DATA PRE-PROCESSING & VISUALIZATION ASSESMENT

Introduction

Data pre-processing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms.

Why is Data pre-processing important?

Pre-processing of data is mainly to check the data quality. The quality can be checked by checking Accuracy, Completeness, Consistency, Timeliness, Believability, Interpretability.

Data Pre-processing (Dataset Link: Click Here)

1. **Data Cleaning**
   a. Fill/ Remove Missing Data based on effects on Visualization Result.
   b. Fix Noisy Data (Make Data of equal sizes and columns must have same data types)
2. **Data integration**
3. **Data reduction**
4. **Data transformation**
5. **Data Transformation**
   a. Normalization of data – means simplifying the dataset like selecting the attributes required for analysis. so, that information can be fetched easily.
   b. Hierarchy Generation, Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

About Data

Data has been scraped and transformed into following files. The data provided in match level summary as well as ball-by-ball details format for all matches from 2008 till the ongoing 2022 season.

I am going to use these datasets mentioned below:-

- all_season_summary.csv - Summary of all matches across all seasons
- all_season_details.csv - Ball-by-ball details of all matches across all seasons
- all_season_batting_card.csv - Batting performance of players, all matches across all seasons
- all_season_bowling_card.csv - Bowling performance of players, all matches across all seasons
- points_table.csv - Overall points table of teams across seasons

DATA VISUALIZATION

Data visualization in python is perhaps one of the most utilized features for data science with python in today's day and age. The libraries in python come with lots of different features that enable users to make highly customized, elegant, and interactive plots.
In this article, we will cover the usage of Matplotlib, Seaborn as well as an introduction to other alternative packages that can be used in python visualization.
Within Matplotlib and Seaborn, we will be covering a few of the most commonly used plots in the data science world for easy visualization.

Requirements

1. VS CODE/ PyCharm/ Data Spell/ Jupyter or any other Python Supported IDE.
2. Python Version 3+ Installed (Install Python Packages - NUMPY, PANDAS, MATPLOTLIB)
3. LINUX/ WINDOWS/ MAC OS SYSTEM
4. Github Knowledge

List Of Visualizations:

1. Total Matches Played In IPL From 2008-2021
2. Season-Wise No Of Matches Played 2008-21
3. Total No Of Matches Played In Particular Stadiums 2008-21
4. Top 10 Man Of The Match Winners Of All Time IPL
5. No Of Total Matches Won By Particular IPL TEAMS 2008-21
6. Total IPL Runs Scored By Individual Players 2008-21
7. No Of Fours Hitted By Individual Players 2008-21
8. No Of Sixes Hitted By Individual Players 2008-21
9. All Time Maximum Run Scored By Orange Cap Winners In A Season from 2008-21
10. All Time Maximum Wickets Taken By Purple Cap Winners In A Season from 2008-21
11. All Time Wickets Taken By Individual Players
12. All Time Maiden Overs By Individual Players
13. All Time No Balls By Individual Players

IMPLEMENTATION

Import Packages

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
```

Loading/ Importing Dataset

```
batdataset = pd.read_csv('datasets/all_season_batting_card.csv')
balldataset = pd.read_csv('datasets/all_season_bowling_card.csv')
all_matches = pd.read_csv('datasets/all_season_summary.csv')
```

View Batting Dataset (First 10)

```
print(batdataset.head(10))
Selecting Required Columns To Create New Cleaned Battiing Dataset
null1 = batdataset.isna().sum()
print(null1)
```

 # -- Deleting Unwanted Columns From Batting Dataset --

```
del batdataset['commentary']
del batdataset['runningOver']
```

 # Fetching and Dropping NULL Valued Rows if any From Batting Dataset

```
null1 = batdataset.isna().sum()
print(null1)
cleaned_bat = batdataset.dropna()
print(cleaned_bat.head())
cleaned_bat.to_csv('cleaned_bat.csv')
```

 # Verifying Null Values In Cleaned Data

```
null2 = cleaned_bat.isna().sum()
print(null2)
```

View Balling Dataset (First 10)

```
print(balldataset.head(10))
Selecting Required Columns To Create New Cleaned Balling Dataset
# Null Values Detection in balling dataset
null2 = balldataset.isna().sum()
print(null2)
```

#Deleting Unwanted Columns From Balling Dataset

```
del balldataset['href']
```

 # Fetching and Dropping NULL Valued Rows if any From Balling Dataset

```
null1 = balldataset.isna().sum()
print(null1)
cleaned_ball = balldataset.dropna()
print(cleaned_ball.head())
cleaned_ball.to_csv('cleaned_ball.csv')
```

 # Verifying Null Values In Cleaned Balling Data

```
null2 = cleaned_ball.isna().sum()
print(null2)
```

 # Setting Colors

```
cmix = ["#00FFFF", "#F0FFFF", "#89CFF0", "#0000FF", "#7393B3", "#088F8F", "#0096FF",
"#5F9EA0","#0047AB", "#6495ED",      "#00FFFF",      "#00008B",      "#6F8FAF",      "#1434A4",
          "#7DF9FF",      "#6082B6",      "#00A36C", "#3F00FF", "#5D3FD3",      "#ADD8E6",      "#191970",
          "#000080",      "#1F51FF",      "#A7C7E7",      "#CCCCFF",      "#B6D0E2",      "#96DED1",
          "#4169E1",      "#0F52BA",      "#9FE2BF",      "#87CEEB",      "#4682B4",      "#008080",
"#40E0D0", "#0437F2", "#40B5AD",      "#0818A8"]

c10 = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2', '#00F3D0', '#bcbd22',
'#17becf']
c5 = ['#1f77b4', '#ff7f0e', '#00F3D0', '#bcbd22', '#17becf']
```

#Visualizations

1. Total Matches Played In IPL From 2008-2021

```
str1 = "Total Matches Played From IPL 2008 to 2021 are: "
str2 = all_matches.shape[0]
print(str1 + str(str2))
```

Output:

