

```
In [69]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as ple
```

## Data Loading

```
In [70]: ipl=pd.read_csv("ipl_2025_auction_players.csv")
```

## Data OverView

```
In [71]: ipl.head()
```

Out[71]:

	Players	Team	Type	Base	Sold
0	Virat Kohli	RCB	BAT	-	21.00
1	Rajat Patidar	RCB	BAT	-	11.00
2	Yash Dayal	RCB	BOWL	-	5.00
3	Jaspri Bumrah	MI	BOWL	-	18.00
4	Suryakumar Yadav	MI	BAT	-	16.35

## Exploratory Data Analysis (EDA)¶

**Let's start with some basic exploratory data analysis to understand the distributions and relationships in the data.**

```
In [72]: ipl.shape
```

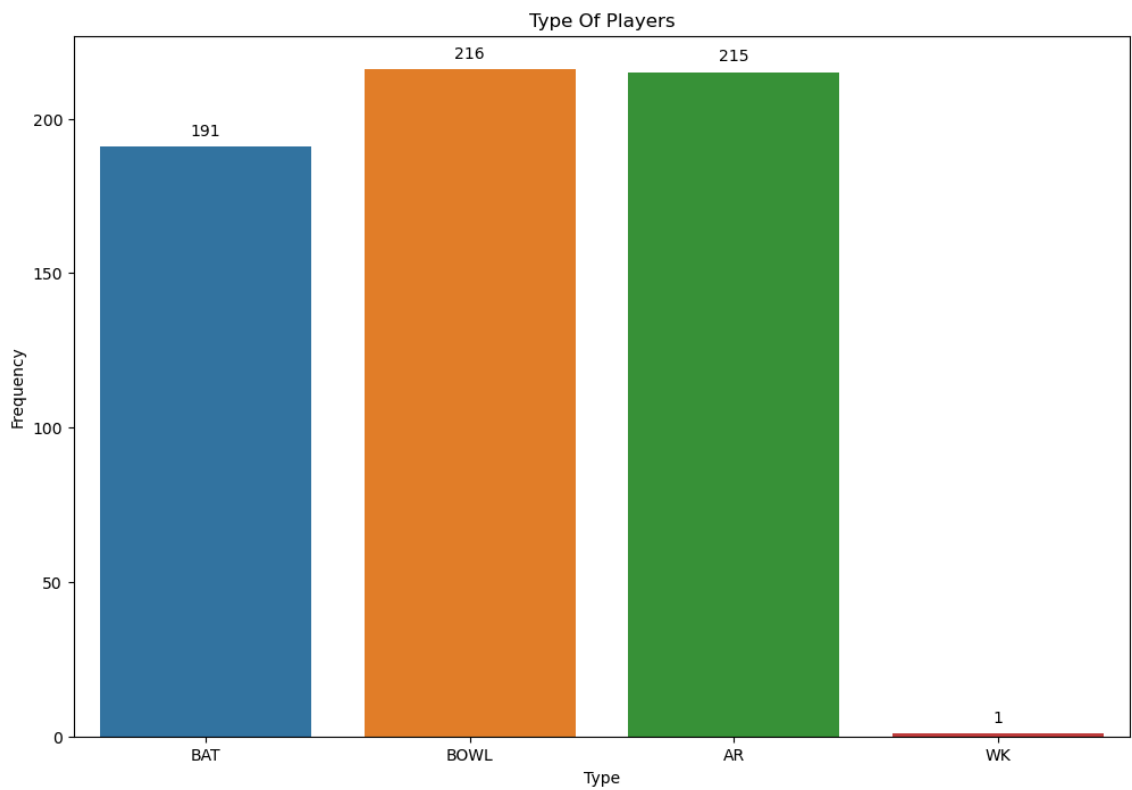
Out[72]: (623, 5)

```
In [73]: ipl.isna().sum()
```

Out[73]: Players 0  
Team 0  
Type 0  
Base 0  
Sold 0  
dtype: int64

```
In [74]: plt.figure(figsize=(12,8))
ax=sns.countplot(data=ipl,x='Type')
# Add data labels on top of the bars
for p in ax.patches:
    ax.annotate(
        f'{int(p.get_height())}',
        (p.get_x() + p.get_width() / 2., p.get_height()),
        ha='center',
        va='bottom',
        fontsize=10,
        color='black',
        xytext=(0, 5),
        textcoords='offset points'
    )

# Add titles and Labels
plt.title('Type Of Players')
plt.xlabel('Type')
plt.ylabel('Frequency')
plt.show()
```



```
In [75]: # Ensure 'Sold' is numeric
ipl['Sold'] = pd.to_numeric(ipl['Sold'], errors='coerce')

# Check for any issues in data
if ipl['Sold'].isnull().any():
    print("Warning: Non-numeric values found in 'Sold'.")
    print(ipl[ipl['Sold'].isnull()]) # Display rows with issues

# Find the highest and lowest sold amounts
max_sold = ipl['Sold'].max()
min_sold = ipl['Sold'].min()

# Get player(s) with highest and lowest sold amounts
highest_player = ipl[ipl['Sold'] == max_sold]
lowest_player = ipl[ipl['Sold'] == min_sold]

# Print the results
print("Highest Sold Player(s):")
print(highest_player)
print("\nLowest Sold Player(s):")
print(lowest_player)
```

Warning: Non-numeric values found in 'Sold'.

	Players	Team	Type	Base	Sold
64	David Warner	-	BAT	2.00	NaN
72	Jonny Bairstow	-	BAT	2.00	NaN
88	Waqar Salamkheil	-	BOWL	0.75	NaN
91	Yash Dhull	-	BAT	0.30	NaN
95	Anmolpreet Singh	-	BAT	0.30	NaN
..	...	...	...	...	...
617	Tripuresh Singh	-	AR	0.30	NaN
618	Vijay Yadav	-	AR	0.30	NaN
620	Saurabh Netravalkar	-	BOWL	0.30	NaN
621	Hardik Tamore	-	WK	0.30	NaN
622	Daryl Mitchell	-	AR	2.00	NaN

[396 rows x 5 columns]

Highest Sold Player(s):

	Players	Team	Type	Base	Sold
48	Rishabh Pant	LSG	BAT	2.00	27.0

Lowest Sold Player(s):

	Players	Team	Type	Base	Sold
96	Atharva Taide	SRH	BAT	0.30	0.3
105	Nishant Sindhu	GT	AR	0.30	0.3
107	Aryan Juyal	LSG	BAT	0.30	0.3
110	Anuj Rawat	GT	BAT	0.30	0.3
111	Luvnith Sisodia	KKR	BAT	0.30	0.3
123	Shreyas Gopal	CSK	BOWL	0.30	0.3
124	Mayank Markande	KKR	BOWL	0.30	0.3
127	Kumar Kartikeya	RR	BOWL	0.30	0.3
128	Manav Suthar	GT	BOWL	0.30	0.3
163	Swastik Chikara	RCB	BAT	0.30	0.3
168	Shaik Rasheed	CSK	BAT	0.30	0.3
169	Himmat Singh	LSG	BAT	0.30	0.3
173	Darshan Nalkande	DC	AR	0.30	0.3
183	Kunal Singh Rathore	RR	BAT	0.30	0.3
186	Mukesh Choudhary	CSK	BOWL	0.30	0.3
191	Arjun Tendulkar	MI	BOWL	0.30	0.3
196	Digvesh Singh	LSG	BOWL	0.30	0.3
238	Sachin Baby	SRH	BAT	0.30	0.3
240	Harnoor Singh	PBKS	BAT	0.30	0.3
243	C Andre Siddarth	CSK	BAT	0.30	0.3
248	Rajvardhan Hangargekar	LSG	AR	0.30	0.3
250	Arshin Kulkarni	LSG	AR	0.30	0.3
262	Ashwani Kumar	MI	BOWL	0.30	0.3
265	Akash Singh	LSG	BOWL	0.30	0.3
315	Manoj Bhandage	RCB	AR	0.30	0.3
316	Praveen Dubey	PBKS	AR	0.30	0.3
317	Ajay Mandal	DC	AR	0.30	0.3
326	Krishnan Shrijith	MI	BAT	0.30	0.3
341	Mohit Rathee	RCB	BOWL	0.30	0.3
370	Aniket Verma	SRH	BAT	0.30	0.3
374	Raj Bawa	MI	AR	0.30	0.3
376	Musheer Khan	PBKS	AR	0.30	0.3
377	Manvanth Kumar	DC	AR	0.30	0.3
379	Suryansh Shedge	PBKS	AR	0.30	0.3
391	Kulwant Khejroliya	GT	BOWL	0.30	0.3
395	Prince Yadav	LSG	BOWL	0.30	0.3
430	Kamlesh Nagarkoti	CSK	AR	0.30	0.3
446	Abhinandan Singh	RCB	BOWL	0.30	0.3
461	Pyla Avinash	PBKS	BAT	0.30	0.3
470	Ramakrishna Ghosh	CSK	AR	0.30	0.3

