*** EDA Assignment-7 ****

Advanced Certification in

Applied Data Science

Machine Learning & IoT

By E&ICT Academy, IIT Guwahati

Name: Rajesh Bisht

Email Id: rbisht.india@gmail.com

Batch-8 , Git: https://github.com/RajeshBisht28/EDA_Assignment_7.git

```
In [ ]:
```

Question-1: What is the maximum number of matches played by an individual player in a season? Print the player name along with the number of matched played.

In [9]:

```
import pandas as pd
df = pd.read_csv('IPL_Assignment.csv')
player_innings = df[['Player', 'Matches']]
print(player_innings.head(15))
```

```
               Player  Matches
0            KL Rahul       14
1      Shikhar Dhawan       17
2        David Warner       16
3        Shreyas Iyer       17
4        Ishan Kishan       14
5        Quinton Kock       16
6    Suryakumar Yadav       16
7    Devdutt Padikkal       15
8         Virat Kohli       15
9        ABD Villiers       15
10      Faf Duplessis       13
11       Shubman Gill       14
12      Manish Pandey       16
13     Mayank Agarwal       11
14        Eoin Morgan       14
```

In [ ]:

```

```

Question-2: Top 2 players with maximum Average who have scored atleast 2 half centuries ?

In [13]:

```
import pandas as pd
df = pd.read_csv('IPL_Assignment.csv')
# Filter players with at least 2 half-centuries
filtered_df = df[df['50'] >= 2]
# Sort by average in descending order
sorted_df = filtered_df.sort_values(by='Avg', ascending=False)
# Select the top 2 players
top_2_players = sorted_df.head(2)
# Display the top 2 players
print(top_2_players[['Player', 'Avg', '50']])
```

```
               Player    Avg  50
36  Wriddhiman Saha  71.33   2
4      Ishan Kishan  57.33   4
```

In [ ]:

Question-3: Create 2 new columns based on Player name. First column will have first name and second column will have last name. Eg: for the player Shikhar Dhawan, Shikhar will be the first name and Dhawan will be the last name.

In [27]:

```
import pandas as pd
# Load the CSV file into a DataFrame
df = pd.read_csv('IPL_Assignment.csv')
# Split for First Name and create new column FirstName
df['FirstName'] = df['Player'].apply(lambda x: x.split()[0] if len(x.split()) > 1 else x)
# Split for Last Name and create new column LastName
df['LastName'] = df['Player'].apply(lambda x: x.split()[1] if len(x.split()) > 1 else '')
print(df[['FirstName', 'LastName']])
```

```
    FirstName LastName
0          KL    Rahul
1     Shikhar   Dhawan
2       David   Warner
3     Shreyas     Iyer
4       Ishan   Kishan
..        ...      ...
128   Khaleel    Ahmed
129  Arshdeep    Singh
130    Daniel     Sams
131 Shreevats  Goswami
132     Trent    Boult

[133 rows x 2 columns]
```

In [ ]:

Quetion-4: Create a new column (Cleaned_Highest_score) based on Highest score variable. Remove the Asterik(*) mark and convert the data type into INT.

In [28]:

```
import pandas as pd
# Read the CSV file into a DataFrame
df = pd.read_csv('IPL_Assignment.csv')
# Remove the asterisk (*) and convert to int
df['Cleaned_Highest_Score'] = df['Highest Score'].str.replace('*', '').astype(int)
# Display the updated DataFrame
print(df[['Highest Score', 'Cleaned_Highest_Score']])
```

```
    Highest Score  Cleaned_Highest_Score
0            132*                    132
1            106*                    106
2             85*                     85
3             88*                     88
4              99                     99
..            ...                    ...
128            0*                      0
129            0*                      0
130            0*                      0
131            0*                      0
132            0*                      0

[133 rows x 2 columns]
```

In [ ]:

Question-5: Print the total number of centuries scored in the entire season.

In [29]:

```
import pandas as pd
# Read the CSV file into a DataFrame
df = pd.read_csv('IPL_Assignment.csv')
# Sum of the total number of 100
total_100 = df['100'].sum()
# Print the total number of centuries
print(f'Total number of 100(S) scored: {total_100}')
```

```
Total number of 100(S) scored: 5
```

In [ ]:

Question-6: Print all the player names whose strike rate is less than the average strike rate of all players in entire season. Print the player name, his strike rate and average strike rate.

In [40]:

```python
import pandas as pd
df = pd.read_csv('IPL_Assignment.csv')
# Filter players where  Strike rate is less than Avg Strike rate
filtered_df = df[df['Avg'] > df['Strike rate']]

#Check if filtered data is Empty ?
if filtered_df.empty:
    print("No any player have less Strike rate than his/her Average Strike rate.")
else:
    print(filtered_df[['Player', 'Avg', 'Strike rate']])
```

```
No any player have less Strike rate than his/her Average Strike rate.
```

