plt.tight_layout()
```



Q7).Create a scatter plot between Wickets Taken and Bowling Economy Rate.

#Using .scatterplot() to create scatter plot between Wickets Taken and Bowling Economy Rate and write x lable and y lable by using .xtitle() and .ylable()

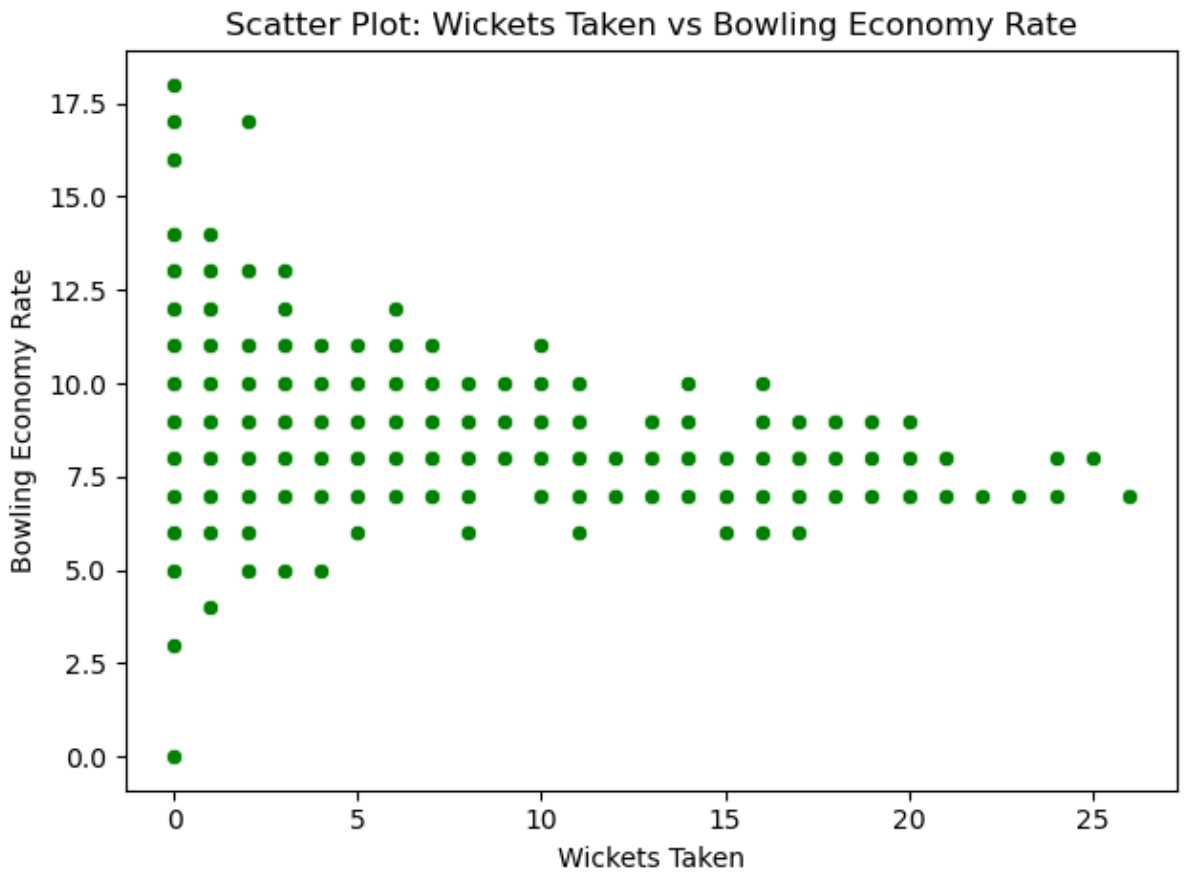
```
sns.scatterplot(x='Wickets Taken', y='Bowling Economy Rate',
data=ipl_df2, color='green')

plt.title('Scatter Plot: Wickets Taken vs Bowling Economy Rate')

plt.xlabel('Wickets Taken')

plt.ylabel('Bowling Economy Rate')