479	Satyanarayana Raju	MI	BOWL	0.30	0.3
497	Bevon Jacobs	MI	BAT	0.30	0.3
514	Ashok Sharma	RR	BOWL	0.30	0.3
601	Tripurana Vijay	DC	AR	0.30	0.3
616	Vignesh Puthur	MI	AR	0.30	0.3

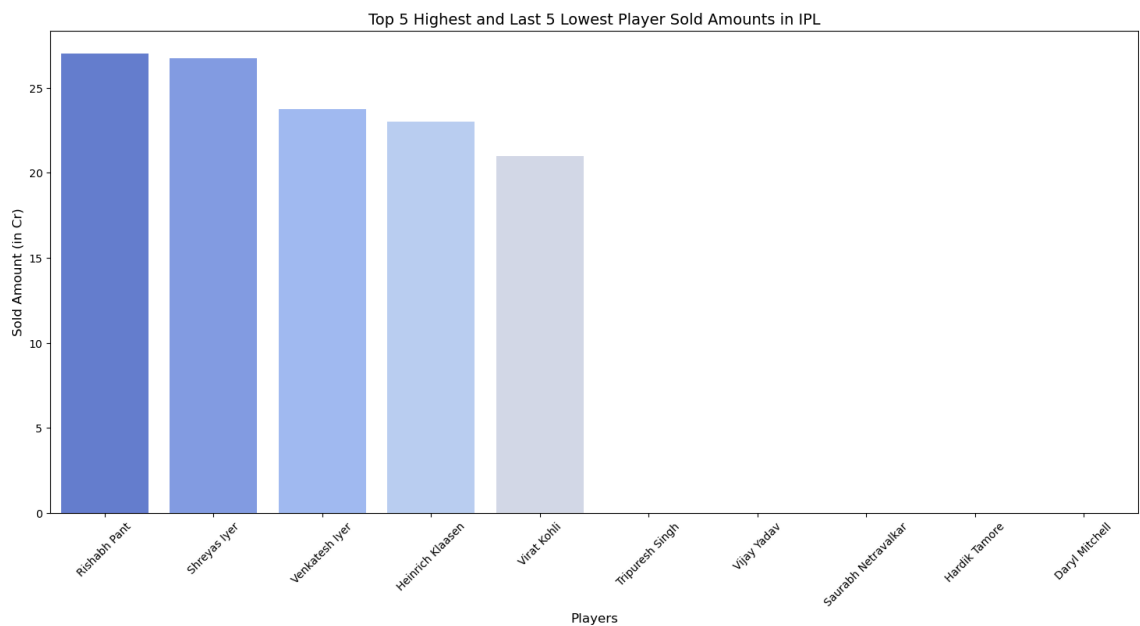
```
In [76]: # Sort data by 'Sold' amounts
sorted_ipl = ipl.sort_values(by='Sold', ascending=False)

# Select top 5 highest and last 5 lowest sold amounts
top_5 = sorted_ipl.head(5)
bottom_5 = sorted_ipl.tail(5)

# Combine top 5 and bottom 5 into a single DataFrame
filtered_ipl = pd.concat([top_5, bottom_5])

# Visualization
plt.figure(figsize=(18, 8))
ax = sns.barplot(data=filtered_ipl, x='Players', y='Sold', palette='coolwarm')

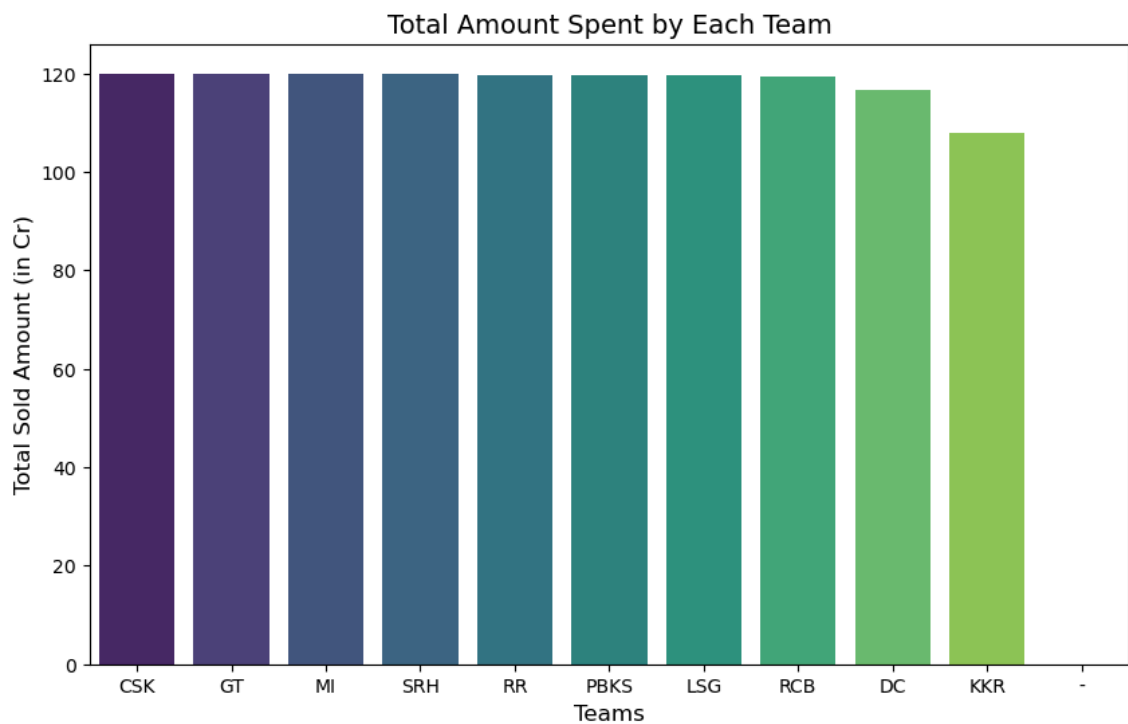
# Titles and Labels
plt.title('Top 5 Highest and Last 5 Lowest Player Sold Amounts in IPL', fontweight='bold')
plt.xlabel('Players', fontsize=12)
plt.ylabel('Sold Amount (in Cr)', fontsize=12)
plt.xticks(rotation=45) # Rotate player names for better visibility
plt.show()
```



```
In [77]: # Calculate total amount spent by each team
team_spending = ipl.groupby('Team')['Sold'].sum().reset_index()

# Sort teams by spending for better visualization
team_spending = team_spending.sort_values(by='Sold', ascending=False)

# Visualization
plt.figure(figsize=(10, 6))
sns.barplot(data=team_spending, x='Team', y='Sold', palette='viridis')
# Titles and Labels
plt.title('Total Amount Spent by Each Team', fontsize=14)
plt.xlabel('Teams', fontsize=12)
plt.ylabel('Total Sold Amount (in Cr)', fontsize=12)
plt.show()
```