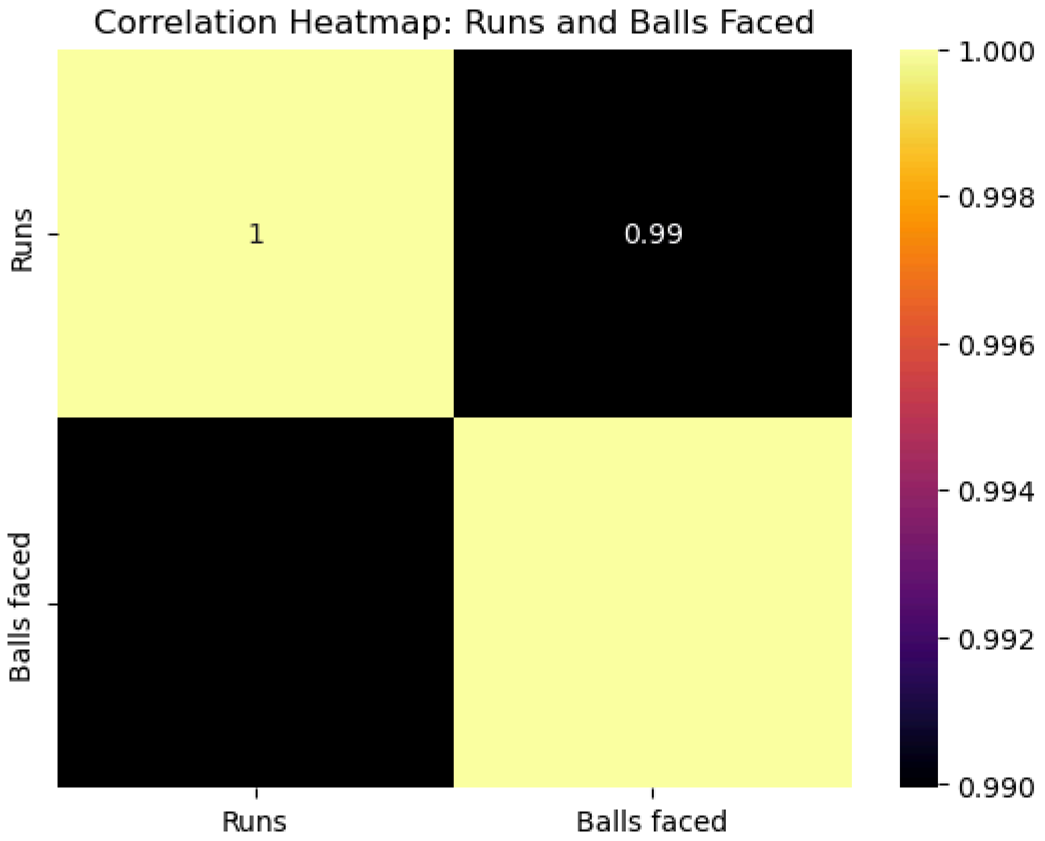
In [ ]:

Question-7: Please check the correlation between the features and create a heat map.

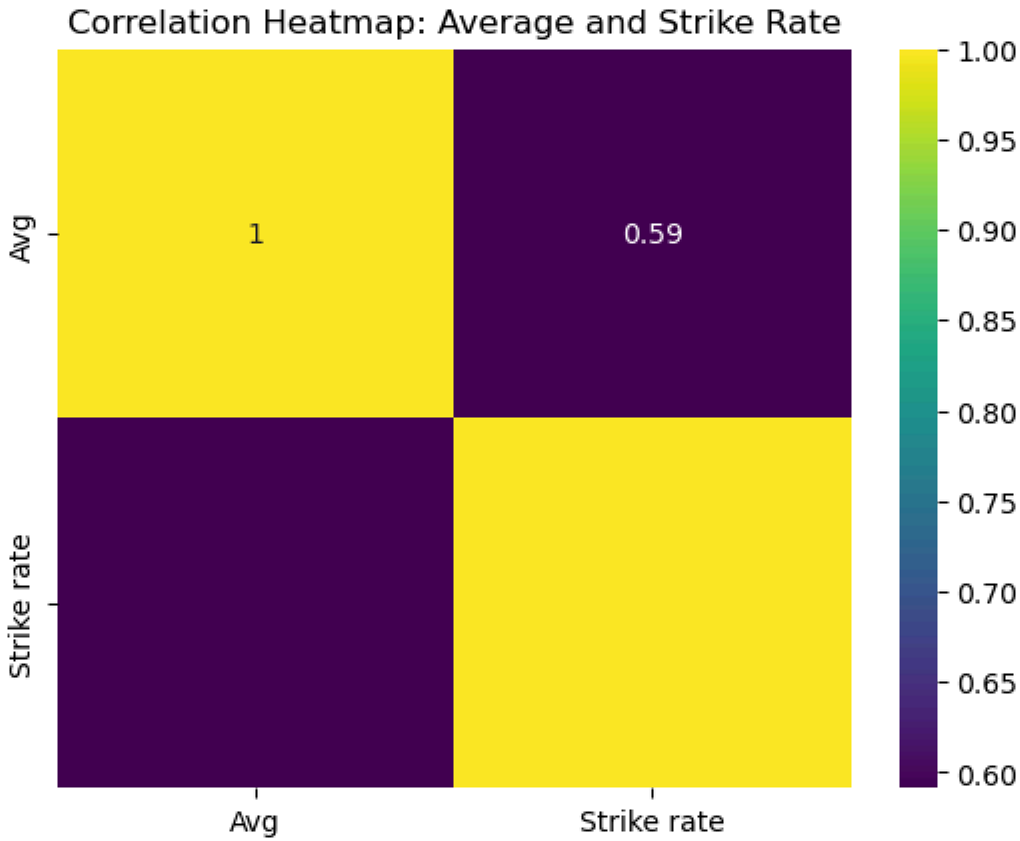
In [49]:

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the CSV file
df = pd.read_csv('IPL_Assignment.csv')