```
Total Matches Played From IPL 2008 to 2021 are: 884
```

Conclusion:

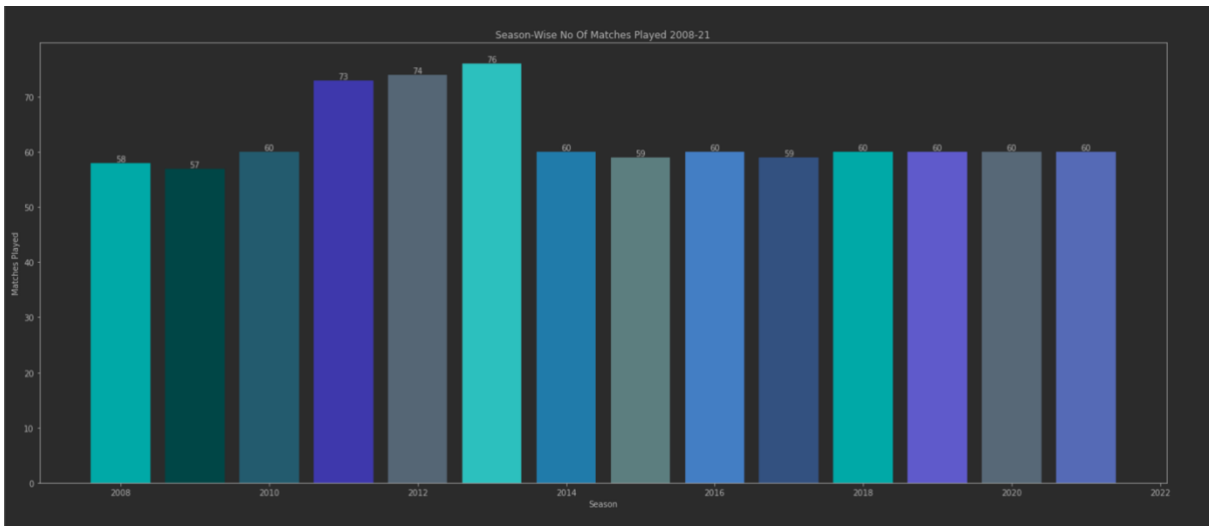
Total Matches Played From IPL 2008 to 2021 are: **884 Matches**

2. Season-Wise No Of Matches Played 2008-21

```
# Visualizing Season Wise Matches Played
data = cleaned_bat.groupby(['match_id',
'season']).count().index.droplevel(level=0).value_counts().sort_index()
x = data
y = data.index
# Figure Size
```

```
fig, ax = plt.subplots(figsize=(25, 10))
plt.bar(y, x, color=cmix, width=0.8)
plt.title("Season-Wise No Of Matches Played 2008-21")
plt.xlabel('Season')
plt.ylabel('Matches Played')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

*SEASON - TOTAL MATCHES

2008 - 58 | 2009 - 57 | 2010 - 60 | 2011 - 73 | 2012 - 74

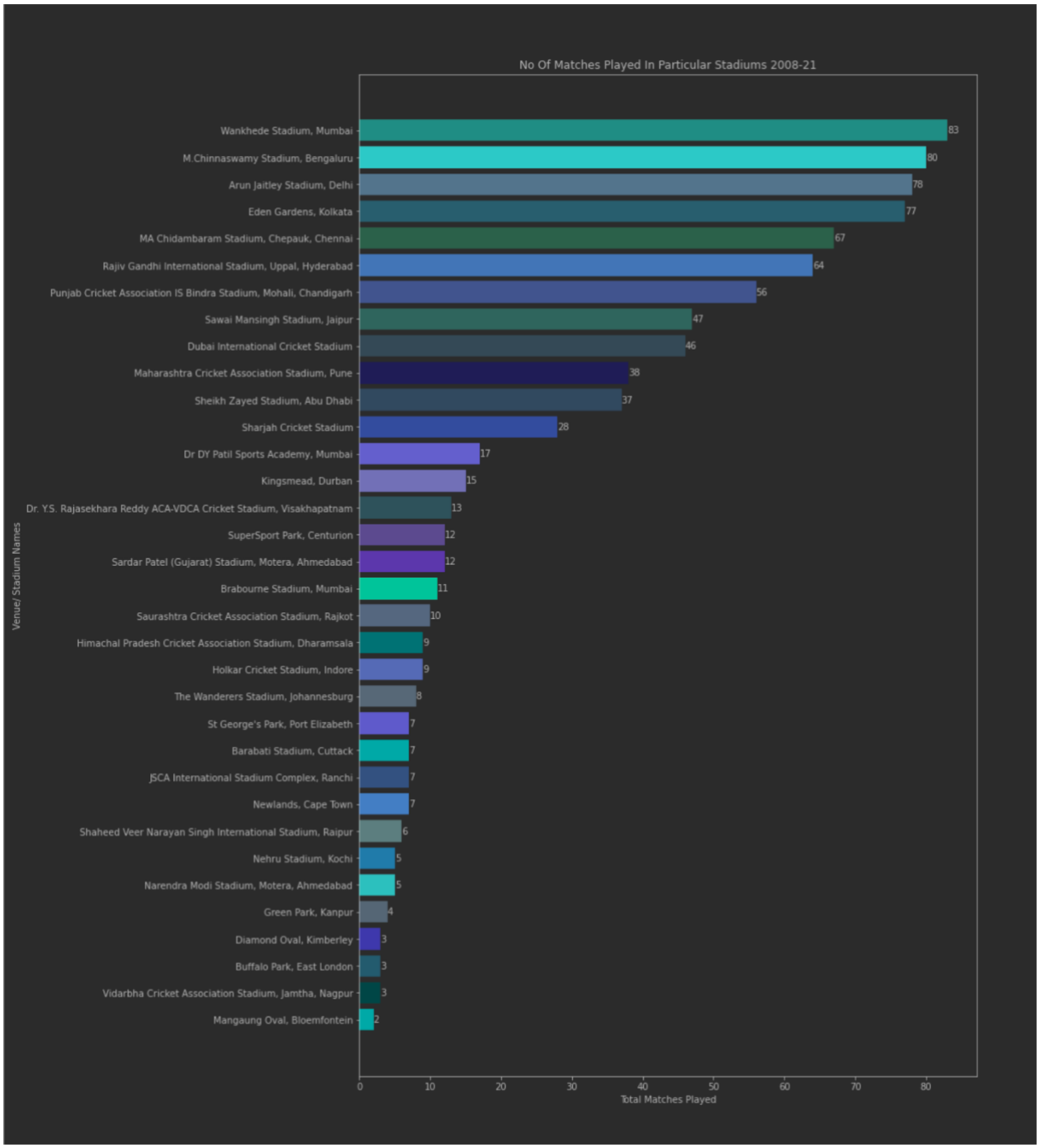
2013 - 76 | 2014 - 60 | 2015 - 59 | 2016 - 60 | 2017 - 59