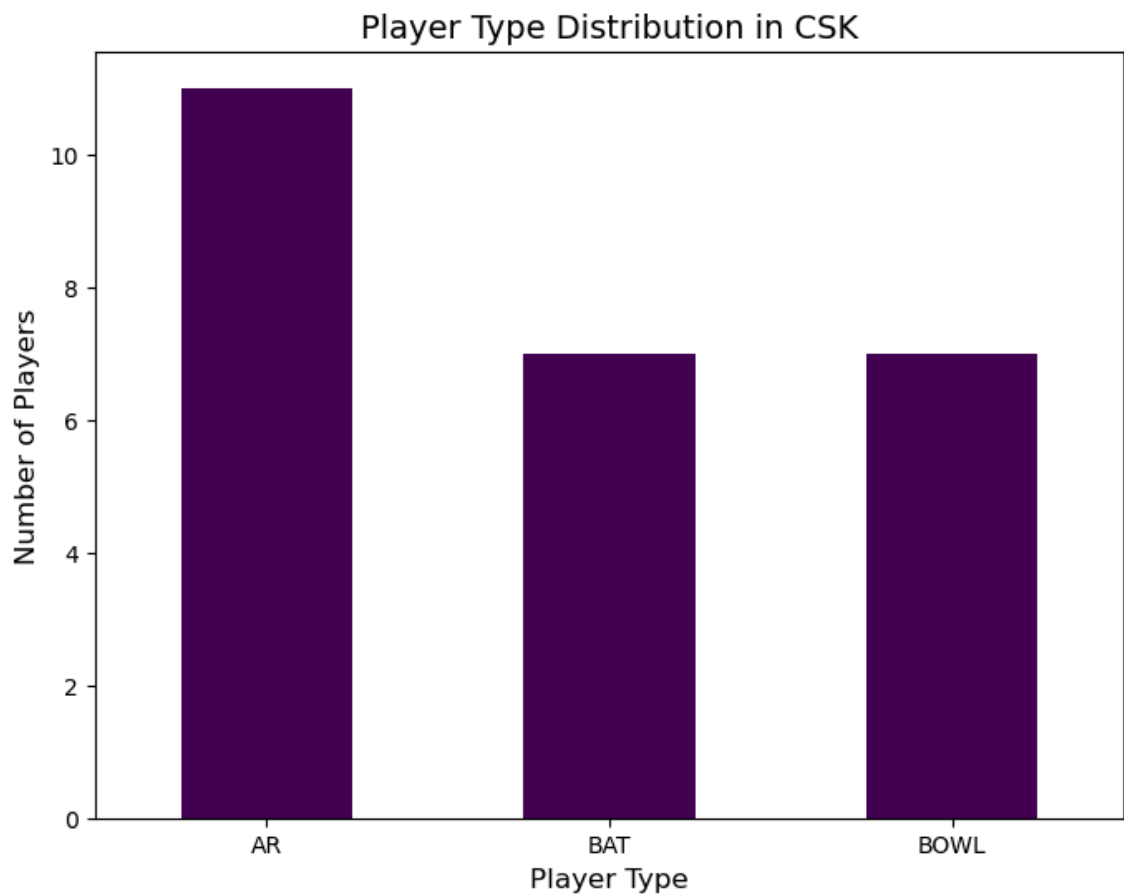
```
In [78]: # Specify the team you want to focus on
team_name = 'CSK'

# Filter data for the selected team
team_data = ipl[ipl['Team'] == team_name]

# Count the number of each player type in the selected team
player_type_count = team_data.groupby('Type').size()

# Plot the result
plt.figure(figsize=(8, 6))
player_type_count.plot(kind='bar', stacked=True, colormap='viridis')

# Add chart details
plt.title(f'Player Type Distribution in {team_name}', fontsize=14)
plt.xlabel('Player Type', fontsize=12)
plt.ylabel('Number of Players', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```



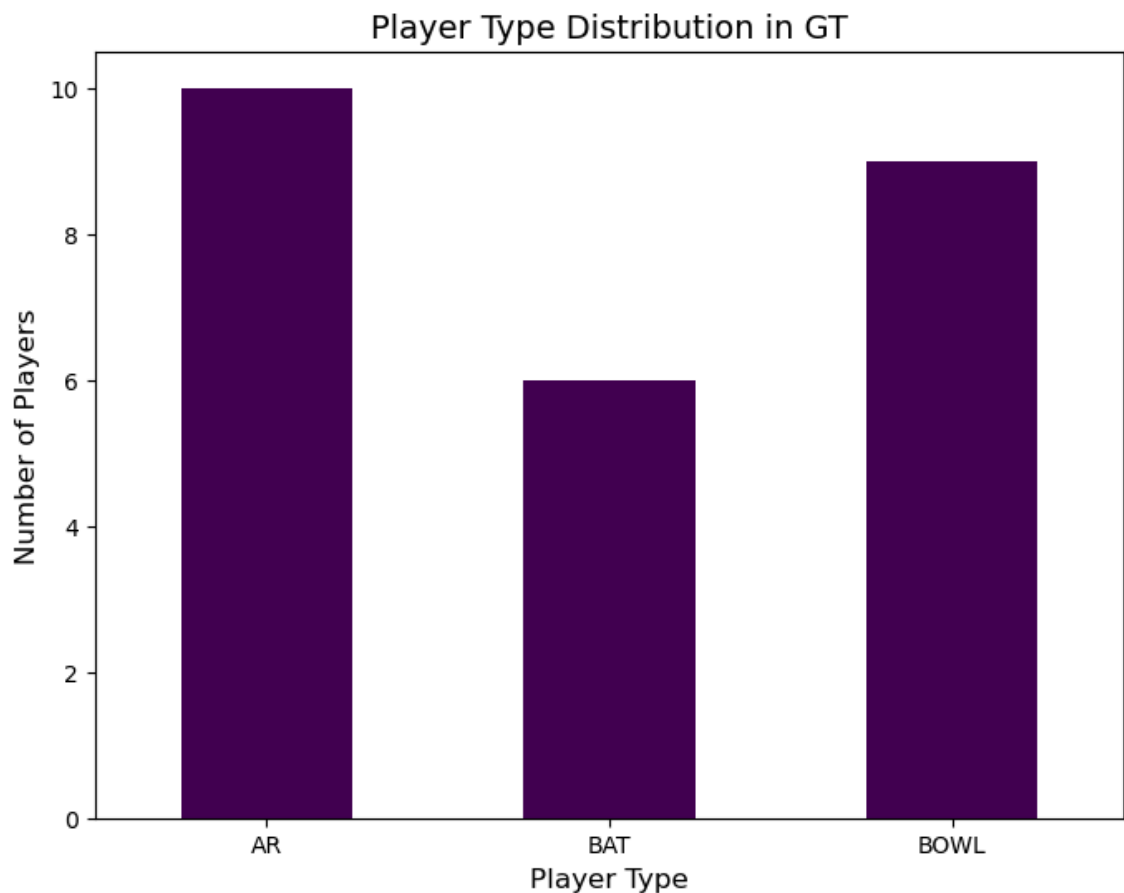
```
In [79]: # Specify the team you want to focus on
team_name = 'GT'

# Filter data for the selected team
team_data = ipl[ipl['Team'] == team_name]

# Count the number of each player type in the selected team
player_type_count = team_data.groupby('Type').size()

# Plot the result
plt.figure(figsize=(8, 6))
player_type_count.plot(kind='bar', stacked=True, colormap='viridis')

# Add chart details
plt.title(f'Player Type Distribution in {team_name}', fontsize=14)
plt.xlabel('Player Type', fontsize=12)
plt.ylabel('Number of Players', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```