# Filter the Runs , Balls faced columns
df_filtered = df[['Runs', 'Balls faced']]

# Calculate the correlation matrix
correlation_matrix = df_filtered.corr()

# Plot the heatmap
sns.heatmap(correlation_matrix, annot=True, cmap='inferno')
plt.title('Correlation Heatmap: Runs and Balls Faced')
plt.show()
print("************************************************************************")
```

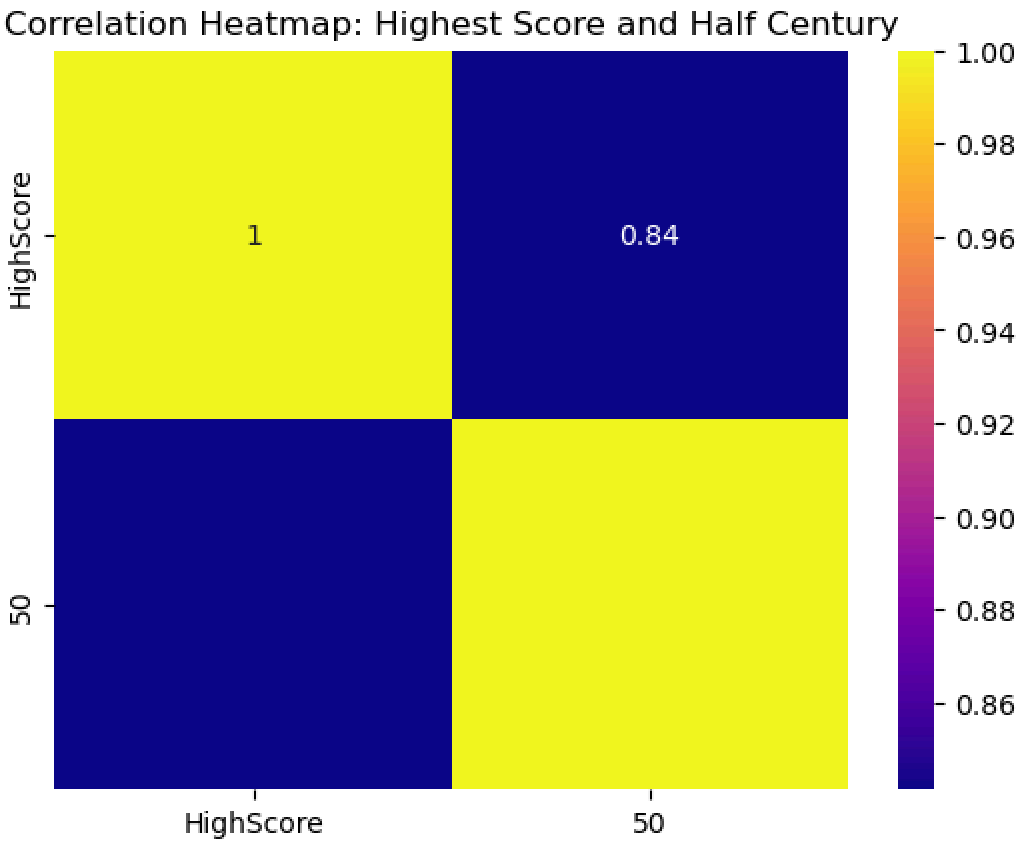
```python
# Filter the Avg , Strike Rate columns
df_filtered = df[['Avg', 'Strike rate']]

# Calculate the correlation matrix
correlation_matrix = df_filtered.corr()

# Plot the heatmap
sns.heatmap(correlation_matrix, annot=True, cmap='viridis')
plt.title('Correlation Heatmap: Average and Strike Rate')
plt.show()


print("***********************************************************************")
df['HighScore'] = df['Highest Score'].str.replace('*', '').astype(int)
# Filter the HighScore , 50 columns
df_filtered = df[['HighScore', '50']]

# Calculate the correlation matrix
correlation_matrix = df_filtered.corr()

# Plot the heatmap
sns.heatmap(correlation_matrix, annot=True, cmap='plasma')
plt.title('Correlation Heatmap: Highest Score and Half Century')
plt.show()
```


Correlation Heatmap: Runs and Balls Faced
Runs
Balls faced
1
0.99
Runs
Balls faced
1.000
0.998
0.996
0.994
0.992
0.990

****


Correlation Heatmap: Average and Strike Rate
Avg
Strike rate
1
0.59
Avg
Strike rate
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60

****



In [ ]:

Question-8: Check the list of players who has an average greater than 50 as well strike rate above 120. Print player name, average and strike rate.

In [52]:

```
import pandas as pd
df = pd.read_csv('IPL_Assignment.csv')
# Filter players  Avg Greater than 50 and Strike Rate above 120
filtered_df = df[(df['Avg'] > 50) & (df['Strike rate'] > 120)]

#Check if filtered data is Empty ?
if filtered_df.empty:
    print("No any player have less Strike rate than his/her Average Strike rate.")
```

```
else:
    print(filtered_df[['Player', 'Avg', 'Strike rate']])
```

```
             Player     Avg  Strike rate
0          KL Rahul   55.83       129.34
4      Ishan Kishan   57.33       145.76
31   Kieron Pollard   53.60       191.42
36  Wriddhiman Saha   71.33       139.86
37  Ruturaj Gaikwad   51.00       120.71
57     Deepak Hooda  101.00       142.25
60       Tom Curran   83.00       133.87
```

In [ ]:

```

```

Question-9: Please check the list of players who has an average greater than 40 and balls faced above 100. Print player name, average and balls faced.

In [53]:

```
import pandas as pd
df = pd.read_csv('IPL_Assignment.csv')
# Filter players where: average greater than 40 and balls faced above 100
filtered_df = df[(df['Avg'] > 40) & (df['Balls faced'] > 100)]

#Check if filtered data is Empty ?
if filtered_df.empty:
    print("No any player have less Strike rate than his/her Average Strike rate.")
else:
    print(filtered_df[['Player', 'Avg', 'Balls faced']])
```

