HINDUSTHAN EDUCATIONAL AND CHARITABLE TRUST
HICET

IBM Career Education

PROJECT DOCUMENTS

Project Title

SPORTS ANALYTICS

Team Number & Teammates:

ABHINANTANA P G - 720722104002
FAAZIL K - 720722104011
LINGESH R - 720722104022
SHAMAN DEV M - 720722104038

Department: B.E CSE

Year : II

Description About the project

Sports analytics involves the application of statistical analysis, data mining, and predictive modeling techniques to sports-related data in order to gain insights, make informed decisions, and improve performance in various aspects of sports. Here's a breakdown of its key components:

1. Performance Analysis:

   Sports analytics is used to analyze and evaluate player and team performance. This includes assessing individual player statistics such as goals scored, assists, shooting accuracy, completion rates, defensive actions, etc., as well as team-level metrics like possession percentage, scoring efficiency, defensive solidity, etc.

2. Tactical Analysis:

   Coaches and teams use sports analytics to analyze opponent strategies, game plans, and patterns of play. By studying historical data and match footage, they can identify opponent strengths and weaknesses, tactical trends, and optimal strategies to counter opponents effectively.

3. Recruitment and Scouting:

   Sports analytics plays a crucial role in player recruitment and scouting processes. Teams utilize statistical models and data-driven methods to identify talented players, assess their potential value, and make informed decisions about player acquisitions, transfers, and draft picks.

4. Injury Prevention and Management:

   Analyzing player biometric data, injury history, workload metrics, and other relevant factors helps teams identify injury risk factors, develop injury prevention strategies, and manage player fitness and recovery effectively. This includes monitoring player workload, fatigue levels, and biomechanical data to reduce the risk of injuries and optimize performance.

5. Fan Engagement and Experience:

   Sports analytics is also used to enhance fan engagement and experience through data-driven insights, personalized content, interactive applications, and fan-focused initiatives. By analyzing fan behavior, preferences, and feedback, sports organizations can tailor their marketing efforts, ticketing strategies, and fan engagement initiatives to better connect with their audience.

6. Game Strategy and Decision Making:

   Coaches and managers use analytics to make data-driven decisions in real-time during matches. This includes analyzing in-game statistics, performance metrics, and situational data to make tactical adjustments, substitutions, and strategic decisions that maximize the team's chances of success.

7. Broadcasting and Media:

   Sports analytics is increasingly being used in broadcasting and media coverage to enhance the viewing experience, provide deeper insights into matches and player performance, and create engaging content for audiences. This includes advanced analytics-based graphics, visualizations, and commentary tools that offer viewers a more immersive and informative experience.

How Data Visualization used in Sports Analytics

Data visualization plays a pivotal role in sports analytics by providing insights, aiding decision-making processes, and enhancing understanding of complex datasets. The visualizations facilitate data-driven decision-making, performance analysis, and strategic planning in sports analytics, enabling teams and coaches to optimize player performance, devise effective game plans, and gain a competitive edge on the field.

Data visualization is used extensively in the provided code to present insights and trends from the IPL 2016 dataset. Here's how it's utilized:

1. Pie Chart:
   The top 10 run scorers are visualized using a pie chart. Each player's contribution to the total runs is represented as a slice of the pie, providing a quick comparison of their contributions.

2. Bubble Chart:
   Two bubble charts are created to visualize the relationship between runs scored and strike rate. One chart displays the performance of the top 15 players, while the other shows the performance of all players. Bubble size represents the number of matches played, allowing for a comparison of performance across different metrics.

3. Heatmap:
   A heatmap is used to visualize the shot distribution of a specific player on a cricket field. This provides insights into where the player tends to hit the ball more frequently.

4. Radar Chart:
   Radar charts are employed to display various performance metrics (such as runs, average, strike rate, centuries, half-centuries, fours, and sixes) of a selected player in comparison to others. This allows for a multivariate comparison of players' performance.

5. Bar Chart:
   A bar chart is used to visualize the number of half-centuries and centuries scored by the top 10 run scorers. This provides a comparison of milestone achievements among players.

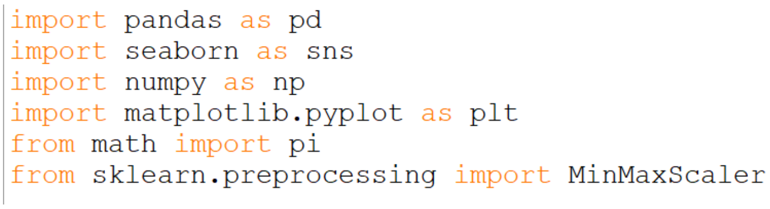
6. Line Plot:

Finally, a line plot is created to compare the number of fours and sixes hit by the top 25 players in the tournament. This allows for tracking the frequency of boundary hits by different players.