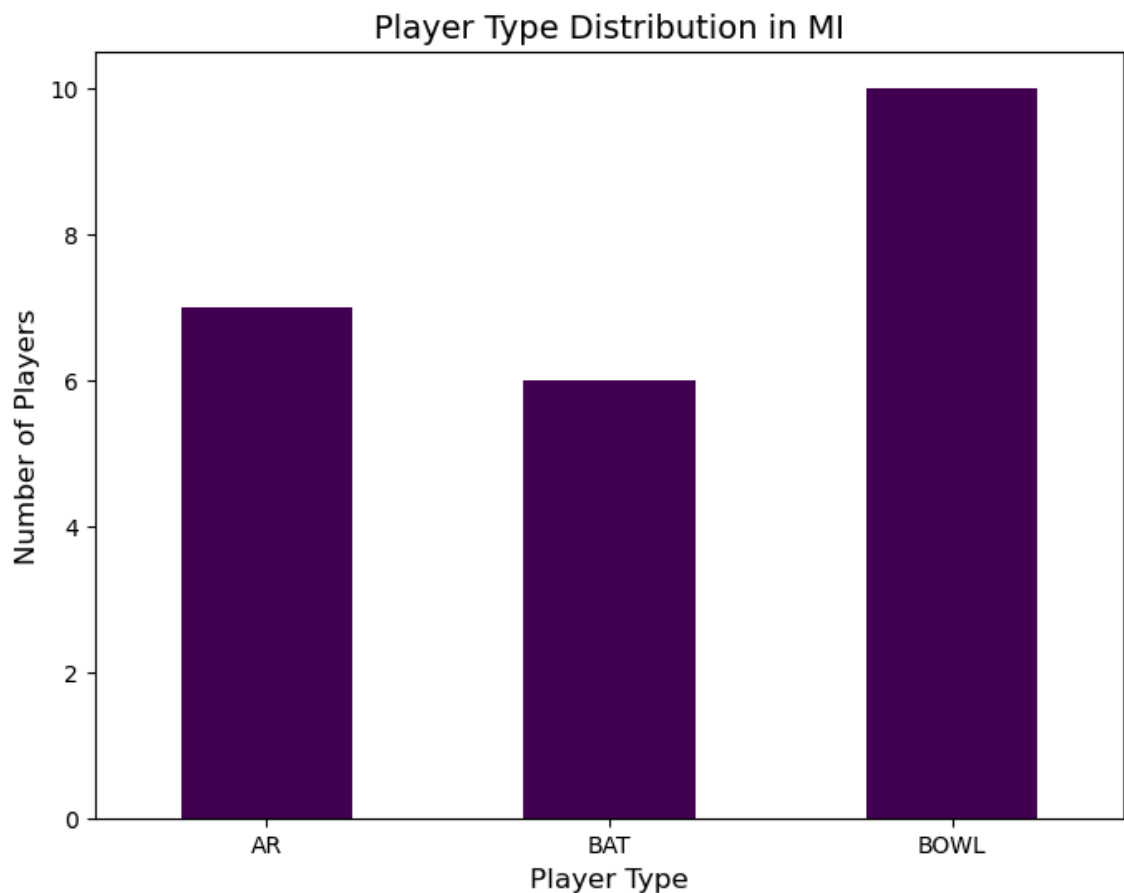
```
In [80]: # Specify the team you want to focus on
team_name = 'MI'

# Filter data for the selected team
team_data = ipl[ipl['Team'] == team_name]

# Count the number of each player type in the selected team
player_type_count = team_data.groupby('Type').size()

# Plot the result
plt.figure(figsize=(8, 6))
player_type_count.plot(kind='bar', stacked=True, colormap='viridis')

# Add chart details
plt.title(f'Player Type Distribution in {team_name}', fontsize=14)
plt.xlabel('Player Type', fontsize=12)
plt.ylabel('Number of Players', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```



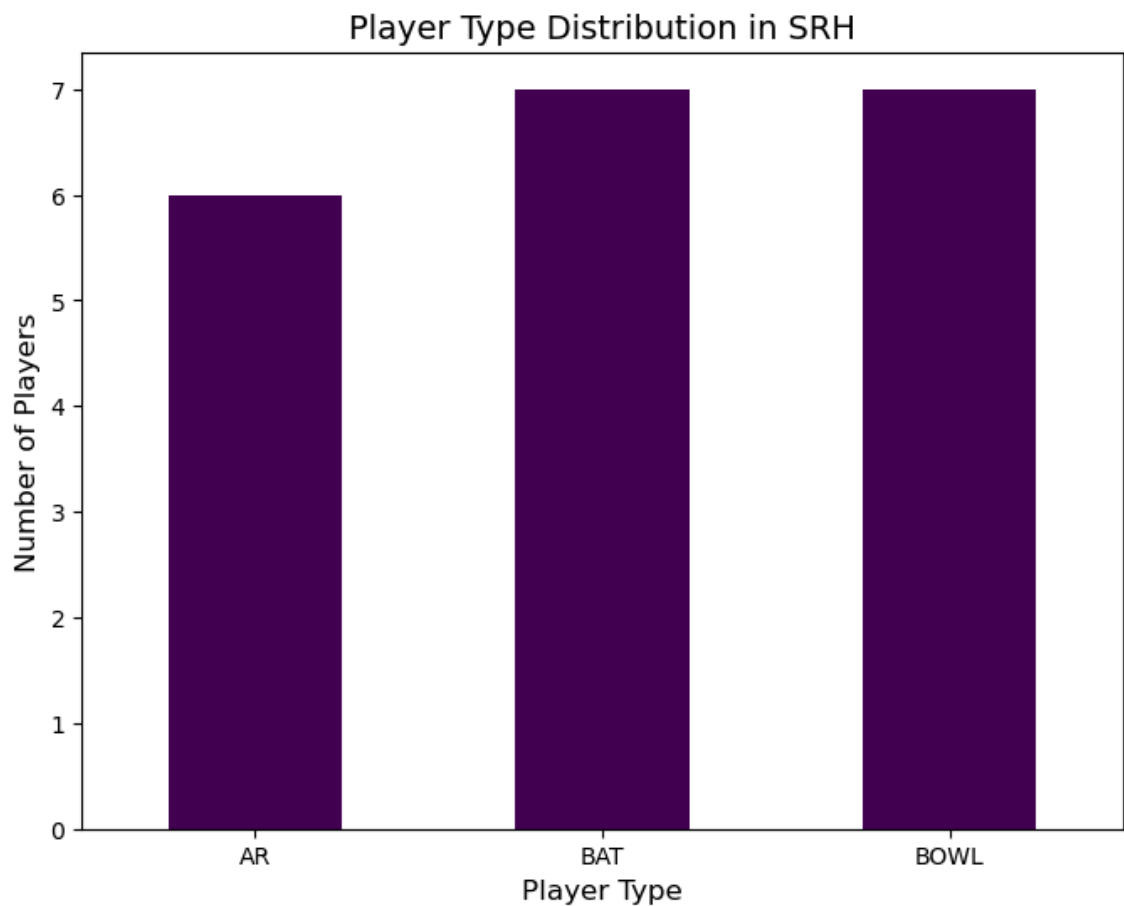
```
In [81]: # Specify the team you want to focus on
team_name = 'SRH'

# Filter data for the selected team
team_data = ipl[ipl['Team'] == team_name]

# Count the number of each player type in the selected team
player_type_count = team_data.groupby('Type').size()

# Plot the result
plt.figure(figsize=(8, 6))
player_type_count.plot(kind='bar', stacked=True, colormap='viridis')

# Add chart details
plt.title(f'Player Type Distribution in {team_name}', fontsize=14)
plt.xlabel('Player Type', fontsize=12)
plt.ylabel('Number of Players', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```