```
              Player    Avg  Balls faced
0           KL Rahul  55.83          518
1     Shikhar Dhawan  44.14          427
4       Ishan Kishan  57.33          354
8        Virat Kohli  42.36          384
9       ABD Villiers  45.40          286
10     Faf Duplessis  40.81          319
14       Eoin Morgan  41.80          302
24   Kane Williamson  45.28          237
27       Chris Gayle  41.14          210
28        Ben Stokes  40.71          200
31    Kieron Pollard  53.60          140
32     Rahul Tewatia  42.50          183
33   Ravindra Jadeja  46.40          135
36   Wriddhiman Saha  71.33          153
37   Ruturaj Gaikwad  51.00          169
```

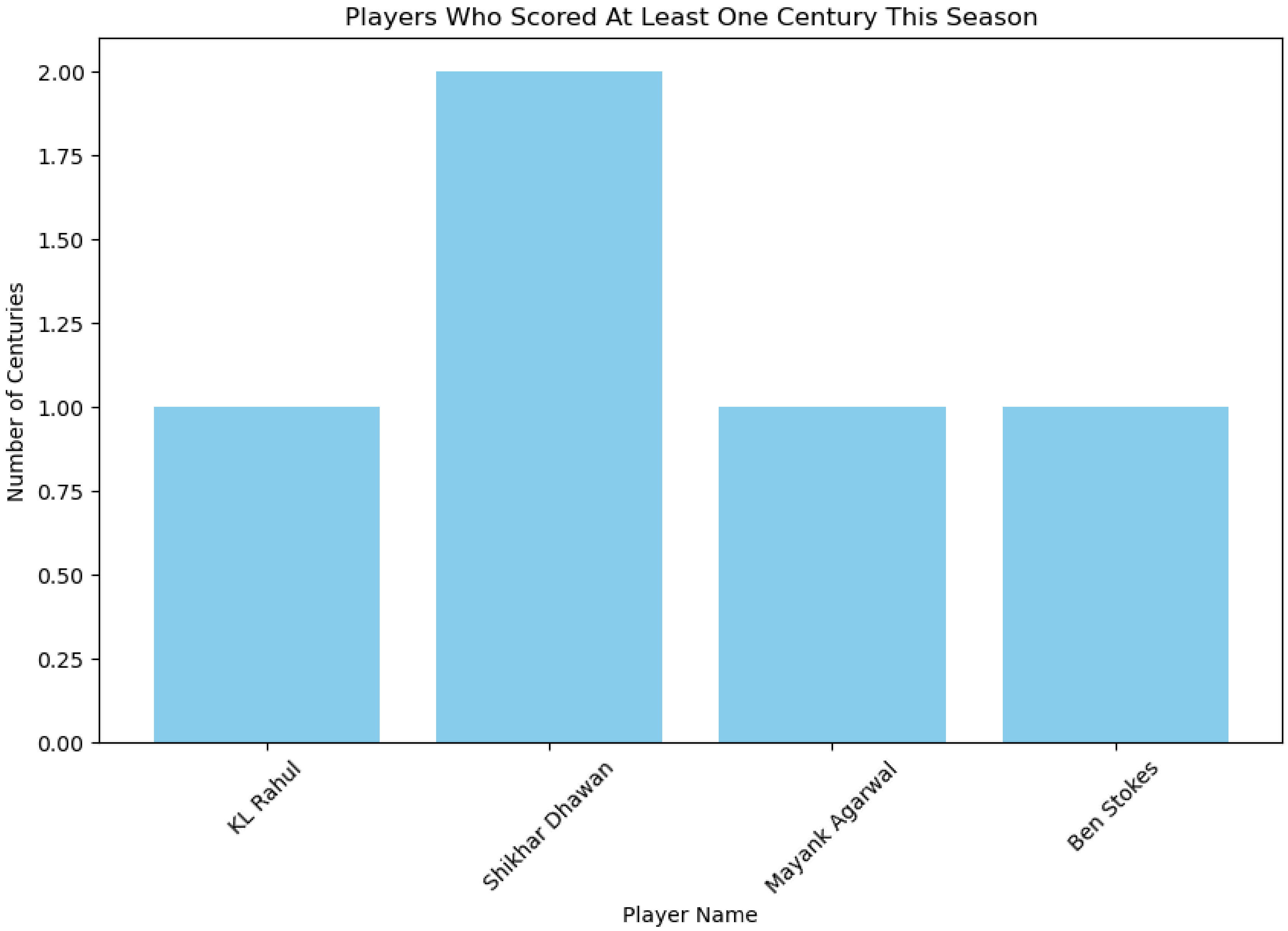
In [ ]:

```
```

Question-10: Players who scored atleast one century in this season. Create visualization.

In [61]:

```python
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('IPL_Assignment.csv')
# Players which have atleast one century
filtered_df = df[(df['100']>0)]
plt.figure(figsize=(10, 6))
plt.bar(filtered_df['Player'], filtered_df['100'], color='skyblue')
plt.xlabel('Player Name')
plt.ylabel('Number of Centuries')
plt.title('Players Who Scored At Least One Century This Season')
plt.xticks(rotation=45)
plt.show()
```



In [ ]:

Question-11: Players who scored atleast 4 half centuries in this season.

```
In [66]: import pandas as pd
         import matplotlib.pyplot as plt

         df = pd.read_csv('IPL_Assignment.csv')
         # Players which have atleast 4 half century
         filtered_df = df[(df['50']>=4)]
         print("Players which have atleast 4 half centuries.")
         print("**********************************************")
         print(filtered_df[['Player']])
```

```
Players which have atleast 4 half centuries.
**********************************************
              Player
0           KL Rahul
1     Shikhar Dhawan
2       David Warner
4       Ishan Kishan
5       Quinton Kock
6   Suryakumar Yadav
7   Devdutt Padikkal
9       ABD Villiers
10     Faf Duplessis
```

```
In [ ]:
```

Question-12: Check the list of players who hit more than 45 boundaries and more than 10 sixes in this season.