By using various types of visualizations, the code effectively communicates different aspects of player performance and statistical insights from the IPL 2016 dataset.

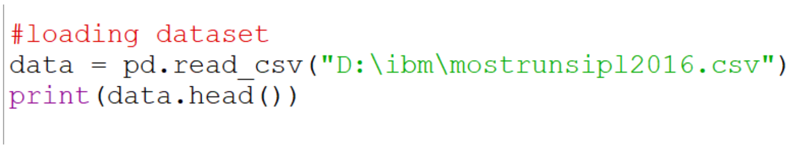
Code and Output

Importing packages :

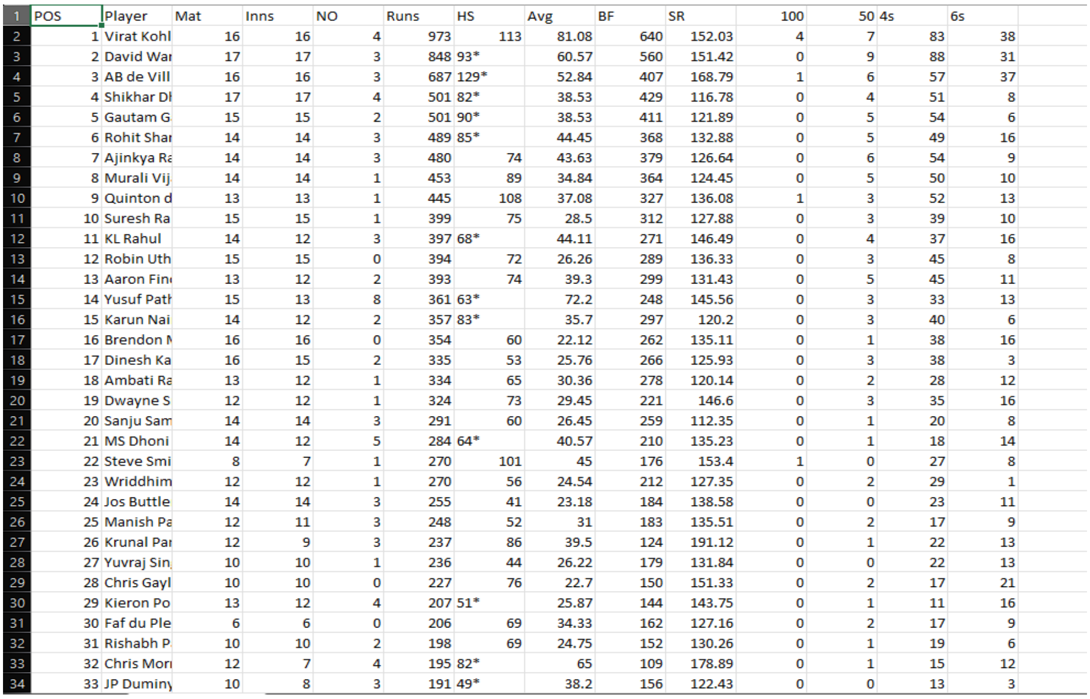


```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from math import pi
from sklearn.preprocessing import MinMaxScaler
```

Loading Dataset :

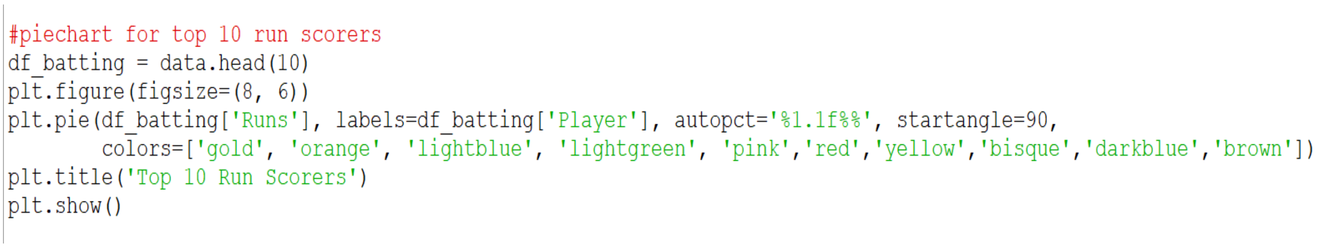


```
#loading dataset
data = pd.read_csv("D:\ibm\mostrunsipl2016.csv")
print(data.head())
```