2018 - 60 | 2019 - 60 | 2020 - 60 | 2021 – 60

3. Total No Of Matches Played In Particular Stadiums 2008-21

```
data = cleaned_bat.groupby(['match_id',
'venue']).count().index.droplevel(level=0).value_counts().sort_values()
x = data
y = data.index
# Figure Size
fig, ax = plt.subplots(figsize=(12, 20))
plt.barh(y, x, color=cmix)
plt.title("No Of Matches Played In Particular Stadiums 2008-21")
plt.ylabel('Venue/ Stadium Names')
plt.xlabel('Total Matches Played')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



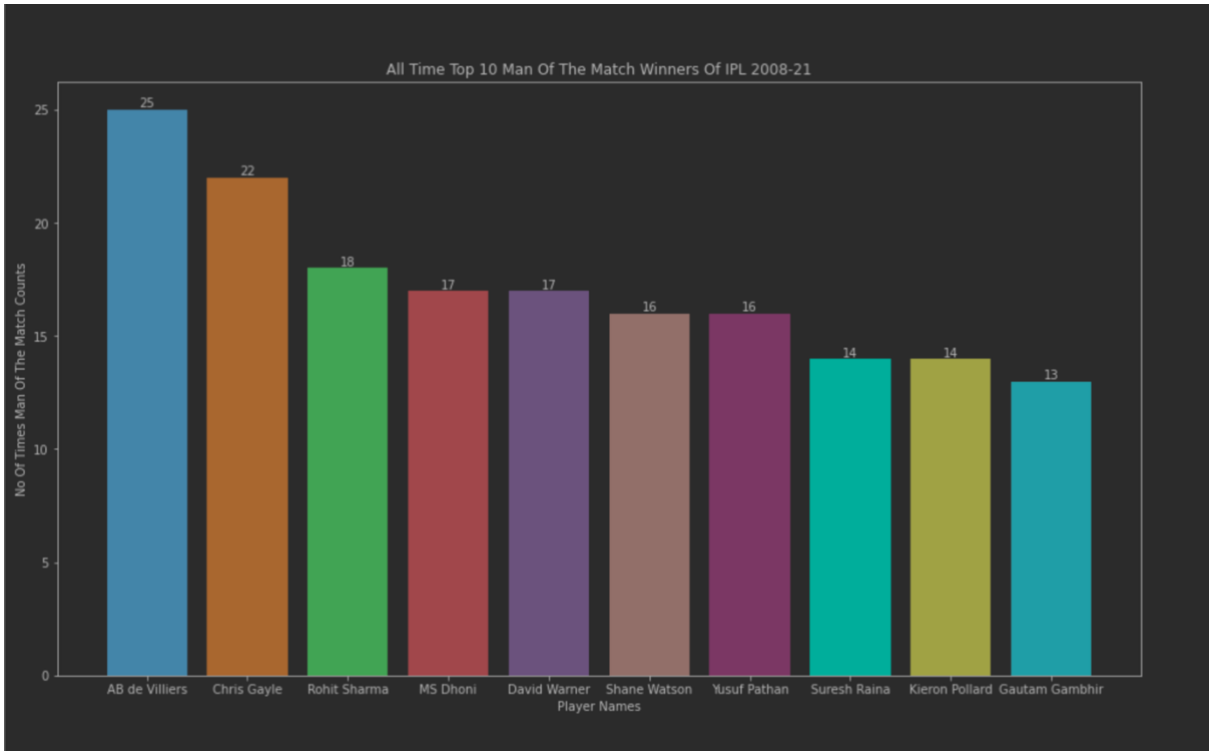
Conclusion:

- **Wankhede Stadium, Mumbai** has hosted most of the IPL Matches from 2008-21 i.e. **83 Matches.**
- M. Chinnaswamy Stadium, Bengaluru has hosted 80 IPL Matches from 2008-21.
- Arun Jaitley Stadium, Delhi has hosted 78 IPL Matches from 2008-21.

4. Top 10 Man Of The Match Winners Of All Time IPL

```
# Top 10 Man Of The Match Winners Of All Time IPL
t5motm = all_matches['pom'].value_counts()[:10].sort_values(ascending=False)
x = t5motm
y = t5motm.index
# Figure Size
fig, ax = plt.subplots(figsize=(16, 9))
plt.bar(y, x, color=c10)
plt.title("All Time Top 10 Man Of The Match Winners Of IPL 2008-21")
plt.ylabel('No Of Times Man Of The Match Counts')
plt.xlabel('Player Names')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

Top 10 Man Of The Match Winners with No Of Times They Got Awarded.

- **25 Times – AB de Villiers**
- 22 Times – Chris Gayle
- 18 Times – Rohit Sharma
- 17 Times – MS Dhoni, David Warner
- 16 Times – Shane Watson, Yusuf Pathan
- 14 Times – Suresh Raina, Kieron Pollard
- 13 Times – Gautam Gambhir