plt.tight_layout()
```



Q8). How many players have taken more than 20 wickets in a single match?

applying some conditions to .unique()

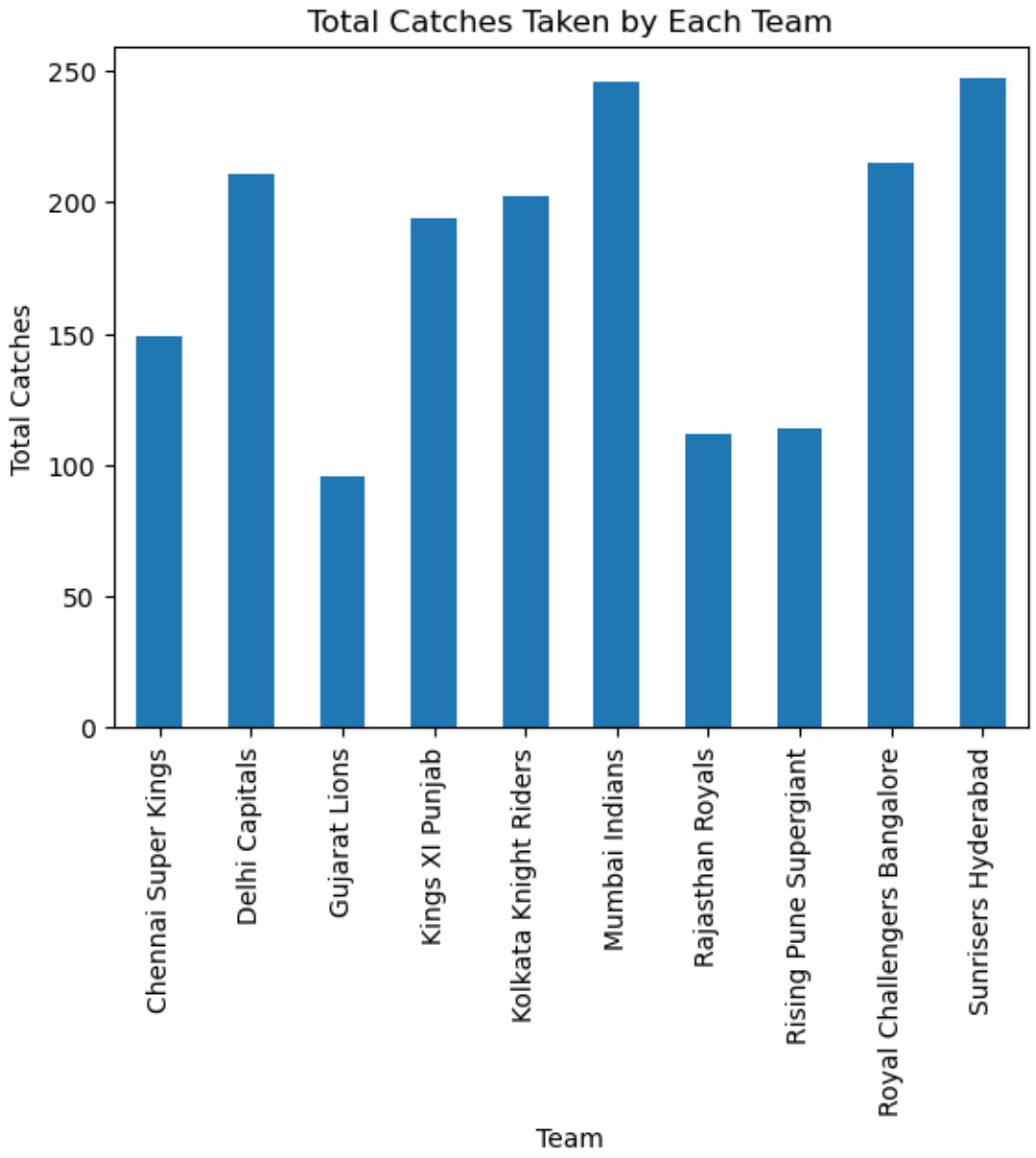
```
players_more_than_20_wickets = ipl_df2[ipl_df2['Wickets Taken'] >
20]['Player'].unique()
```

```
players_more_than_20_wickets
```

Q9).Create a bar chart showing the total number of catches taken by each team.

#Firstly grouping and adding them and plot graph by using .plot()

```
catches_by_team = ipl_df2.groupby('Team')['Catches Taken'].sum()
catches_by_team.plot(kind='bar')
plt.title('Total Catches Taken by Each Team')
plt.xlabel('Team')
plt.ylabel('Total Catches')
plt.show()
```



Q10).Calculate and display the top 3 players with the highest number of Not Out innings.

#nlargest() is used to retrieve the top N largest values from a particular column or a combination of columns.

```
top_not_out_players = ipl_df2.nlargest(3, 'Not Out')[['Player', 'Not Out']]
```

```
top_not_out_players
```

Section 5: Inferences and Conclusion

In this analysis I used the IPL Player Stats-2016 till 2019.csv file from the data.world . Following are my conclusions about it

1. We find that in IPL 2017 team ‘Mumbai Indians’ has scored maximum run then team ‘Royal Challengers Bangalore’ .
2. Some player like Ashish Reddy has large average batting strike rate while some have zero(lowest) average batting strike rate like AS Roy.
3. Batting strike rate shows that the batting performance of AS Roy is very poor.
4. From above observation we can conclude that the the Gujarat Lions has strong bowlers in their team ,which will greatly benefit them.
5. From the negative correlation we can conclude that as Batting Average is increaning Bowling Average tends to decrease and vice versa .
6. We found out that the Ms Dhoni player of Chennai Super Kings has the highest odi among all the players of all team.His performance was best in IPL 2019 where he played in total 15 maches ,he scored total 416 run out of 309 balls which shows his great potential in cricket and also has taken 11 catches during this year.
7. We can see from run scored graph there is large variation in scoring run and Bowling average.
8. We can see from scatter graph between Bowling Economy rate and Wicket take ,a large of wicket has been taken by players having Bowiling economy rate between 6 to 10
9. Total 9 players has taken wickets more than 20 in overall matches.
10. From the bar chart we can draw a conclusion that the team ‘Mumbai Indians’ and ‘Sunrisers Hyderabad’ has very strong fielding strategy as they have taken maximum number of catches during their matches while team ‘Gujarat

Lions' fielding strategy was not good enough, as they have taken lowest number of catches about 90.

11. MS Dhoni ,HH Pandya and Mandeep Singh has maximum number of not out innings where MS Dhoni and HH Pandya has total 9 not out and Mandeep Singh has 8 not out innings.