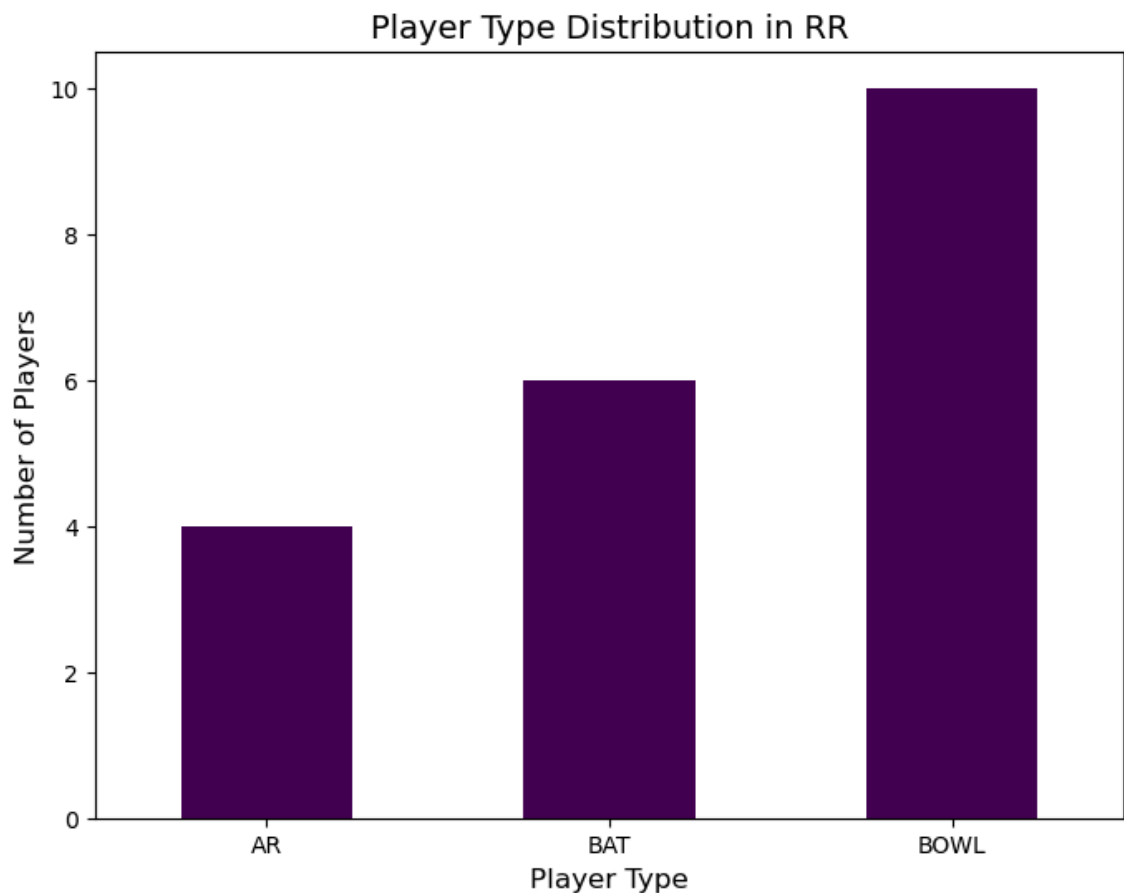
```
In [82]: # Specify the team you want to focus on
team_name = 'RR'

# Filter data for the selected team
team_data = ipl[ipl['Team'] == team_name]

# Count the number of each player type in the selected team
player_type_count = team_data.groupby('Type').size()

# Plot the result
plt.figure(figsize=(8, 6))
player_type_count.plot(kind='bar', stacked=True, colormap='viridis')

# Add chart details
plt.title(f'Player Type Distribution in {team_name}', fontsize=14)
plt.xlabel('Player Type', fontsize=12)
plt.ylabel('Number of Players', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```



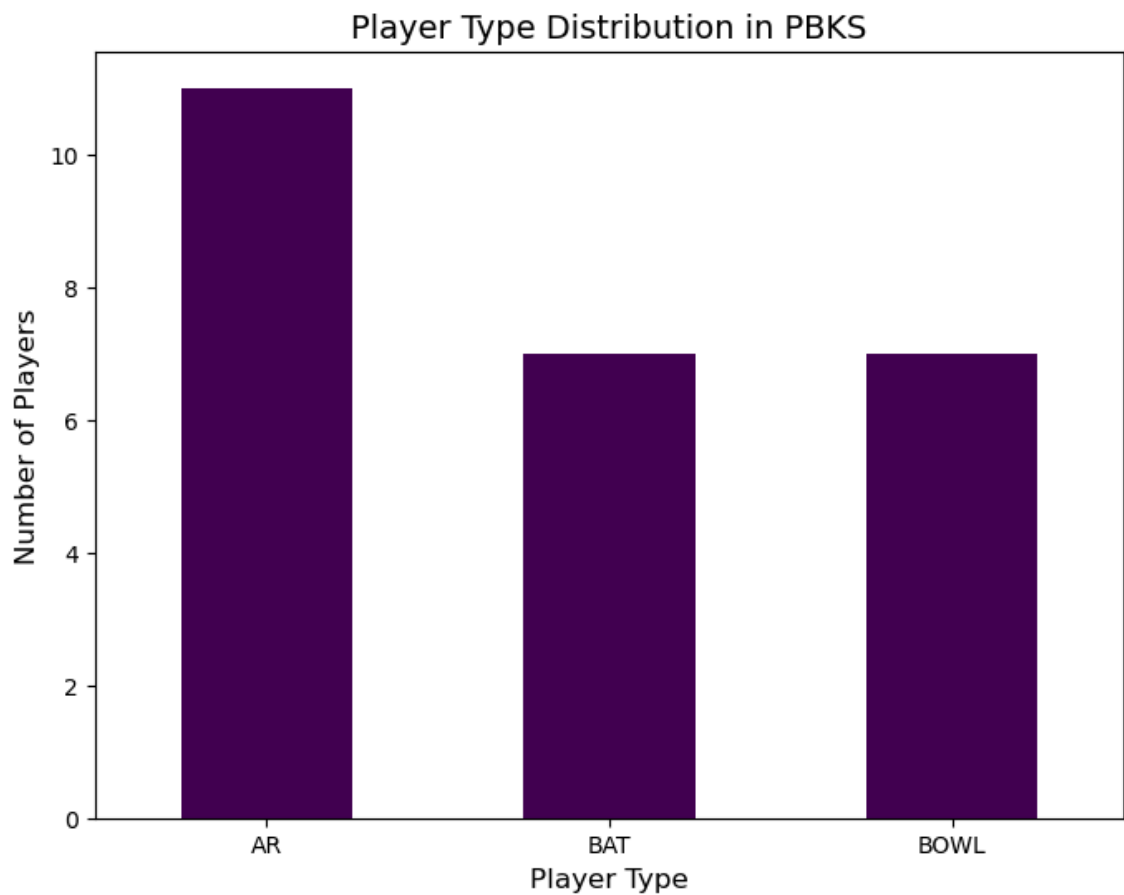
```
In [83]: # Specify the team you want to focus on
team_name = 'PBKS'

# Filter data for the selected team
team_data = ipl[ipl['Team'] == team_name]

# Count the number of each player type in the selected team
player_type_count = team_data.groupby('Type').size()

# Plot the result
plt.figure(figsize=(8, 6))
player_type_count.plot(kind='bar', stacked=True, colormap='viridis')

# Add chart details
plt.title(f'Player Type Distribution in {team_name}', fontsize=14)
plt.xlabel('Player Type', fontsize=12)
plt.ylabel('Number of Players', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```