```
In [73]: import pandas as pd
         import matplotlib.pyplot as plt

         df = pd.read_csv('IPL_Assignment.csv')
         # print(df['4s'] + df['6s'])
         df['Boundaries'] = df['4s'] + df['6s']
         filtered_df = df[(df['Boundaries'] > 45) & (df['6s'] > 10)]
         # Players which have atleast 4 half century

         print("Players who hit more than 45 boundaries and more than 10 sixes in this season")
         print("*******************************************************************************")
         print(filtered_df[['Player', 'Boundaries', '6s']])
```

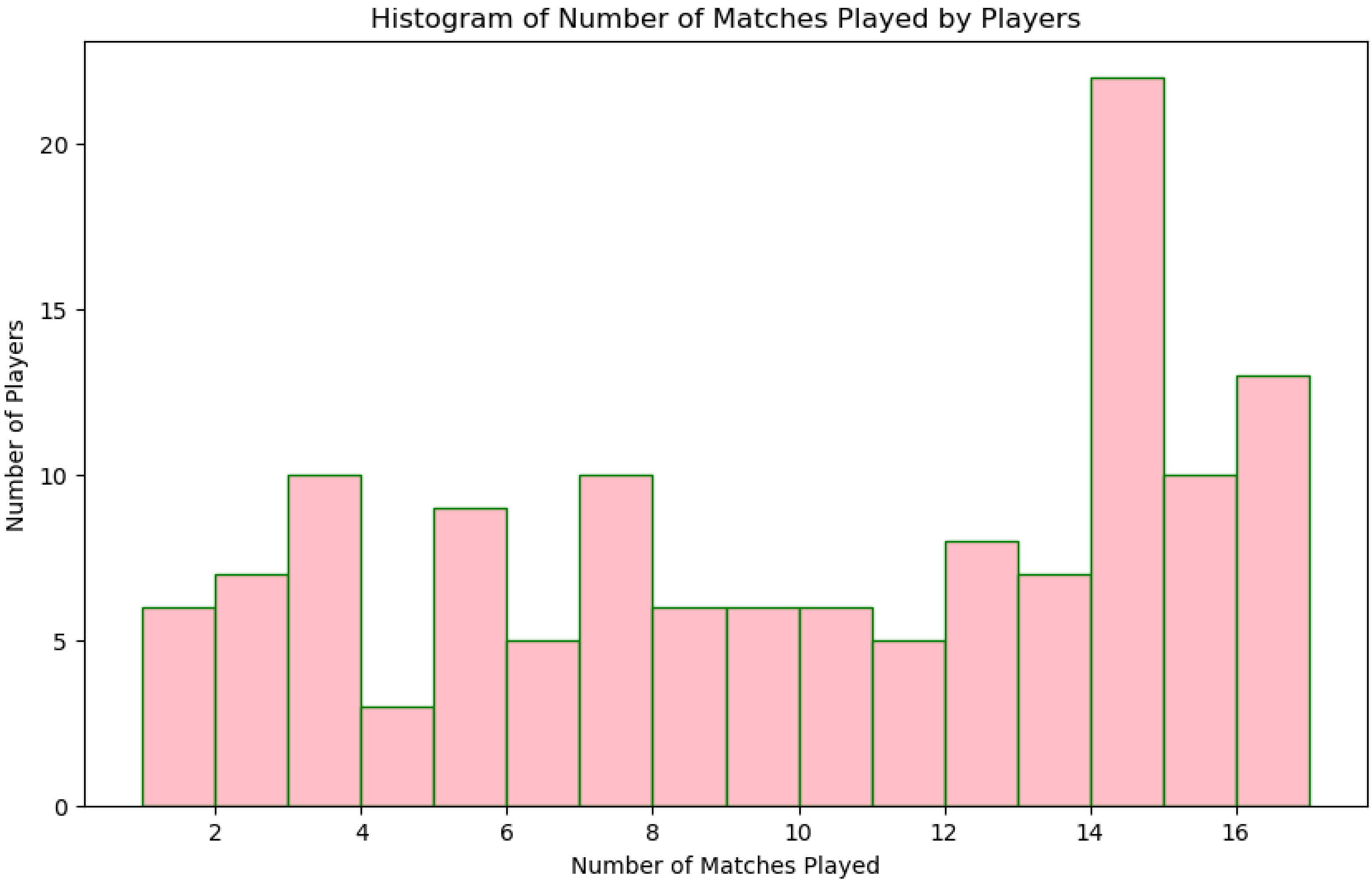
```
Players who hit more than 45 boundaries and more than 10 sixes in this season
********************************************************************************
              Player  Boundaries  6s
0           KL Rahul          81  23
1      Shikhar Dhawan         79  12
2        David Warner         66  14
3        Shreyas Iyer         56  16
4        Ishan Kishan         66  30
5        Quinton Kock         68  22
6    Suryakumar Yadav         72  11
9        ABD Villiers         56  23
10      Faf Duplessis         56  14
12      Manish Pandey         53  18
13     Mayank Agarwal         59  15
14        Eoin Morgan         56  24
15       Sanju Samson         47  26
17     Nicholas Pooran        48  25
18        Nitish Rana         55  12
19     Marcus Stoinis         47  16
22       Rohit Sharma         46  19
26       Shane Watson         46  13
```

In [ ]:

Question-13: Plot a histogram of number of matches played in a season by players

In [78]:

```python
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('IPL_Assignment.csv')
plt.figure(figsize=(10, 6))
bins_values = range(min(df['Matches']), max(df['Matches']) + 1, 1)
plt.hist(df['Matches'], bins=bins_values, color='pink', edgecolor='green')

plt.xlabel('Number of Matches Played')
plt.ylabel('Number of Players')
plt.title('Histogram of Number of Matches Played by Players')
plt.grid(False)
plt.show()
```



In [ ]:

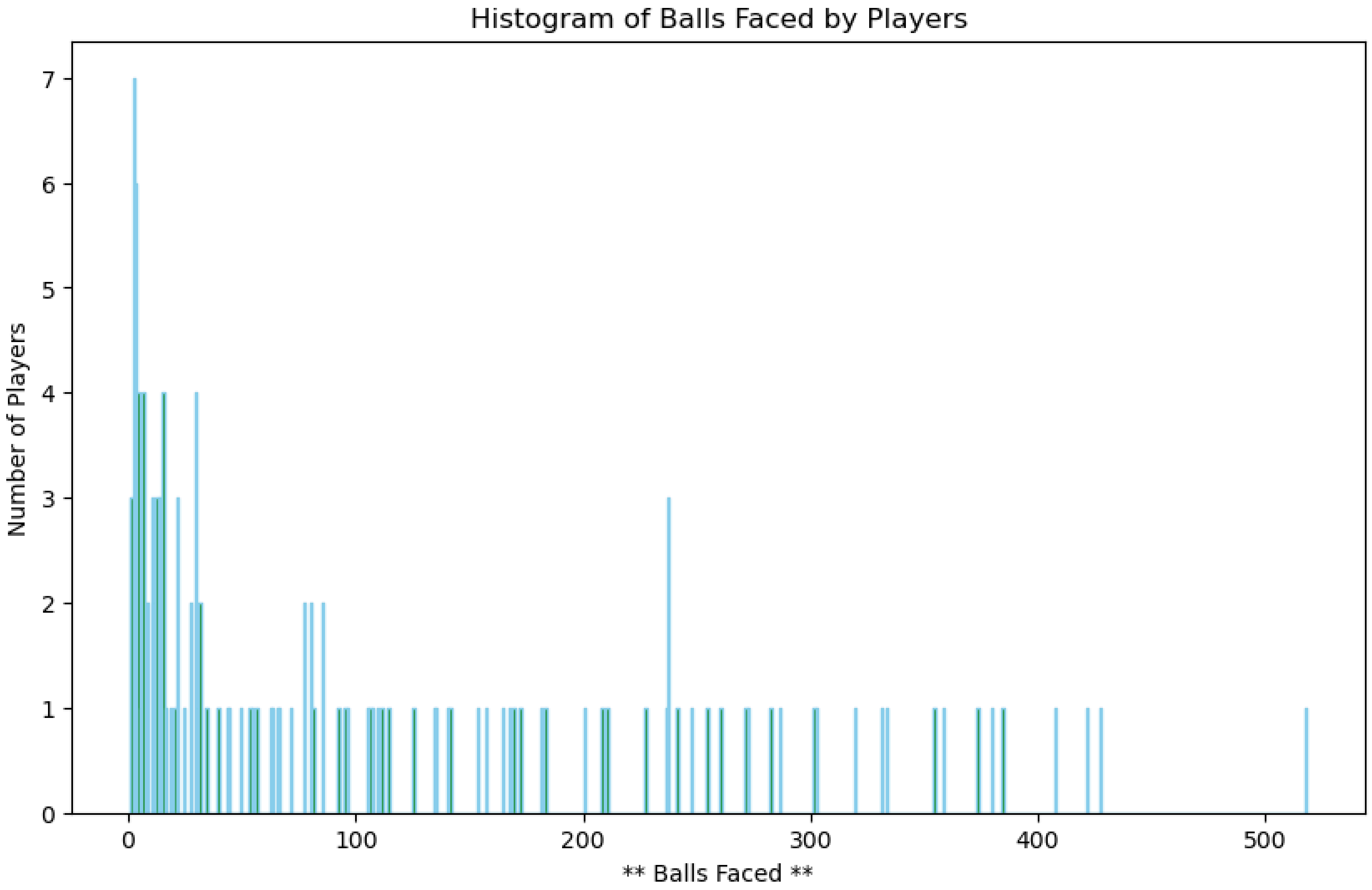
```

```

Question-14: Plot the histogram of balls faced by players.

In [79]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
df = pd.read_csv('IPL_Assignment.csv')
plt.figure(figsize=(10, 6))
bins_values = range(min(df['Balls faced']), max(df['Balls faced']) + 1, 1)
plt.hist(df['Balls faced'], bins=bins_values, color='green', edgecolor='skyblue')

plt.xlabel('** Balls Faced **')
plt.ylabel('Number of Players')
plt.title('Histogram of Balls Faced by Players')
plt.grid(False)
plt.show()
```



In [ ]:

Question-15: Top 10 players with most runs in a season.

In [82]:

```
import pandas as pd
df = pd.read_csv('IPL_Assignment.csv')
# Sort by run in descending order
```

```
sorted_df = df.sort_values(by='Runs', ascending=False)
# Select the top 10 players
top_players = sorted_df.head(10)
# Display the top 10 players
print("Top 10 players with most runs in a season.");
print("*************************************************")
print(top_players[['Player', 'Runs']])
```

```
Top 10 players with most runs in a season.
*************************************************
             Player  Runs
0          KL Rahul   670
1    Shikhar Dhawan   618
2      David Warner   548
3      Shreyas Iyer   519
4      Ishan Kishan   516
5      Quinton Kock   503
6  Suryakumar Yadav   480
7  Devdutt Padikkal   473
8       Virat Kohli   466
9      ABD Villiers   454
```

In [ ]:

Question-16: Print the players who played the match but didn't get the batting.