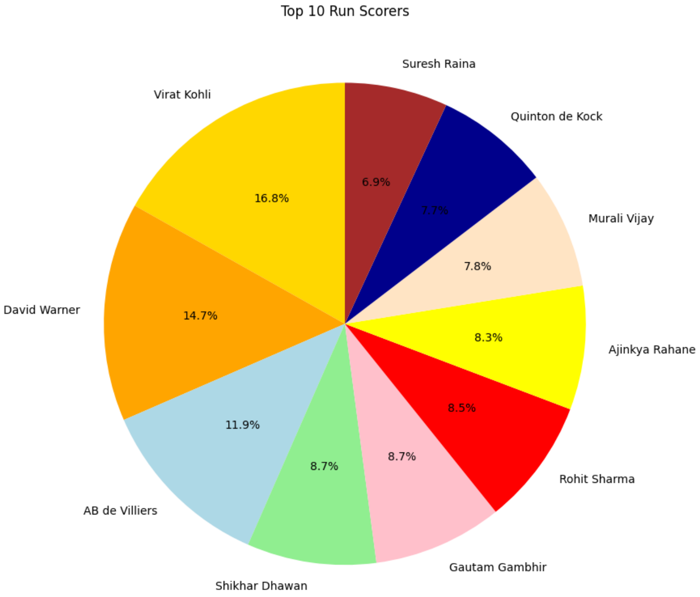
1	POS	Player	Mat	Inns	NO	Runs	HS	Avg	BF	SR	100	50	4s	6s
2	1	Virat Kohl	16	16	4	973	113	81.08	640	152.03	4	7	83	38
3	2	David War	17	17	3	848	93*	60.57	560	151.42	0	9	88	31
4	3	AB de Vill	16	16	3	687	129*	52.84	407	168.79	1	6	57	37
5	4	Shikhar Dl	17	17	4	501	82*	38.53	429	116.78	0	4	51	8
6	5	Gautam G	15	15	2	501	90*	38.53	411	121.89	0	5	54	6
7	6	Rohit Shar	14	14	3	489	85*	44.45	368	132.88	0	5	49	16
8	7	Ajinkya Ra	14	14	3	480	74	43.63	379	126.64	0	6	54	9
9	8	Murali Vij	14	14	1	453	89	34.84	364	124.45	0	5	50	10
10	9	Quinton d	13	13	1	445	108	37.08	327	136.08	1	3	52	13
11	10	Suresh Ra	15	15	1	399	75	28.5	312	127.88	0	3	39	10
12	11	KL Rahul	14	12	3	397	68*	44.11	271	146.49	0	4	37	16
13	12	Robin Uth	15	15	0	394	72	26.26	289	136.33	0	3	45	8
14	13	Aaron Fin	13	12	2	393	74	39.3	299	131.43	0	5	45	11
15	14	Yusuf Path	15	13	8	361	63*	72.2	248	145.56	0	3	33	13
16	15	Karun Nai	14	12	2	357	83*	35.7	297	120.2	0	3	40	6
17	16	Brendon M	16	16	0	354	60	22.12	262	135.11	0	1	38	16
18	17	Dinesh Ka	16	15	2	335	53	25.76	266	125.93	0	3	38	3
19	18	Ambati Ra	13	12	1	334	65	30.36	278	120.14	0	2	28	12
20	19	Dwayne S	12	12	1	324	73	29.45	221	146.6	0	3	35	16
21	20	Sanju Sam	14	14	3	291	60	26.45	259	112.35	0	1	20	8
22	21	MS Dhoni	14	12	5	284	64*	40.57	210	135.23	0	1	18	14
23	22	Steve Smi	8	7	1	270	101	45	176	153.4	1	0	27	8
24	23	Wriddhim	12	12	1	270	56	24.54	212	127.35	0	2	29	1
25	24	Jos Buttle	14	14	3	255	41	23.18	184	138.58	0	0	23	11
26	25	Manish Pa	12	11	3	248	52	31	183	135.51	0	2	17	9
27	26	Krunal Pa	12	9	3	237	86	39.5	124	191.12	0	1	22	13
28	27	Yuvraj Sin	10	10	1	236	44	26.22	179	131.84	0	0	22	13
29	28	Chris Gayl	10	10	0	227	76	22.7	150	151.33	0	2	17	21
30	29	Kieron Po	13	12	4	207	51*	25.87	144	143.75	0	1	11	16
31	30	Faf du Ple	6	6	0	206	69	34.33	162	127.16	0	2	17	9
32	31	Rishabh P	10	10	2	198	69	24.75	152	130.26	0	1	19	6
33	32	Chris Mor	12	7	4	195	82*	65	109	178.89	0	1	15	12
34	33	JP Duminy	10	8	3	191	49*	38.2	156	122.43	0	0	13	3

Pie Chart :

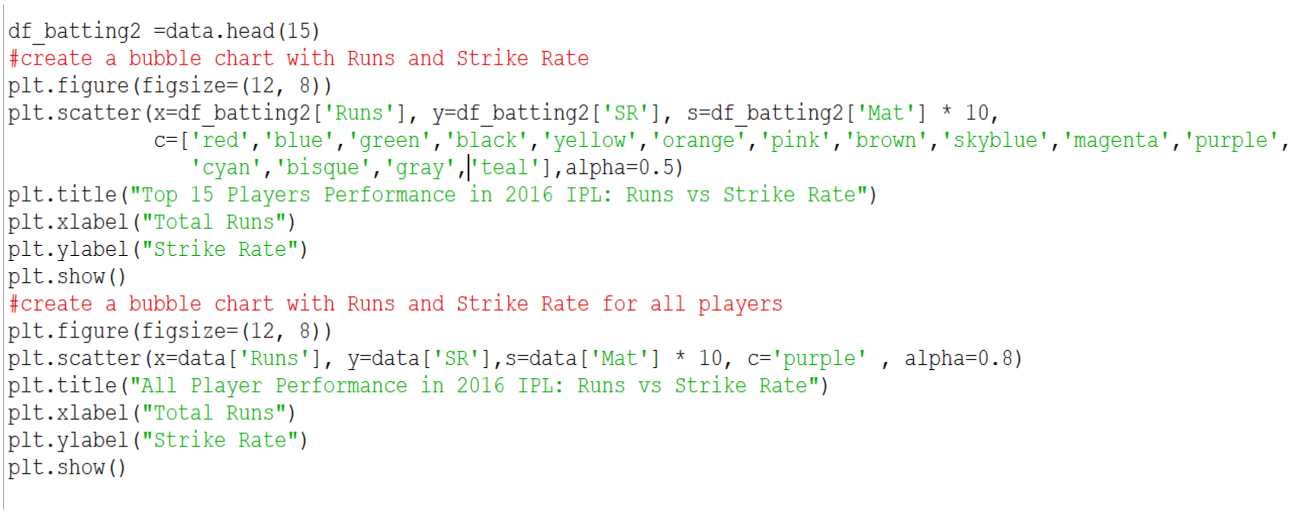


```
#piechart for top 10 run scorers
df_batting = data.head(10)
plt.figure(figsize=(8, 6))
plt.pie(df_batting['Runs'], labels=df_batting['Player'], autopct='%1.1f%%', startangle=90,
        colors=['gold', 'orange', 'lightblue', 'lightgreen', 'pink','red','yellow','bisque','darkblue','brown'])
plt.title('Top 10 Run Scorers')
plt.show()
```

Output :

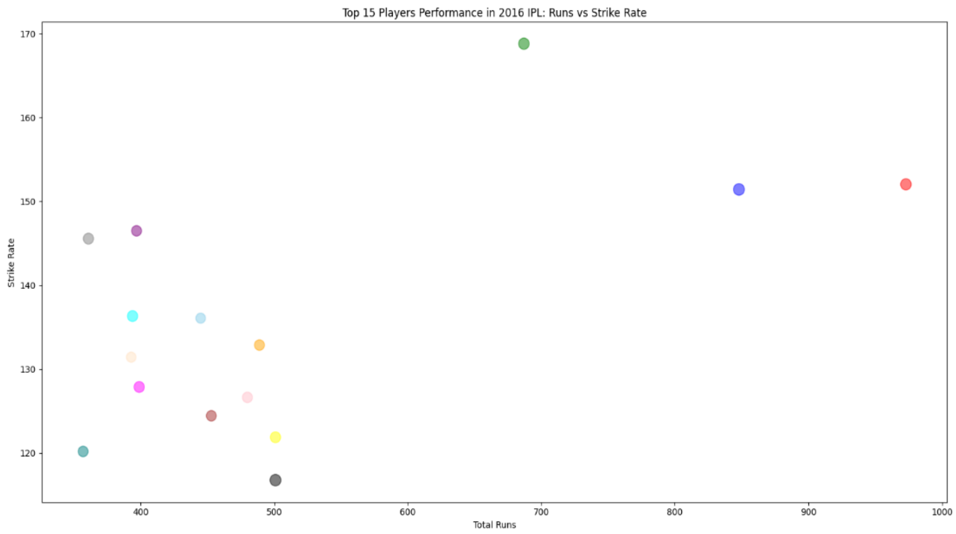


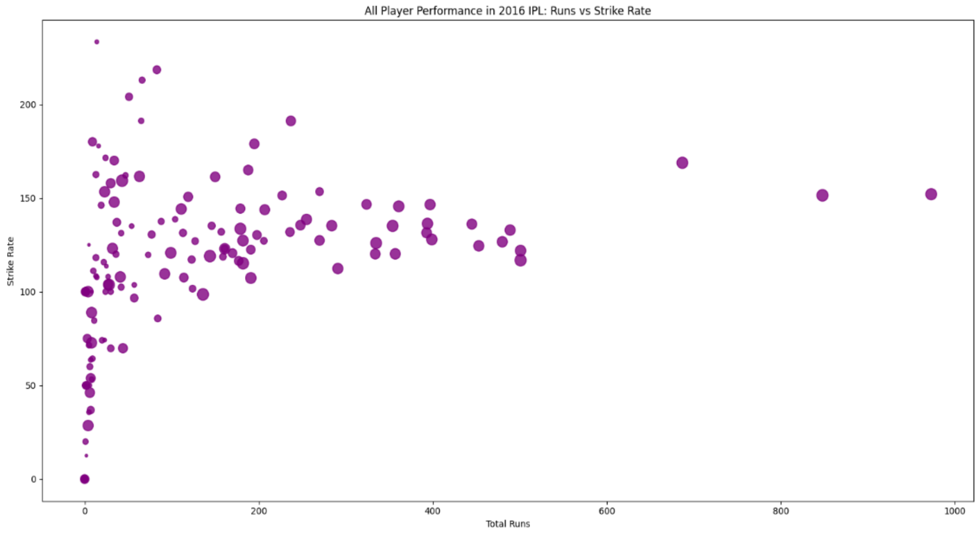
Bubble Chart :



```
df_batting2 =data.head(15)
#create a bubble chart with Runs and Strike Rate
plt.figure(figsize=(12, 8))
plt.scatter(x=df_batting2['Runs'], y=df_batting2['SR'], s=df_batting2['Mat'] * 10,
            c=['red','blue','green','black','yellow','orange','pink','brown','skyblue','magenta','purple',
               'cyan','bisque','gray','teal'],alpha=0.5)
plt.title("Top 15 Players Performance in 2016 IPL: Runs vs Strike Rate")
plt.xlabel("Total Runs")
plt.ylabel("Strike Rate")
plt.show()
#create a bubble chart with Runs and Strike Rate for all players
plt.figure(figsize=(12, 8))
plt.scatter(x=data['Runs'], y=data['SR'],s=data['Mat'] * 10, c='purple' , alpha=0.8)
plt.title("All Player Performance in 2016 IPL: Runs vs Strike Rate")
plt.xlabel("Total Runs")
plt.ylabel("Strike Rate")
plt.show()
```

Output :

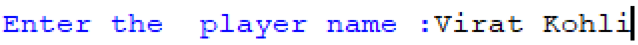




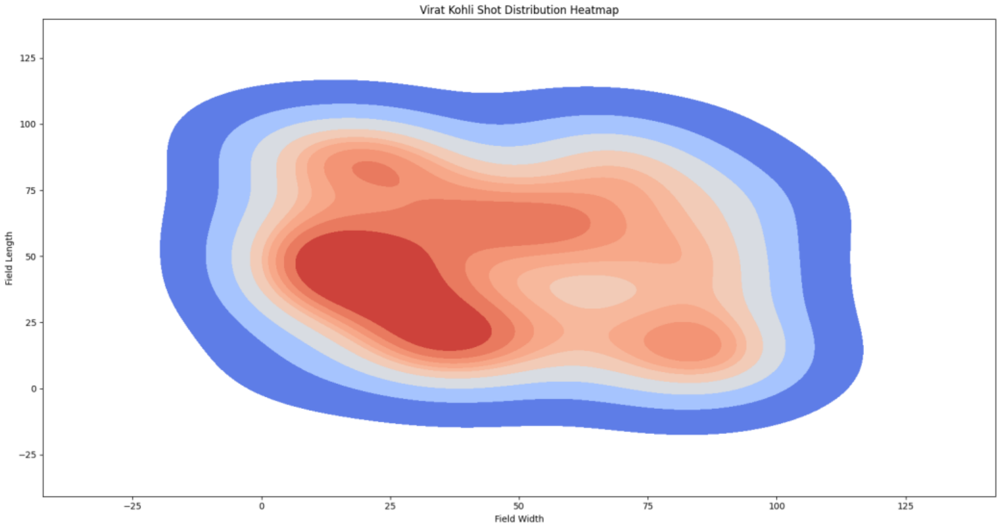
Heatmap :

```
Player_name = input('Enter the  player name :')
shot_data = {
    'Player': [Player_name] * 100,
    'x': np.random.randint(0, 101, 100),  # x-coordinates
    'y': np.random.randint(0, 101, 100)   # y-coordinates
}
df_shots = pd.DataFrame(shot_data)
# Create a heatmap for shot distribution on a cricket field
plt.figure(figsize=(10, 6))
sns.kdeplot(data=df_shots, x='x', y='y', cmap='coolwarm', fill=True, thresh=0.05)
plt.title(Player_name+' Shot Distribution Heatmap')
plt.xlabel('Field Width')
plt.ylabel('Field Length')
plt.show()
```