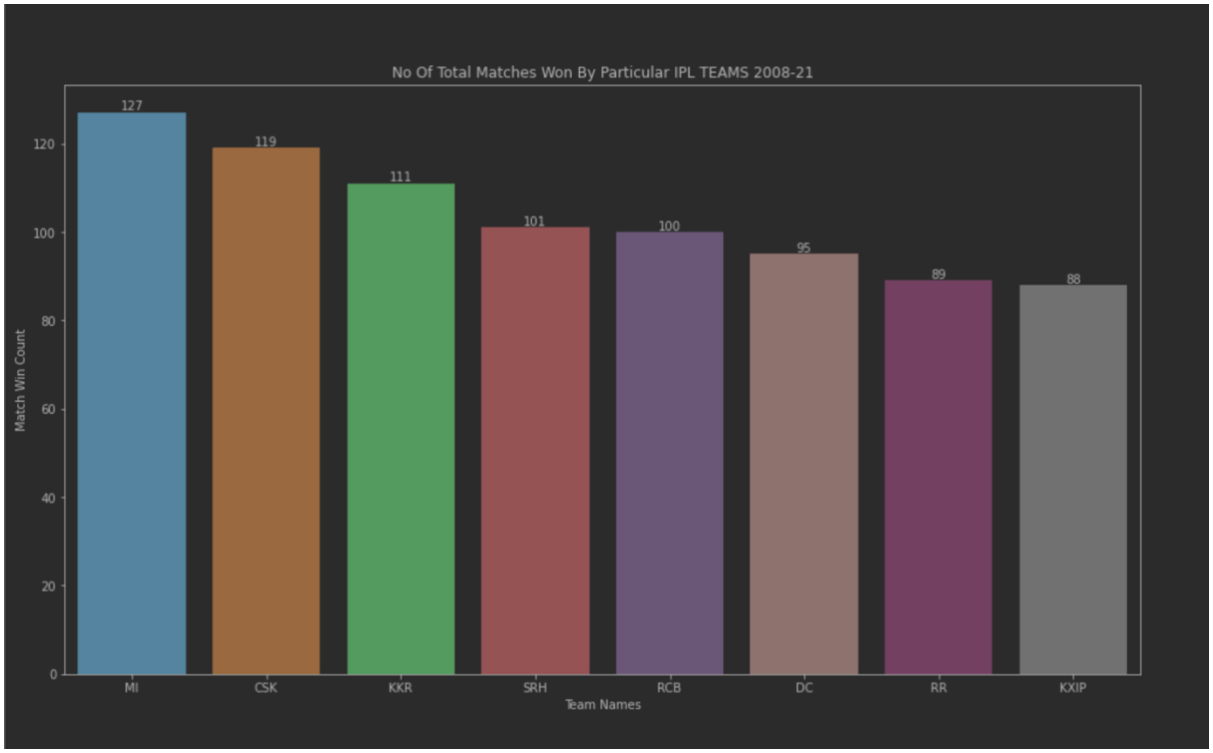
5. No Of Total Matches Won By Particular IPL TEAMS 2008-21

```
all_matches['Year'] = all_matches['season']
all_matches['winner'].value_counts()[:5].sort_values()

plt.figure(figsize=(16, 9))
ax = sns.countplot(x='winner', data=all_matches, order=all_matches['winner'].value_counts()[:8].index)

plt.title("No Of Total Matches Won By Particular IPL TEAMS 2008-21")
plt.ylabel('Match Win Count')
plt.xlabel('Team Names')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

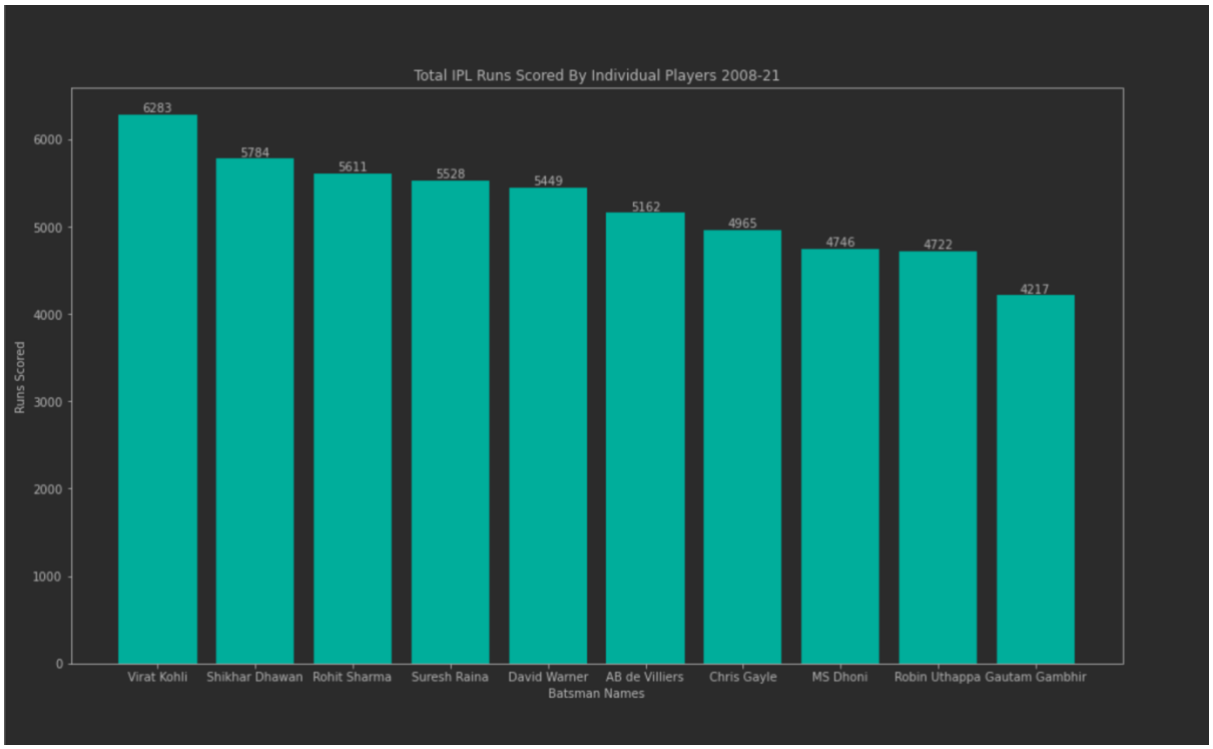
Total No Of Matches Won By Particular IPL Teams 2008-21(Top to Bottom) List:

- **127 - MI**
- 119 - CSK
- 111 - KKR
- 101 - SRH
- 100 - RCB
- 95 - DC
- 89 - RR
- 88 - KXIP

6. Total IPL Runs Scored By Individual Players 2008-21

```
fig, ax = plt.subplots(figsize=(16, 9))
data = cleaned_bat.groupby(['fullName'])['runs'].sum().sort_values(ascending=False)[:10]
x = data
y = data.index
plt.title("Total IPL Runs Scored By Individual Players 2008-21")
plt.bar(y, x, color='#00F3D0')
plt.xlabel('Batsman Names')
plt.ylabel('Runs Scored')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

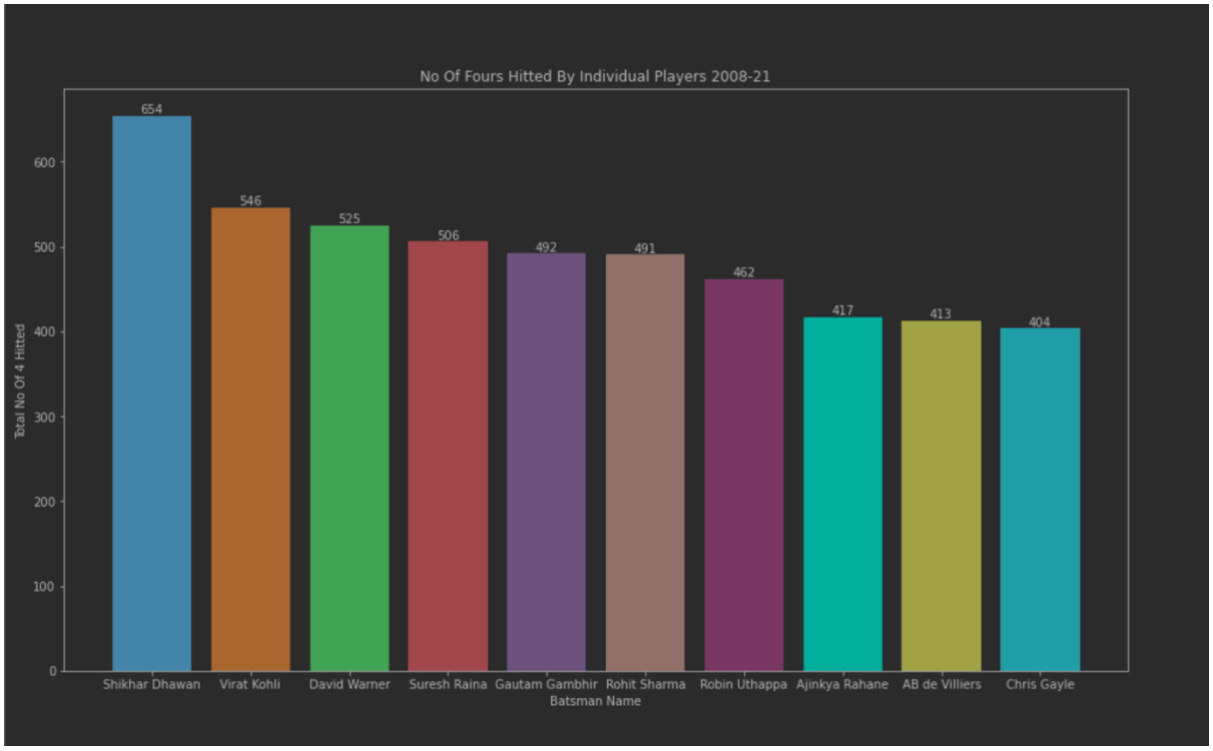
Total IPL Runs Scored By Individual Players 2008-21

- **Virat Kohli – 6283**
- Shikhar Dhawan – 5784
- Rohit Sharma – 5611
- Suresh Raina – 5528
- David Warner – 5449
- AB de Villiers – 5162
- Chris Gayle – 4965
- MS Dhoni – 4746
- Robin Uthappa – 4722
- Gautam Gambhir – 4217

7. No Of Fours Hitted By Individual Players 2008-21

```
fig, ax = plt.subplots(figsize=(16, 9))
data = cleaned_bat.groupby(['fullName'])['fours'].sum().sort_values(ascending=False)[:10]
x = data
y = data.index
plt.title("No Of Fours Hitted By Individual Players 2008-21")
plt.bar(y, x, color=c10)
plt.xlabel('Batsman Name')
plt.ylabel('Total No Of 4 Hitted')
for bars in ax.containers:
ax.bar_label(bars)
```

Output:



Conclusion:

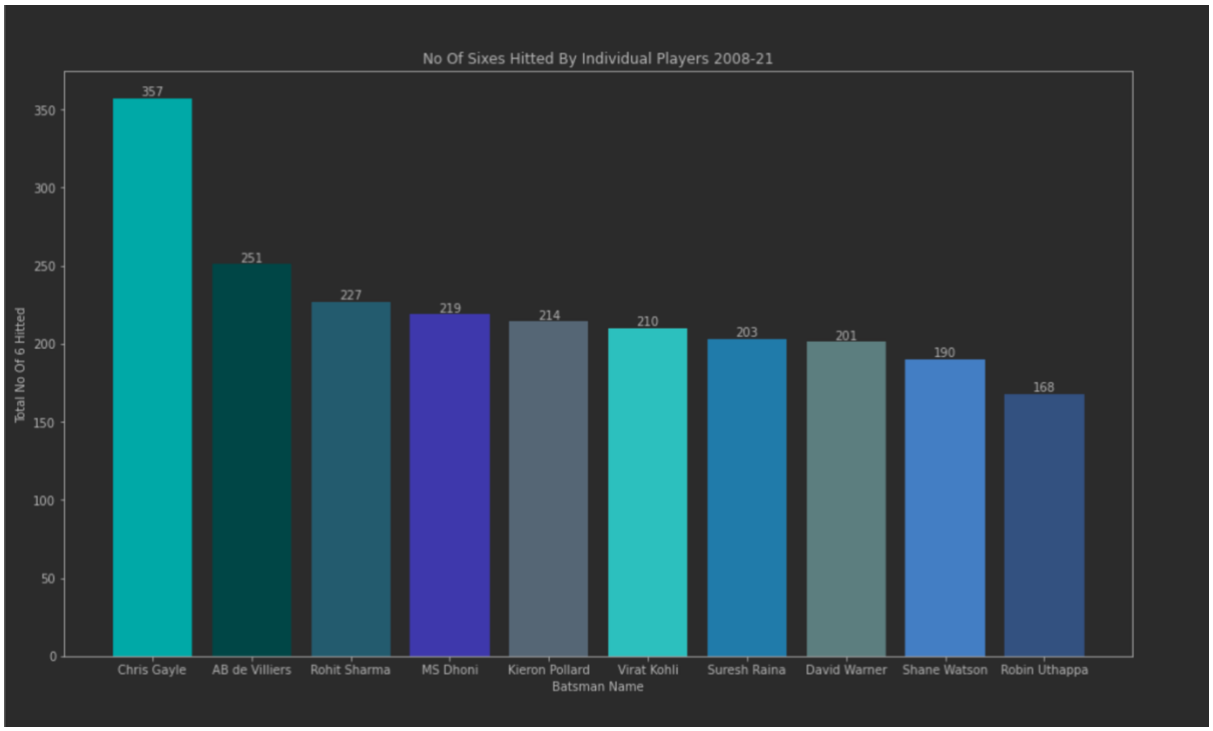
Most No Fours Hitted By Individual Players in IPL 2008-21.

- **Shikhar Dhawan – 654**
- Virat Kohli – 546
- David Warner – 525
- Suresh Raina – 506
- Gautam Gambhir – 492
- Rohit Sharma – 491
- Robin Uthappa – 462
- Ajinkya Rahane – 417
- AB de Villiers – 413
- Chris Gayle – 404

8. No Of Sixes Hitted By Individual Players 2008-21

```
fig, ax = plt.subplots(figsize=(16, 9))
data1 = cleaned_bat.groupby(['fullName'])['sixes'].sum().sort_values(ascending=False)[:10]
x = data1
y = data1.index
plt.title("No Of Sixes Hitted By Individual Players 2008-21")
plt.bar(y, x, color=cmix)
plt.xlabel('Batsman Name')
plt.ylabel('Total No Of 6 Hitted')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

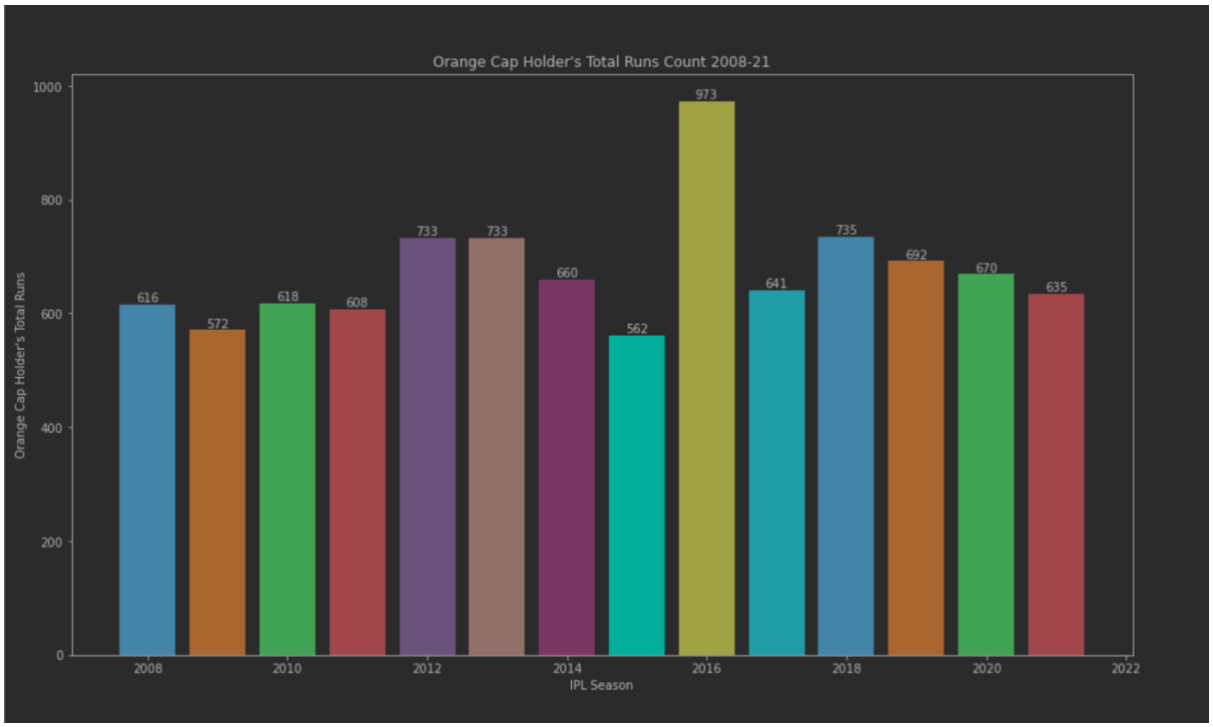
Most No Sixes Hitted By Individual Players in IPL 2008-21.

- **Chris Gayle – 357**
- AB de Villiers – 251
- Rohit Sharma – 227
- MS Dhoni – 219
- Kieron Pollard – 214
- Virat Kohli – 210
- Suresh Raina – 203
- David Warner – 201
- Shane Watson – 190
- Robin Uthappa – 168

9. All Time Maximum Run Scored By Orange Cap Winners In A Season from 2008-21

```
fig, ax = plt.subplots(figsize=(16, 9))
data2 = cleaned_bat.groupby(['season', 'fullName'])['runs'].sum().groupby('season').max()
x = data2
y = data2.index
plt.title("Orange Cap Holder's Total Runs Count 2008-21")
plt.bar(y, x, color=c10)
plt.ylabel("Orange Cap Holder's Total Runs")
plt.xlabel('IPL Season')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

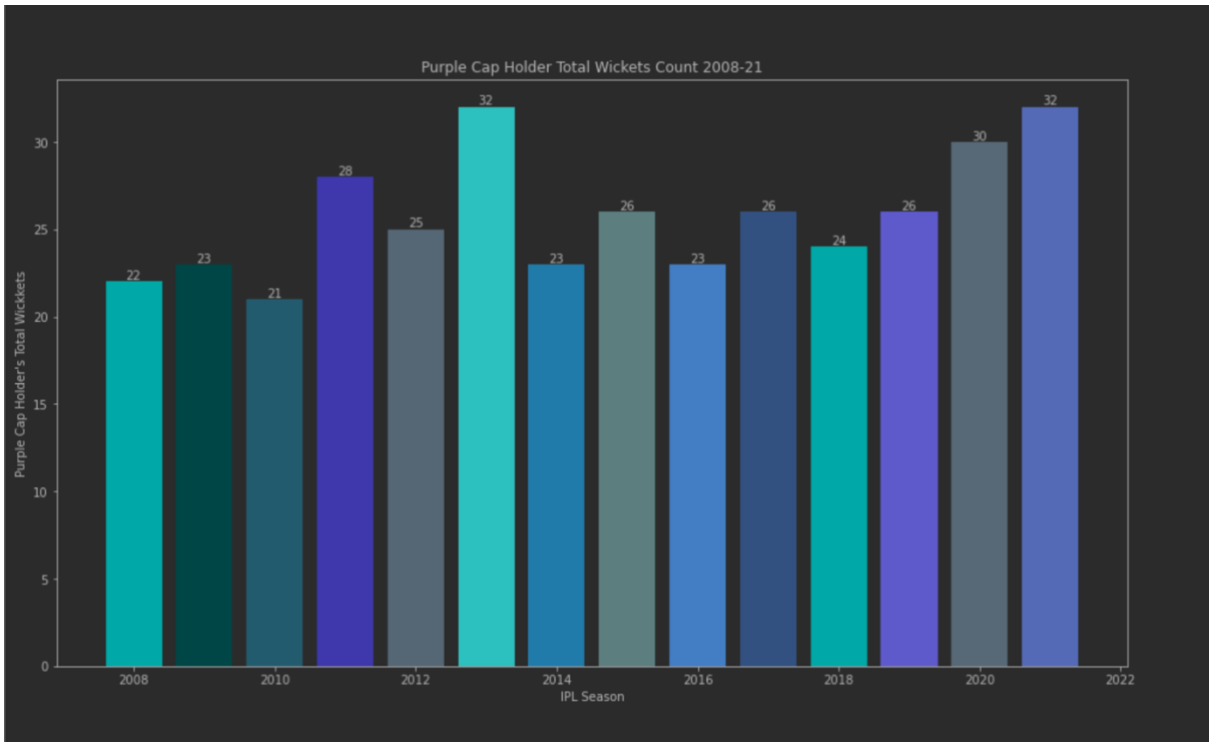
All Time Maximum Run Scored By Orange Cap Winners In A Season from 2008-21

- In **Season 2016** The Orange Cap Winner Scored Max Runs i.e. **Total of 973**.

10. All Time Maximum Wickets Taken By Purple Cap Winners In A Season from 2008-21

```
fig, ax = plt.subplots(figsize=(16, 9))
data2 = cleaned_ball.groupby(['season', 'fullName'])['wickets'].sum().groupby('season').max()
x = data2
y = data2.index
plt.title("Purple Cap Holder Total Wickets Count 2008-21")
plt.bar(y, x, color=cmix)
plt.ylabel("Purple Cap Holder's Total Wickkets")
plt.xlabel('IPL Season')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

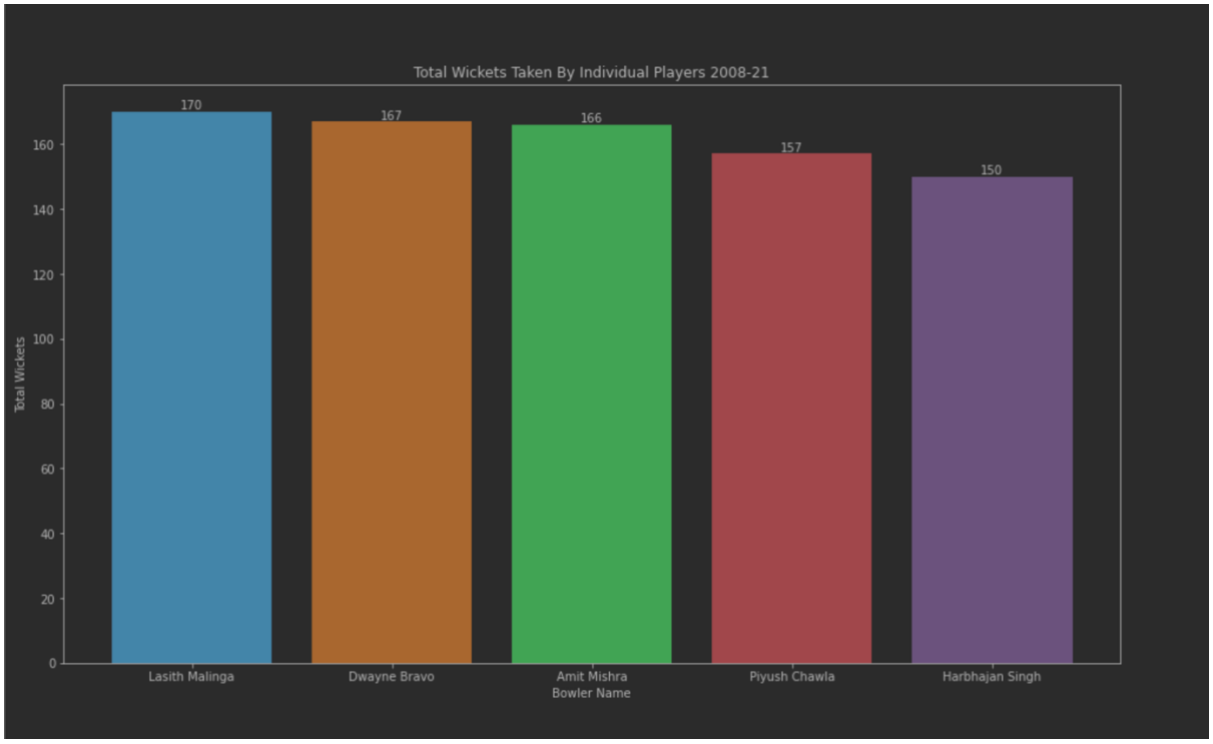
All Time Maximum Wickets Taken By Purple Cap Winners In A Season from 2008-21

- In **Season 2013** The Purple Cap Winner Taken Max Wickets i.e. **32**.
- In **Season 2021** also The Purple Cap Winner Taken Max Wickets i.e. **32**.

11. All Time Wickets Taken By Individual Players

```
fig, ax = plt.subplots(figsize=(16, 9))
data = cleaned_ball.groupby(['fullName'])['wickets'].sum().sort_values(ascending=False)[:5]
x = data
y = data.index
plt.title("Total Wickets Taken By Individual Players 2008-21")
plt.bar(y, x, color=c10)
plt.xlabel('Bowler Name')
plt.ylabel('Total Wickets')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

All Time Wickets Taken By Individual Players 2008-21

1 – Lasith Malinga [170 Wickets]

2 – Dwayne Bravo [167 Wickets]

3 – Amit Mishra [166 Wickets]

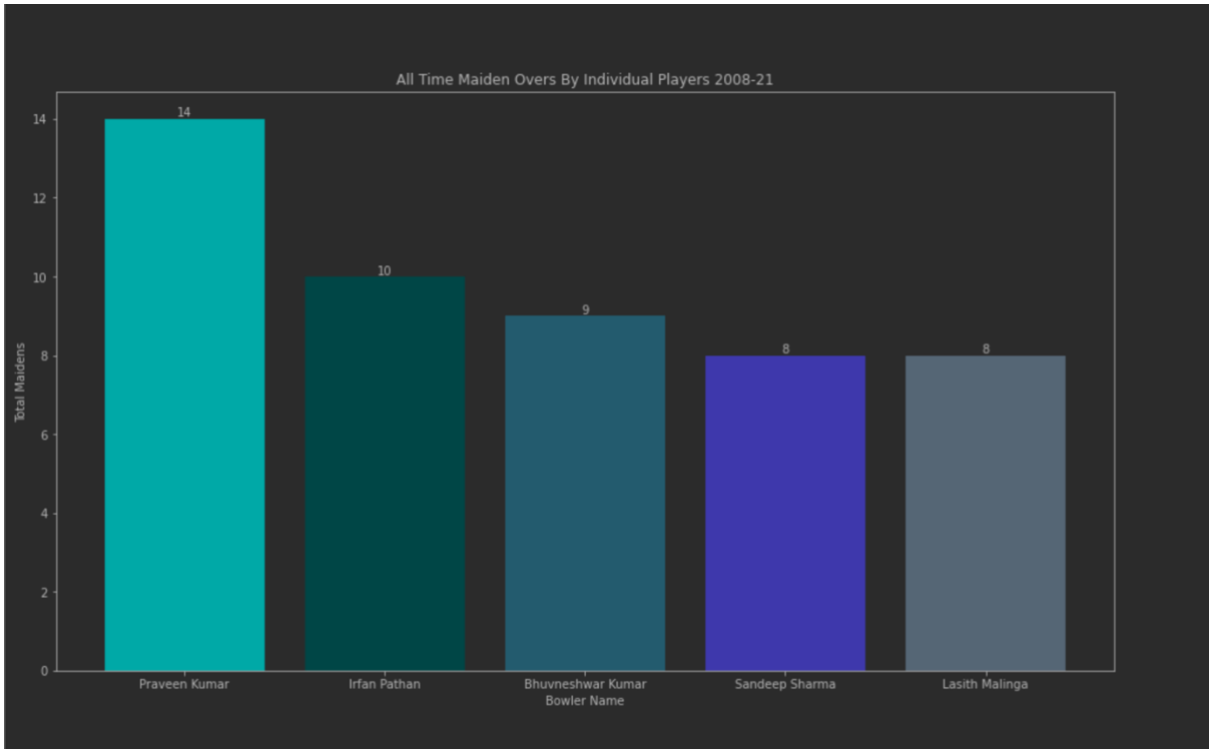
4 – Piyush Chawla [157 Wickets]

5 – Harbhajan Singh [150 Wickets]

12. All Time Maiden Overs By Individual Players

```
fig, ax = plt.subplots(figsize=(16, 9))
data = cleaned_ball.groupby(['fullName'])['maidens'].sum().sort_values(ascending=False)[:5]
x = data
y = data.index
plt.title("All Time Maiden Overs By Individual Players 2008-21")
plt.bar(y, x, color=cmix)
plt.xlabel('Bowler Name')
plt.ylabel('Total Maidens')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

All Time **Maiden Overs** By Individual Players

1 - Praveen Kumar [14 Maidens]

2 - Irfan Pathan [10 Maidens]

3 - Bhuvneshwar Kumar [09 Maidens]

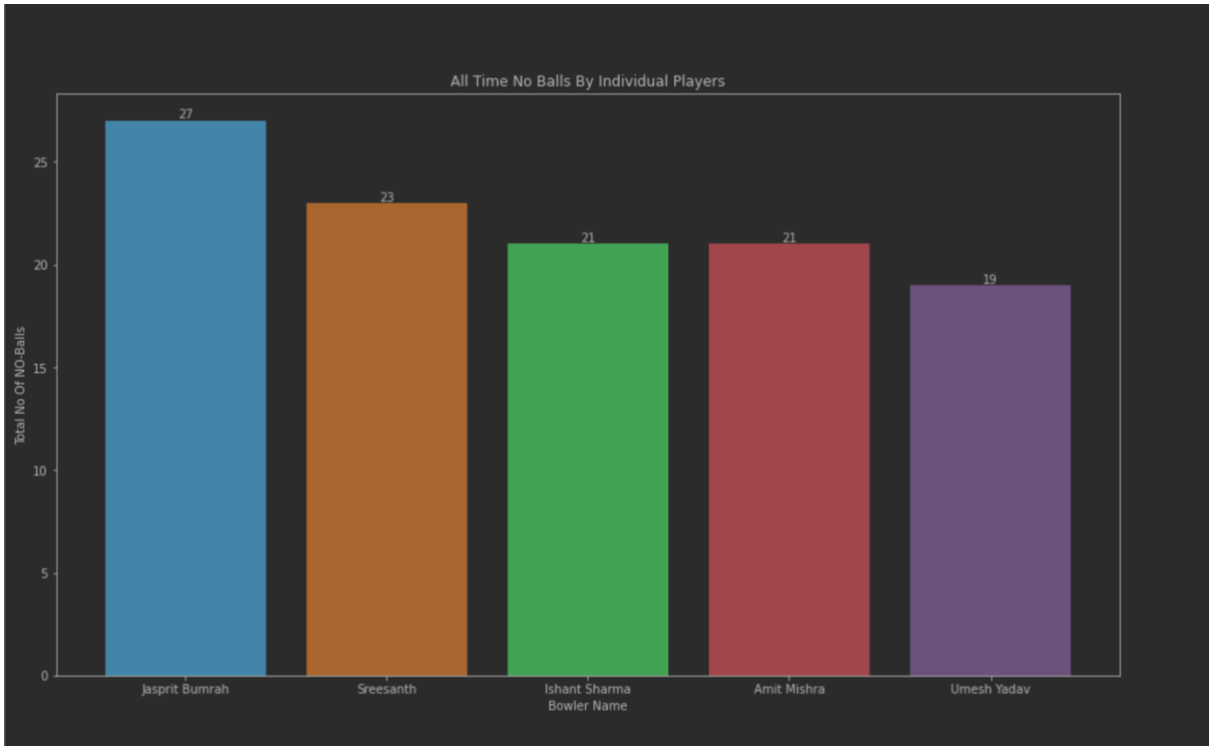
4 - Sandeep Sharma [08 Maidens]

5 - Lasith Malinga [08 Maidens]

13. All Time No Balls By Individual Players

```
fig, ax = plt.subplots(figsize=(16, 9))
data = cleaned_ball.groupby(['fullName'])['noballs'].sum().sort_values(ascending=False)[:5]
x = data
y = data.index
plt.title("All Time No Balls By Individual Players")
plt.bar(y, x, color=c10)
plt.xlabel('Bowler Name')
plt.ylabel('Total No Of NO-Balls')
for bars in ax.containers:
ax.bar_label(bars)
plt.show()
```

Output:



Conclusion:

All Time No Balls By Individual Players

1 – Jasprit Bumrah [27 NoBalls]

2 – Sreesanth [23 NoBalls]

3 – Ishant Sharma [21 NoBalls]

4 – Amit Mishra [21 NoBalls]

5 – Umesh Yadav [19 NoBalls]