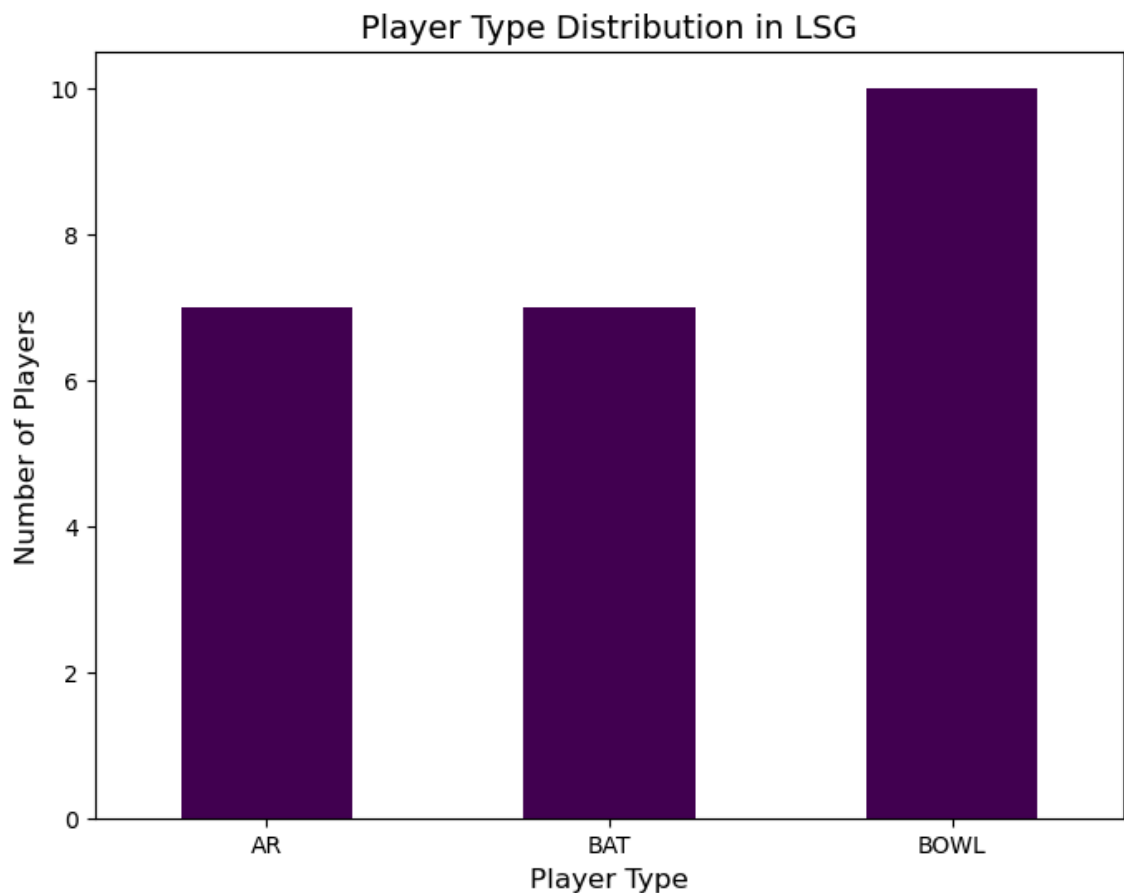
```
In [84]: # Specify the team you want to focus on
team_name = 'LSG'

# Filter data for the selected team
team_data = ipl[ipl['Team'] == team_name]

# Count the number of each player type in the selected team
player_type_count = team_data.groupby('Type').size()

# Plot the result
plt.figure(figsize=(8, 6))
player_type_count.plot(kind='bar', stacked=True, colormap='viridis')

# Add chart details
plt.title(f'Player Type Distribution in {team_name}', fontsize=14)
plt.xlabel('Player Type', fontsize=12)
plt.ylabel('Number of Players', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```



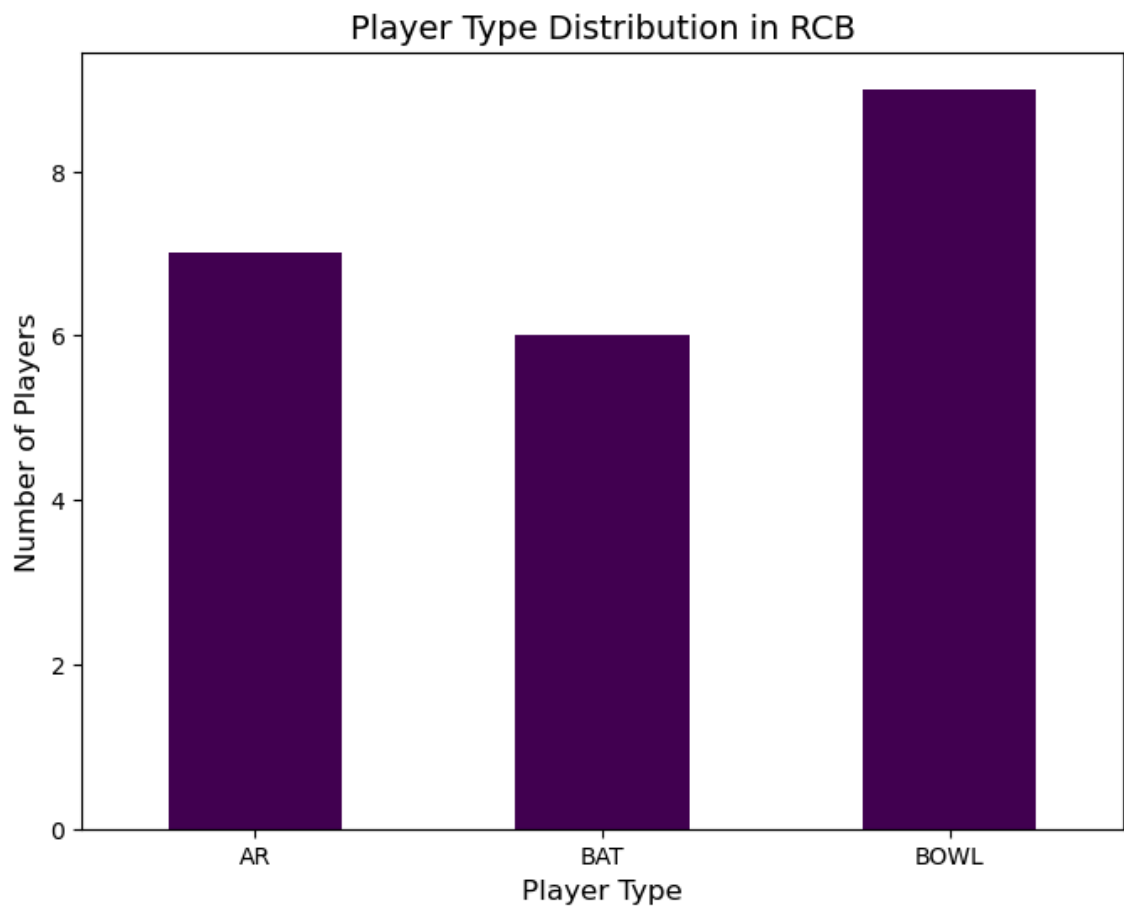
```
In [85]: # Specify the team you want to focus on
team_name = 'RCB'

# Filter data for the selected team
team_data = ipl[ipl['Team'] == team_name]

# Count the number of each player type in the selected team
player_type_count = team_data.groupby('Type').size()

# Plot the result
plt.figure(figsize=(8, 6))
player_type_count.plot(kind='bar', stacked=True, colormap='viridis')

# Add chart details
plt.title(f'Player Type Distribution in {team_name}', fontsize=14)
plt.xlabel('Player Type', fontsize=12)
plt.ylabel('Number of Players', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```