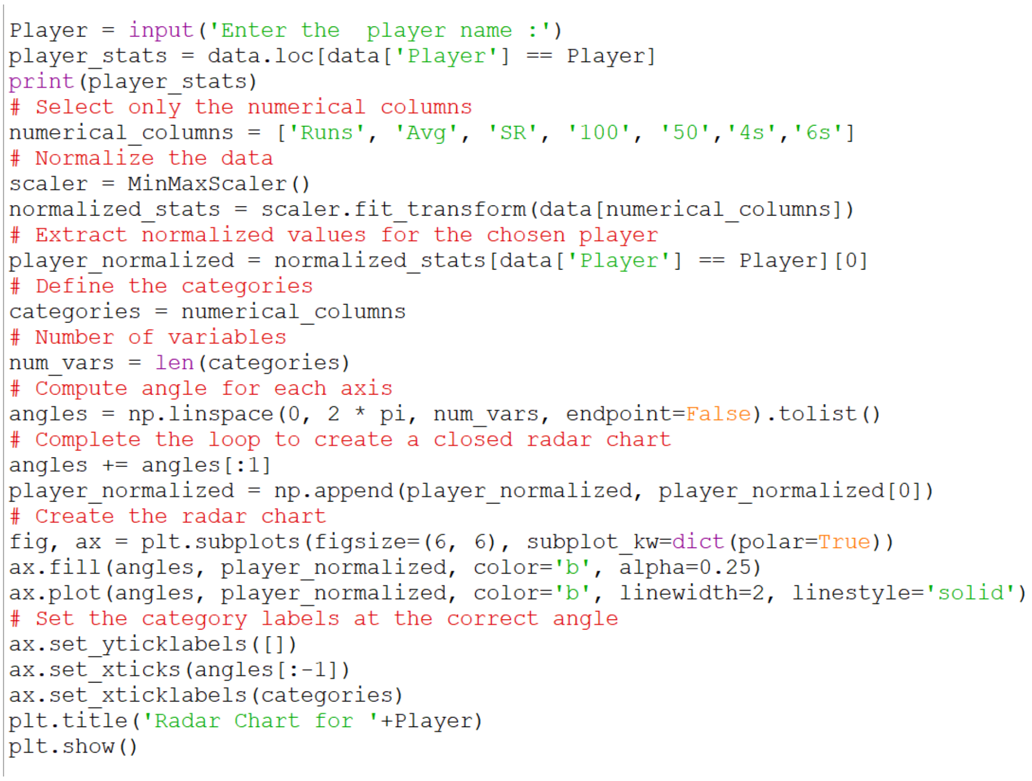
Output :



```
Enter the  player name :Virat Kohli
```

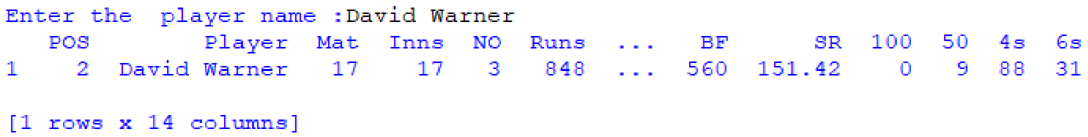


Radar Chart :



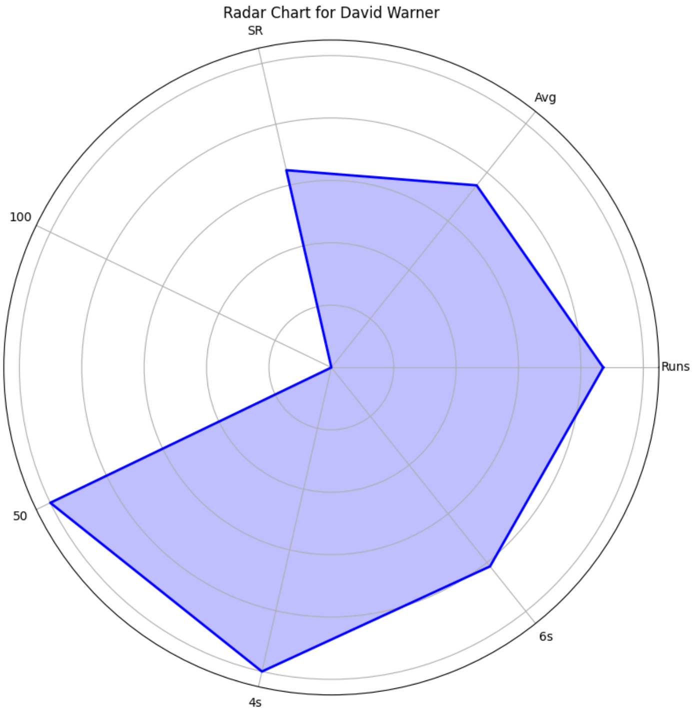
```
Player = input('Enter the  player name :')
player_stats = data.loc[data['Player'] == Player]
print(player_stats)
# Select only the numerical columns
numerical_columns = ['Runs', 'Avg', 'SR', '100', '50','4s','6s']
# Normalize the data
scaler = MinMaxScaler()
normalized_stats = scaler.fit_transform(data[numerical_columns])
# Extract normalized values for the chosen player
player_normalized = normalized_stats[data['Player'] == Player][0]
# Define the categories
categories = numerical_columns
# Number of variables
num_vars = len(categories)
# Compute angle for each axis
angles = np.linspace(0, 2 * pi, num_vars, endpoint=False).tolist()
# Complete the loop to create a closed radar chart
angles += angles[:1]
player_normalized = np.append(player_normalized, player_normalized[0])
# Create the radar chart
fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(polar=True))
ax.fill(angles, player_normalized, color='b', alpha=0.25)
ax.plot(angles, player_normalized, color='b', linewidth=2, linestyle='solid')
# Set the category labels at the correct angle
ax.set_yticklabels([])
ax.set_xticks(angles[:-1])
ax.set_xticklabels(categories)
plt.title('Radar Chart for '+Player)
plt.show()
```

Output :

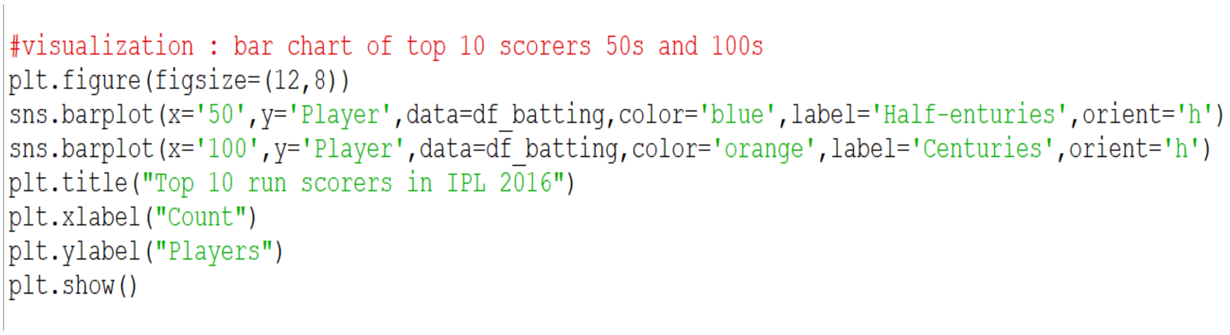


```
Enter the  player name :David Warner
   POS          Player  Mat  Inns  NO  Runs  ...   BF      SR  100  50  4s  6s
1    2  David Warner   17    17   3   848  ...  560  151.42    0   9  88  31

[1 rows x 14 columns]
```

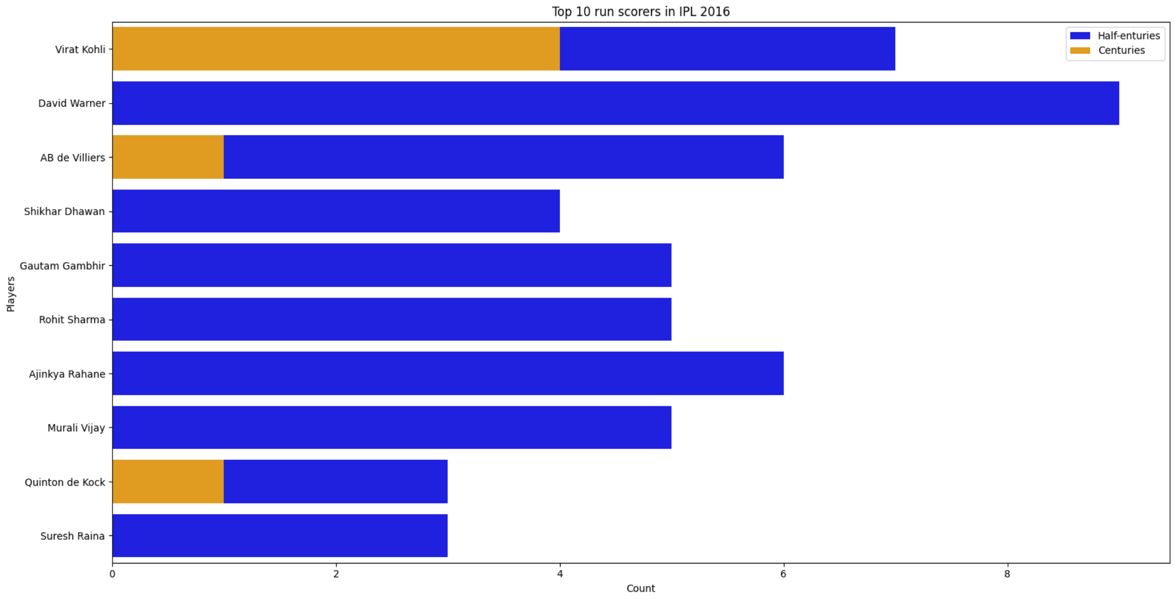


Bar Chart :



```
#visualization : bar chart of top 10 scorers 50s and 100s
plt.figure(figsize=(12,8))
sns.barplot(x='50',y='Player',data=df_batting,color='blue',label='Half-enturies',orient='h')
sns.barplot(x='100',y='Player',data=df_batting,color='orange',label='Centuries',orient='h')
plt.title("Top 10 run scorers in IPL 2016")
plt.xlabel("Count")
plt.ylabel("Players")
plt.show()
```

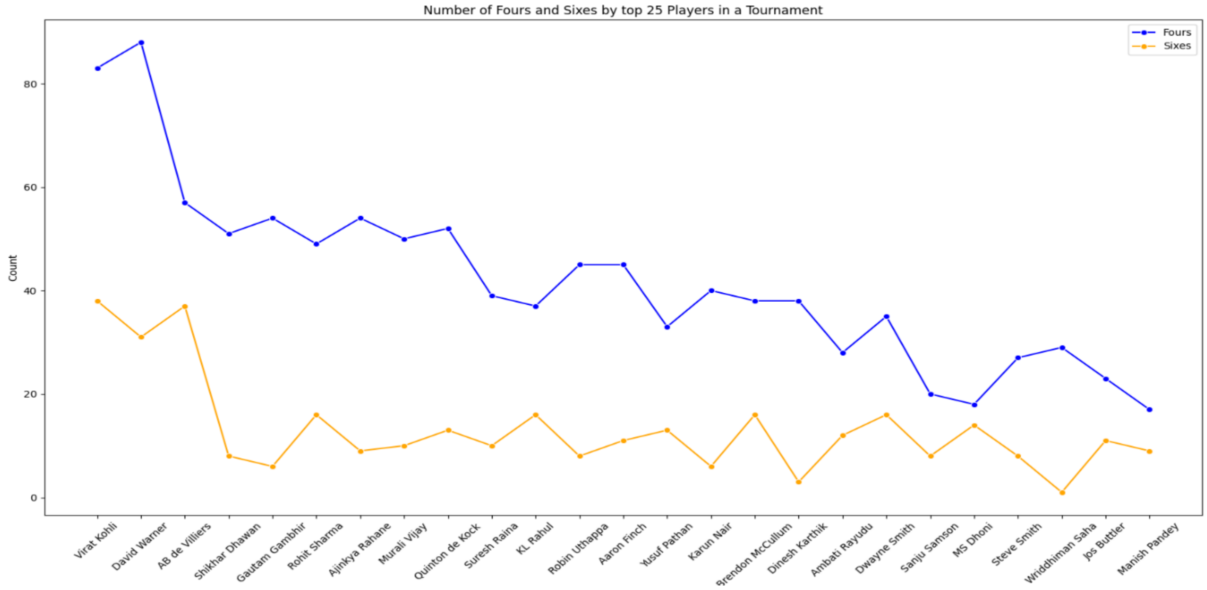
Output :



Line Chart :

```
df = data.head(25)
# Create a line plot for the 4s and 6s
plt.figure(figsize=(10, 6))
sns.lineplot(x='Player', y='4s', data=df, label='Fours', marker='o', linestyle='-', color='blue')
sns.lineplot(x='Player', y='6s', data=df, label='Sixes', marker='o', linestyle='-', color='orange')
plt.title('Number of Fours and Sixes by top 25 Players in a Tournament')
plt.xlabel('Player')
plt.ylabel('Count')
plt.legend()
# Rotate x-axis labels for better visibility
plt.xticks(rotation=45)
plt.show()
```

Output :



Github Link :

https://github.com/lingesh1174/IBM.git