In [84]:

```
import pandas as pd
df = pd.read_csv('IPL_Assignment.csv')
# Player whoever not Faced any ball : Not did batting
filtered_df = df[df['Balls faced'] == 0]

#Check if filtered data is Empty ?
if filtered_df.empty:
    print("Not any players who played the match but didn't get the batting")
else:
    print(filtered_df[['Player', 'Balls faced']])
```

```
Not any players who played the match but didn't get the batting
```

In [ ]:

Question-17: Create a new column to show the percentage of total runs scored in 4s and 6s. Then print the top 5 players with maximum percentage.

In [95]:

```
import pandas as pd
df = pd.read_csv('IPL_Assignment.csv')
# Calculate runs from 4s and 6s
df['runs_from_fours'] = df['4s'] * 4
df['runs_from_sixes'] = df['6s'] * 6
df['total_fours_sixes'] = df['runs_from_fours'] + df['runs_from_sixes']
# Calculate percentage of total runs
df['percentage_fours_sixes'] = (df['total_fours_sixes'] / df['Runs']) * 100
df = df.dropna()
result_df = df.sort_values(by='percentage_fours_sixes', ascending=False)
# Convert percentage to integer to remove decimal points
result_df['percentage_fours_sixes'] = result_df['percentage_fours_sixes'].astype(int)
disp_df = result_df.head(5)
print(disp_df[['Player', 'percentage_fours_sixes']])
```

```
             Player  percentage_fours_sixes
109      Andrew Tye                     100
74     Chris Morris                      76
48    Andre Russell                      76
29    Hardik Pandya                      73
47     Sunil Narine                      72
```

In [ ]:

Question-18: Print the players with top 5 Not out percentages (Not Out percentage can be calculated as number of Not outs divided by Innings).

In [103…

```
import pandas as pd
df = pd.read_csv('IPL_Assignment.csv')
# Calculate percentage of Not_Out_percent
df['Not_Out_percent'] = (df['Not Out'] / df['Inns']) * 100
df = df.dropna()
result_df = df.sort_values(by='Not_Out_percent', ascending=False)
# Convert percentage to integer to remove decimal points
result_df['Not_Out_percent'] = result_df['Not_Out_percent'].astype(int)
disp_df = result_df.head(5)
```

```
print("Top 5 Players with Highest Not out percentages")
print("************************************************")
print(disp_df[['Player', 'Not_Out_percent']])
```

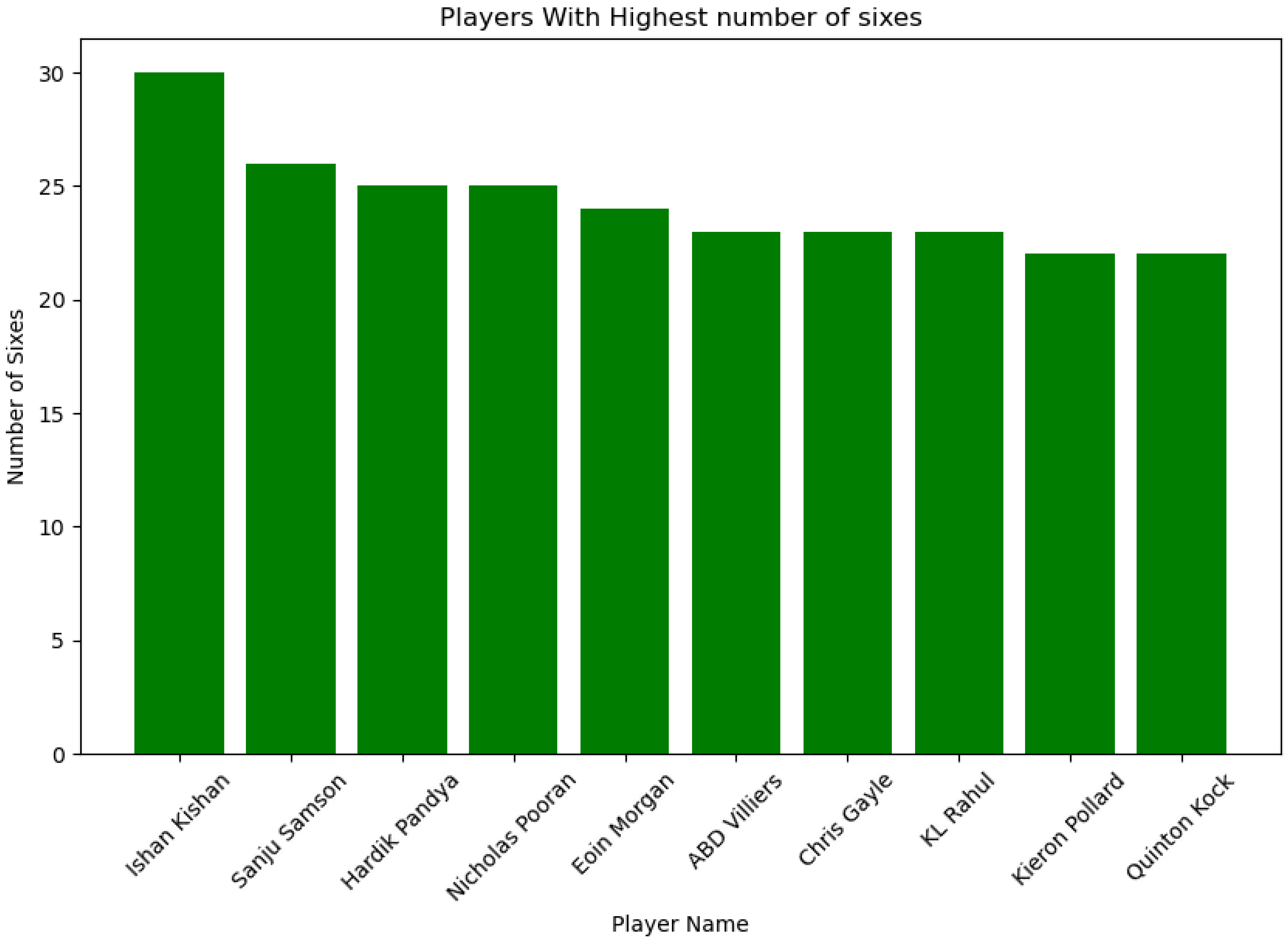
```
Top 5 Players with Highest Not out percentages
************************************************
              Player  Not_Out_percent
122    Shahbaz Ahmed              100
97     Mohammad Nabi              100
114      T Natarajan              100
116     Rahul Chahar              100
113  Dhawal Kulkarni              100
```

In [ ]:

Question-19: Create visualization of top 10 players with highest number of sixes.

In [107…

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('IPL_Assignment.csv')
# Sort by Sixes in ascending order
result_df = df.sort_values(by='6s', ascending=False)
#Pick top 10 Series
disp_df = result_df.head(10)
plt.figure(figsize=(10, 6))
plt.bar(disp_df['Player'], disp_df['6s'], color='green')
plt.xlabel('Player Name')
plt.ylabel('Number of Sixes')
plt.title('Players With Highest number of sixes')
plt.xticks(rotation=45)
plt.show()
```

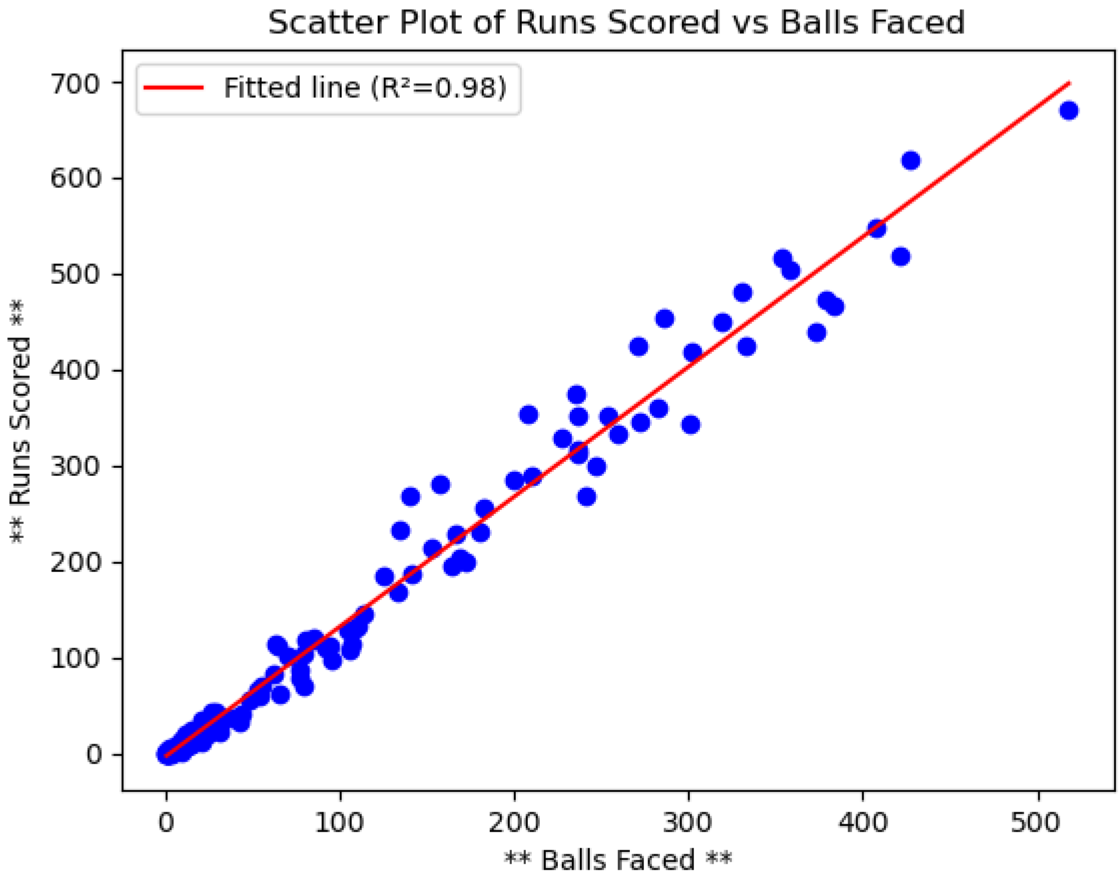

Players With Highest number of sixes
Number of Sixes
0
5
10
15
20
25
30
Ishan Kishan
Sanju Samson
Hardik Pandya
Nicholas Pooran
Eoin Morgan
ABD Villiers
Chris Gayle
KL Rahul
Kieron Pollard
Quinton Kock
Player Name

In [ ]:

Question-20: Scatter plot of runs scored by a player v/s balls faced in a season. Then find the relationship between these 2 variables.

In [111…

```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import linregress

df = pd.read_csv('IPL_Assignment.csv')

plt.scatter(df['Balls faced'], df['Runs'], color='blue')
plt.xlabel('** Balls Faced **')
plt.ylabel('** Runs Scored **')
plt.title('Scatter Plot of Runs Scored vs Balls Faced')

# Fit a trend line
slope, intercept, r_value, p_value, std_err = linregress(df['Balls faced'], df['Runs'])
plt.plot(df['Balls faced'], intercept + slope * df['Balls faced'], 'r', label=f'Fitted line (R²={r_value**2:.2f})')
# Show legend
plt.legend()
plt.show()
```



Thanks : Rajesh Bisht , rbisht.india@gmai.com

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```