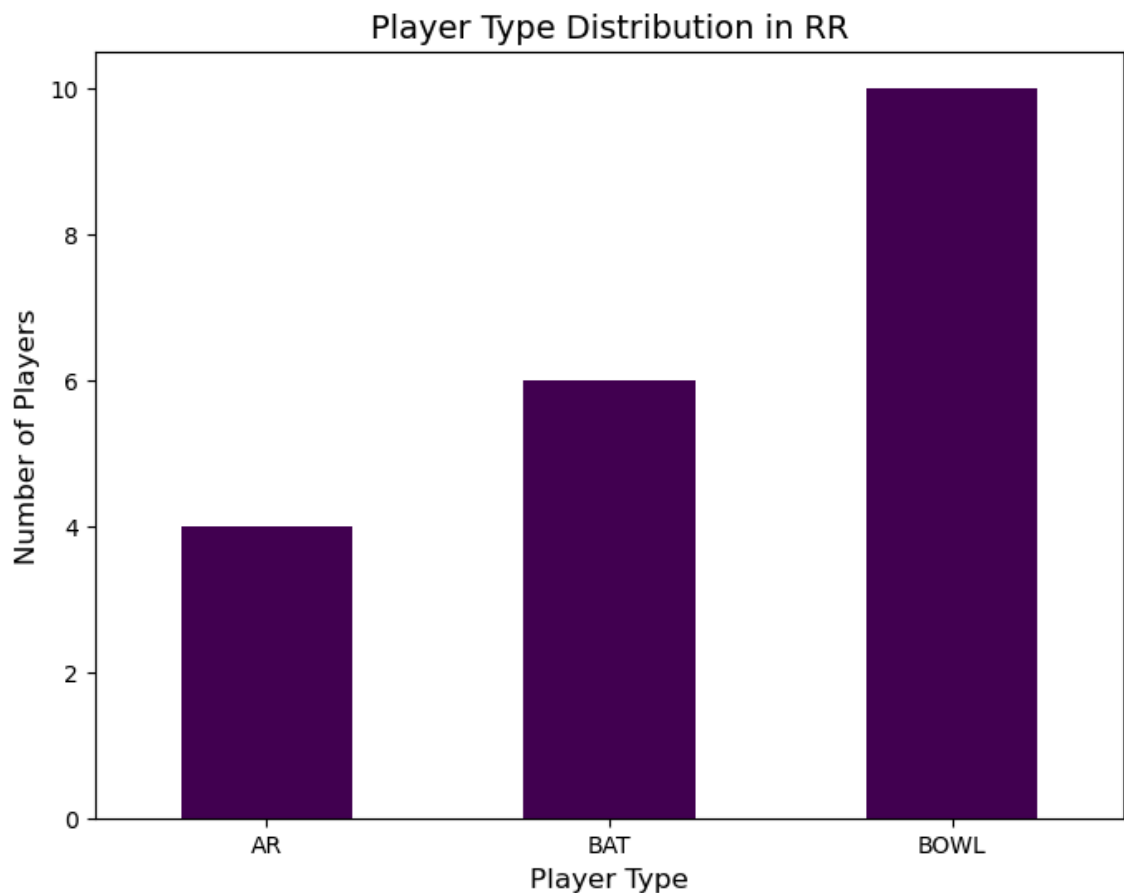
```
In [86]: # Specify the team you want to focus on
team_name = 'RR'

# Filter data for the selected team
team_data = ipl[ipl['Team'] == team_name]

# Count the number of each player type in the selected team
player_type_count = team_data.groupby('Type').size()

# Plot the result
plt.figure(figsize=(8, 6))
player_type_count.plot(kind='bar', stacked=True, colormap='viridis')

# Add chart details
plt.title(f'Player Type Distribution in {team_name}', fontsize=14)
plt.xlabel('Player Type', fontsize=12)
plt.ylabel('Number of Players', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```



```

In [87]: # Filter the dataset for batsmen only
batsmen_data = ipl[ipl['Type'] == 'BAT']

# Drop rows with NaN values in the 'Sold' column to avoid issues with idxmax
batsmen_data = batsmen_data.dropna(subset=['Sold'])

# Find the highest sold batsman in each team
highest_sold_batsman = batsmen_data.loc[batsmen_data.groupby('Team')['Sold'].idxmax()]

# Display the result
print(highest_sold_batsman[['Team', 'Players', 'Sold']])

# Set the figure size explicitly with reasonable values
plt.figure(figsize=(10, 6)) # Use 10x6 inches for the chart

# Create bar chart for highest sold batsmen
bars = plt.bar(highest_sold_batsman['Team'], highest_sold_batsman['Sold'],

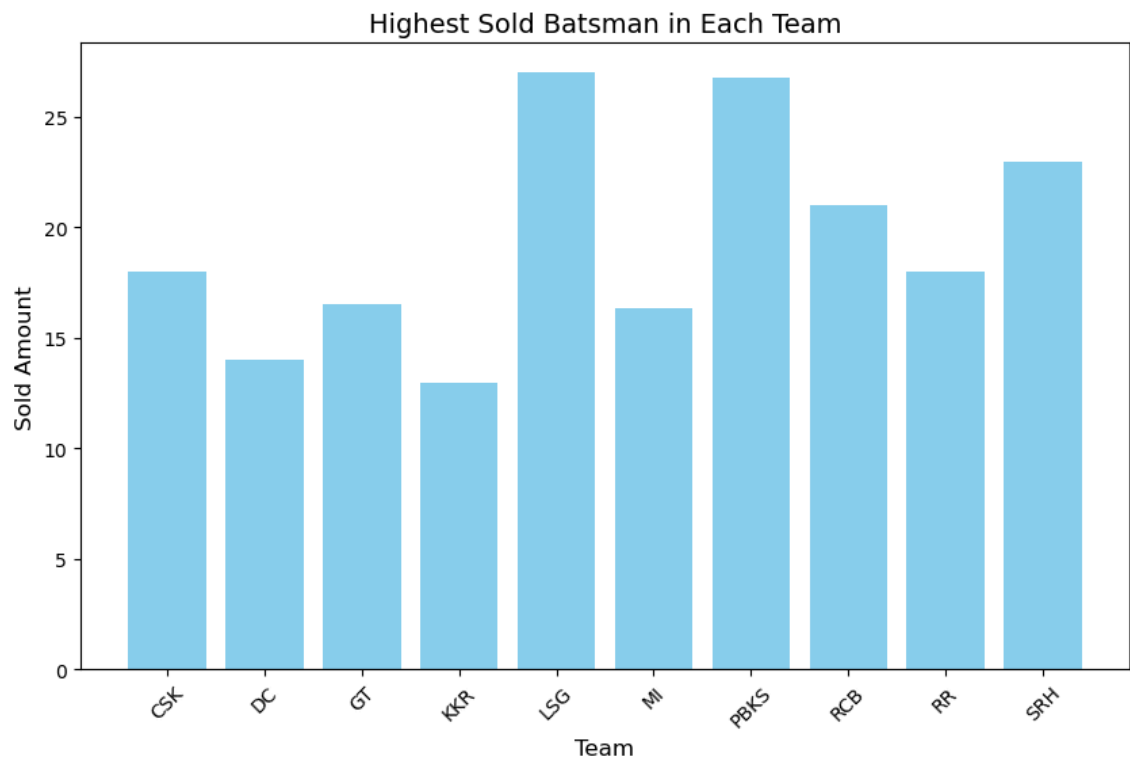
# Add chart details
plt.title('Highest Sold Batsman in Each Team', fontsize=14)
plt.xlabel('Team', fontsize=12)
plt.ylabel('Sold Amount', fontsize=12)
plt.xticks(rotation=45)

# Show the plot
plt.show()

```

	Team	Players	Sold
13	CSK	Ruturaj Gaikwad	18.00
55	DC	KL Rahul	14.00
35	GT	Shubman Gill	16.50
22	KKR	Rinku Singh	13.00
48	LSG	Rishabh Pant	27.00
4	MI	Suryakumar Yadav	16.35
47	PBKS	Shreyas Iyer	26.75
0	RCB	Virat Kohli	21.00
28	RR	Sanju Samson	18.00
8	SRH	Heinrich Klaasen	23.00





```

In [88]: # Filter the dataset for batsmen only
batsmen_data = ipl[ipl['Type'] == 'BOWL']

# Drop rows with NaN values in the 'Sold' column to avoid issues with idxmax
batsmen_data = batsmen_data.dropna(subset=['Sold'])

# Find the highest sold batsman in each team
highest_sold_batsman = batsmen_data.loc[batsmen_data.groupby('Team')['Sold'].idxmax()]

# Display the result
print(highest_sold_batsman[['Team', 'Players', 'Sold']])

# Set the figure size explicitly with reasonable values
plt.figure(figsize=(10, 6)) # Use 10x6 inches for the chart

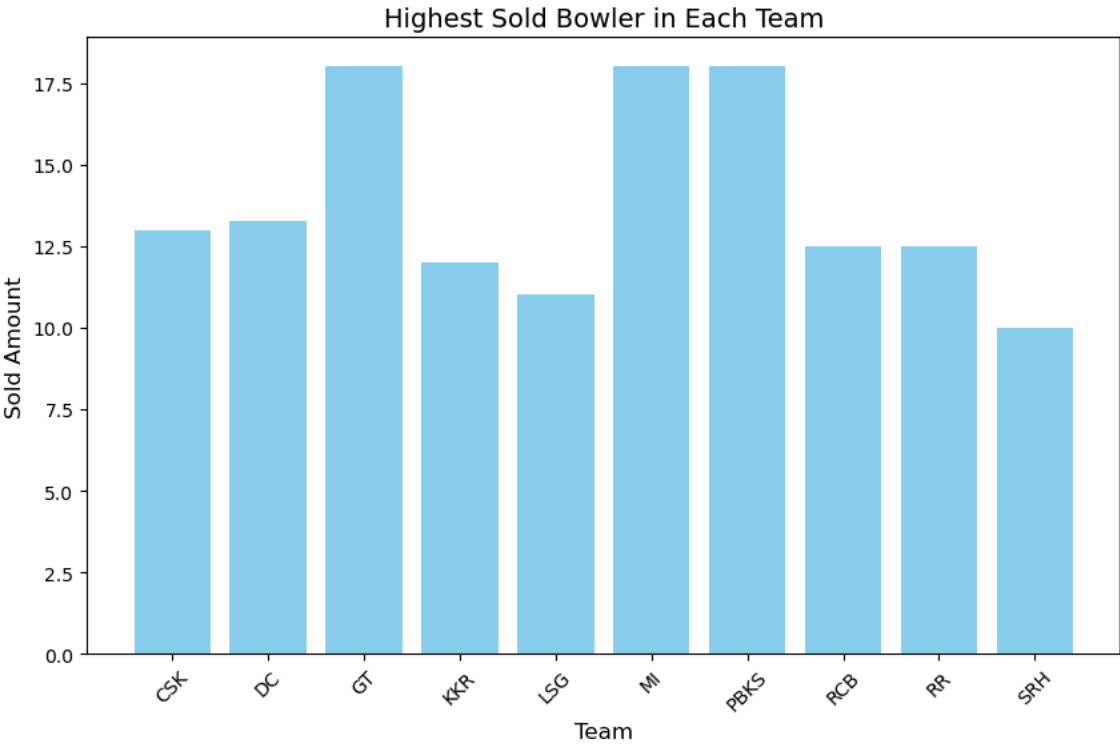
# Create bar chart for highest sold batsmen
bars = plt.bar(highest_sold_batsman['Team'], highest_sold_batsman['Sold'],

# Add chart details
plt.title('Highest Sold Bowler in Each Team', fontsize=14)
plt.xlabel('Team', fontsize=12)
plt.ylabel('Sold Amount', fontsize=12)
plt.xticks(rotation=45)

# Show the plot
plt.show()

```

	Team	Players	Sold
15	CSK	Matheesha Pathirana	13.00
19	DC	Kuldeep Yadav	13.25
34	GT	Rashid Khan	18.00
24	KKR	Sunil Narine	12.00
40	LSG	Ravi Bishnoi	11.00
3	MI	Jasprit Bumrah	18.00
50	PBKS	Arshdeep Singh	18.00
80	RCB	Josh Hazlewood	12.50
619	RR	Jofra Archer	12.50
56	SRH	Mohammed Shami	10.00



```

In [89]: # Filter the dataset for batsmen only
batsmen_data = ipl[ipl['Type'] == 'AR']

# Drop rows with NaN values in the 'Sold' column to avoid issues with idxmax
batsmen_data = batsmen_data.dropna(subset=['Sold'])

# Find the highest sold batsman in each team
highest_sold_batsman = batsmen_data.loc[batsmen_data.groupby('Team')['Sold'].idxmax()]

# Display the result
print(highest_sold_batsman[['Team', 'Players', 'Sold']])

# Set the figure size explicitly with reasonable values
plt.figure(figsize=(10, 6)) # Use 10x6 inches for the chart

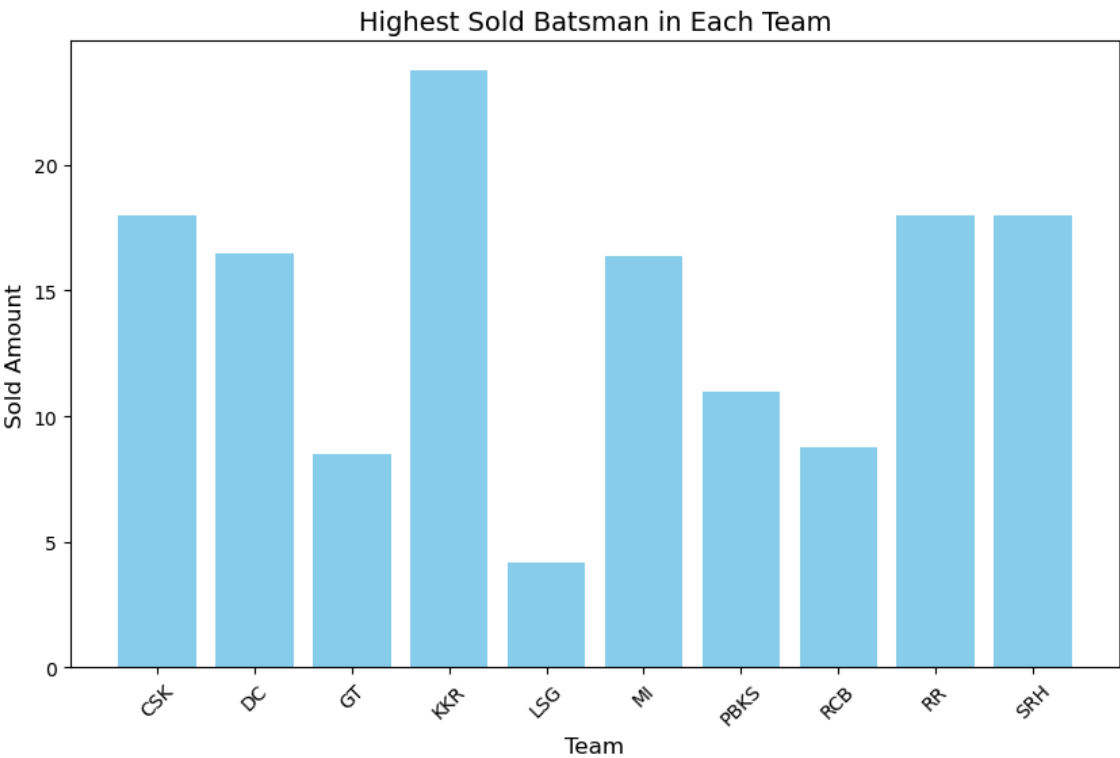
# Create bar chart for highest sold batsmen
bars = plt.bar(highest_sold_batsman['Team'], highest_sold_batsman['Sold'],

# Add chart details
plt.title('Highest Sold Batsman in Each Team', fontsize=14)
plt.xlabel('Team', fontsize=12)
plt.ylabel('Sold Amount', fontsize=12)
plt.xticks(rotation=45)

# Show the plot
plt.show()

```

	Team	Players	Sold
14	CSK	Ravindra Jadeja	18.00
18	DC	Axar Patel	16.50
36	GT	Sai Sudharsan	8.50
66	KKR	Venkatesh Iyer	23.75
102	LSG	Abdul Samad	4.20
5	MI	Hardik Pandya	16.35
71	PBKS	Marcus Stoinis	11.00
53	RCB	Liam Livingstone	8.75
29	RR	Yashasvi Jaiswal	18.00
9	SRH	Pat Cummins	18.00



In [ ]: