

Visualization Library Documentation (with IPL Data)

1. Library Overview

Matplotlib

Matplotlib is a versatile library used for creating static, animated, and interactive plots in Python.

Best For: Custom visualizations, static charts, flexible plot styling.

Seaborn

Seaborn is a high-level interface built on Matplotlib, designed for making statistical graphics.

Best For: Cleaner, insightful visualizations with less code and good default styles.

2. Graph Types Using IPL Sample Data

Dataset Columns Used:

- Player Name (Unnamed: 4)
- Runs (Unnamed: 9)
- Overcount (Unnamed: 10)
- Teams (Unnamed: 3)

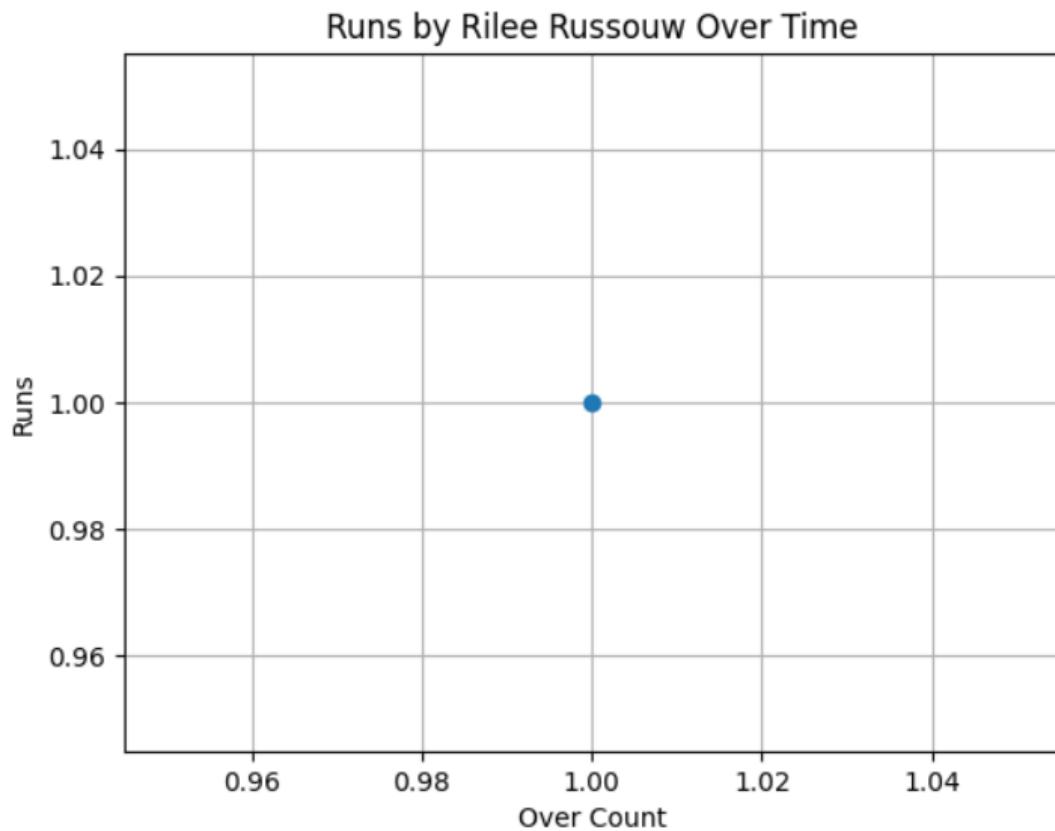
A. Matplotlib

1. Line Plot – Over vs Runs (Top 1 Player's Performance)

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

df = pd.read_excel("IPL sample data.xlsx", sheet_name="Sheet1", header=3)
df.dropna(how='all', inplace=True)
df['Runs'] = pd.to_numeric(df['Unnamed: 9'], errors='coerce')

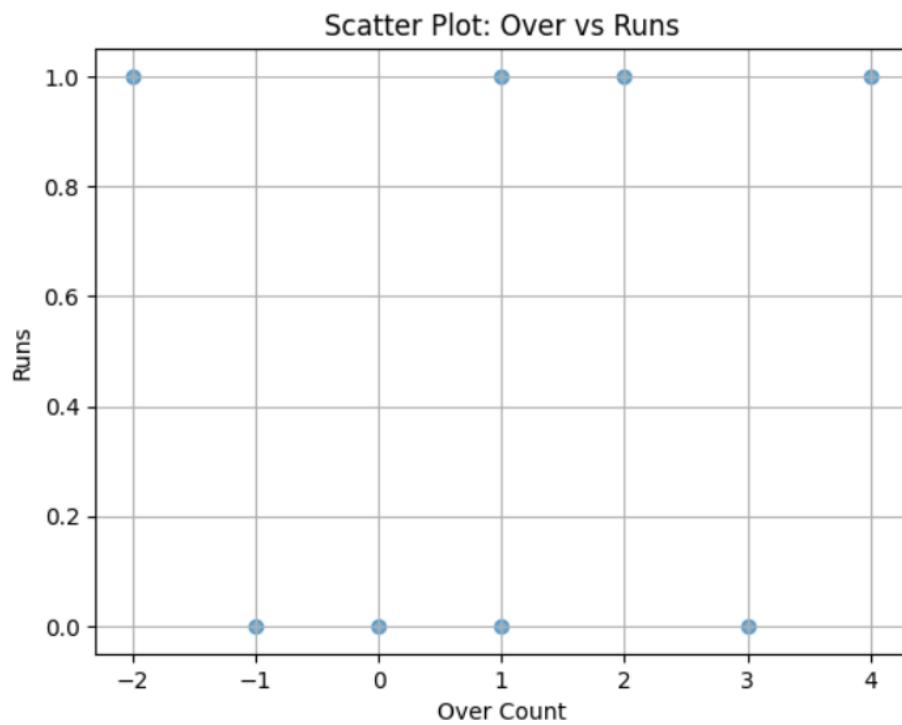
# Filter for one player
player_data = df[df['Unnamed: 4'] == 'Rilee russouw']
plt.plot(player_data['Unnamed: 10'], player_data['Runs'], marker='o')
plt.title("Runs by Rilee Russouw Over Time")
plt.xlabel("Over Count")
plt.ylabel("Runs")
plt.grid(True)
plt.show()
```



2. Scatter Plot – Runs vs Overcount

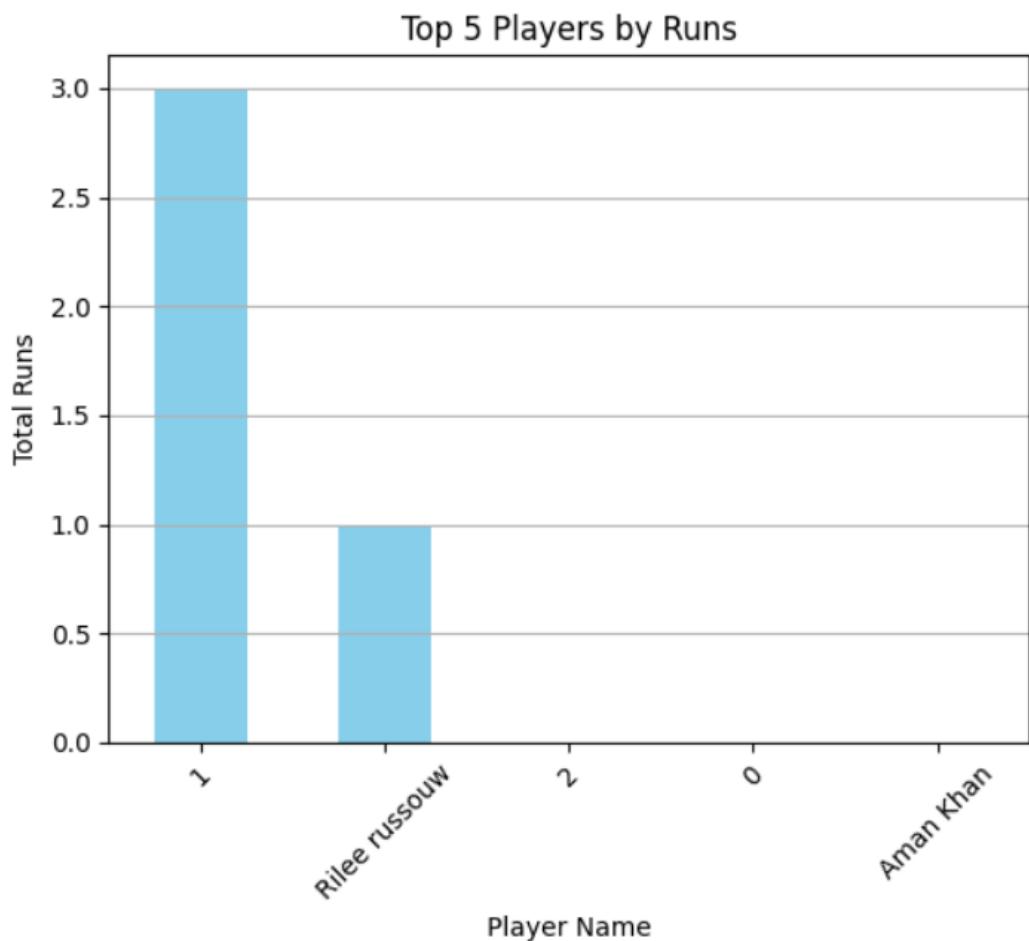
```
plt.scatter(df['Unnamed: 10'], df['Runs'], alpha=0.6)
plt.title("Scatter Plot: Over vs Runs")
plt.xlabel("Over Count")
plt.ylabel("Runs")
plt.grid(True)
plt.show()
```

```
Index(['Unnamed: 0', 'Unnamed: 1', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4',
       'Unnamed: 5', 'Unnamed: 6', 'Unnamed: 7', 'Unnamed: 8', 'Unnamed: 9',
       'Unnamed: 10', 'Unnamed: 11', 'Unnamed: 12', 'Runs'],
      dtype='object')
Unnamed: 10    35
Runs          46
dtype: int64
```



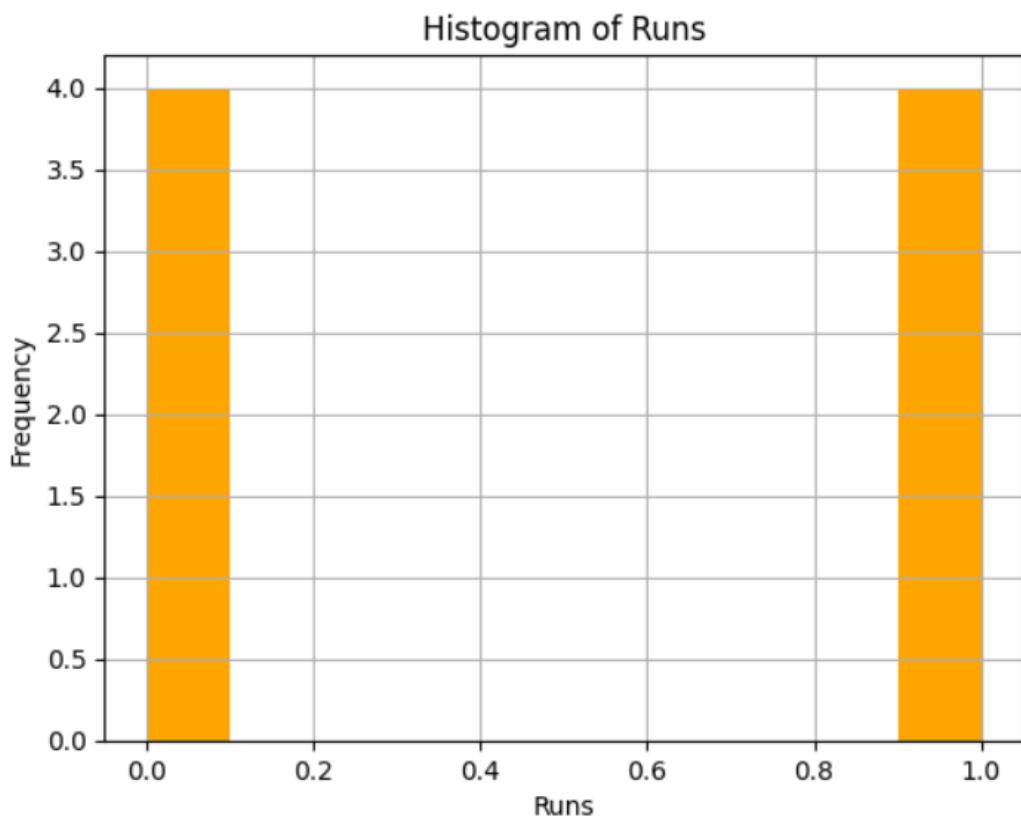
3. Bar Chart – Total Runs by Top 5 Players

```
top_players = df.groupby('Unnamed: 4')[['Runs']].sum().sort_values(ascending=False).head(5)
top_players.plot(kind='bar', color='skyblue')
plt.title("Top 5 Players by Runs")
plt.xlabel("Player Name")
plt.ylabel("Total Runs")
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```



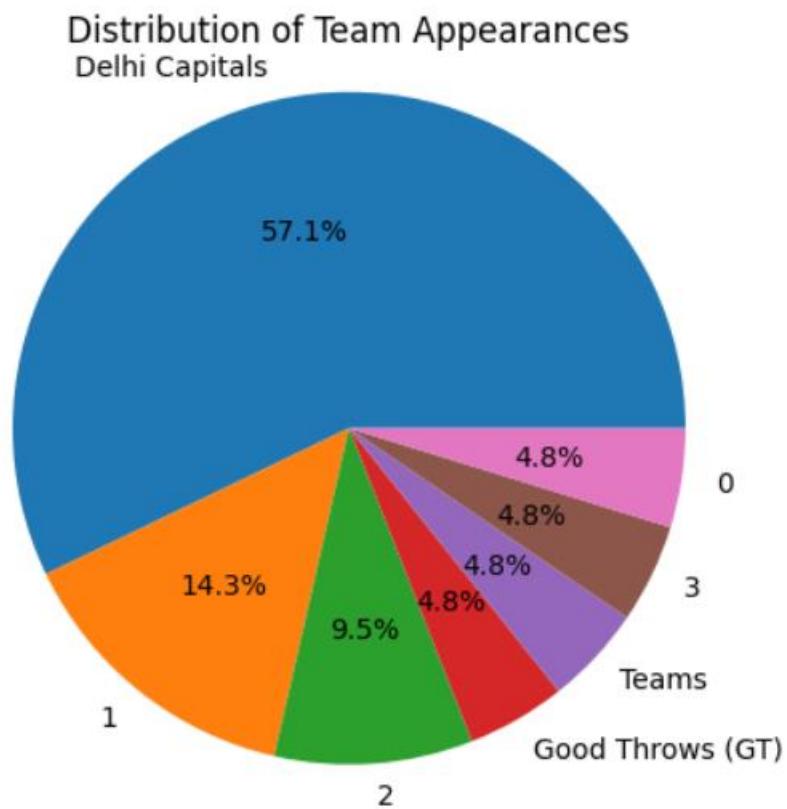
4. Histogram – Distribution of Runs

```
df['Runs'].hist(bins=10, color='orange')
plt.title("Histogram of Runs")
plt.xlabel("Runs")
plt.ylabel("Frequency")
plt.grid(True)
plt.show()
```



5. Pie Chart – Team Participation

```
team_counts = df['Unnamed: 3'].value_counts()  
plt.pie(team_counts, labels=team_counts.index, autopct='%.1f%%')  
plt.title("Distribution of Team Appearances")  
plt.axis('equal')  
plt.show()
```



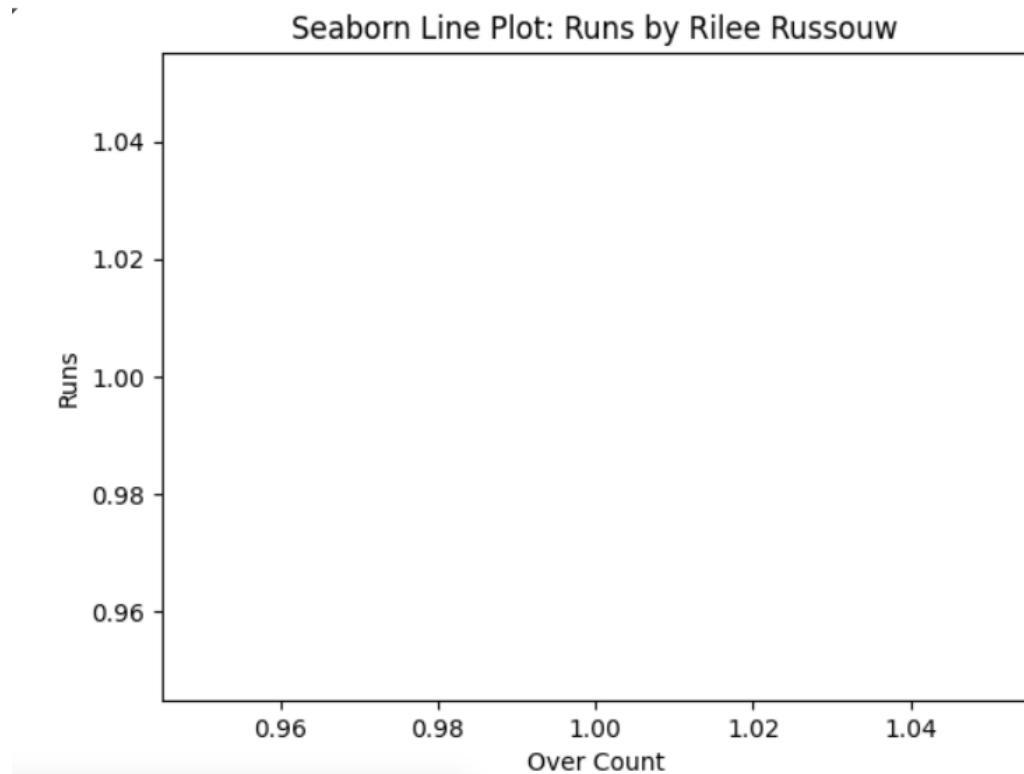
B. Seaborn

```
import seaborn as sns

# Clean again
df['Runs'] = pd.to_numeric(df['Unnamed: 9'], errors='coerce')
df = df.dropna(subset=['Runs'])
```

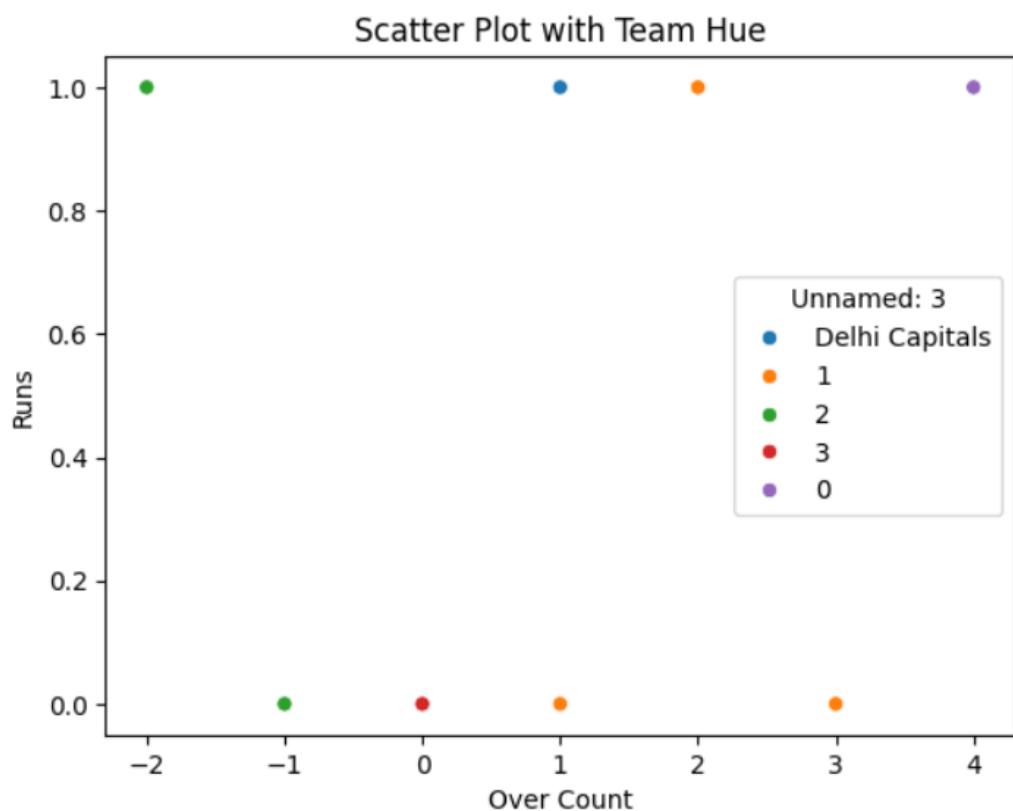
1. Line Plot – Over vs Runs (Same player)

```
sns.lineplot(data=player_data, x='Unnamed: 10', y='Runs')
plt.title("Seaborn Line Plot: Runs by Rilee Russouw")
plt.xlabel("Over Count")
plt.ylabel("Runs")
plt.show()
```



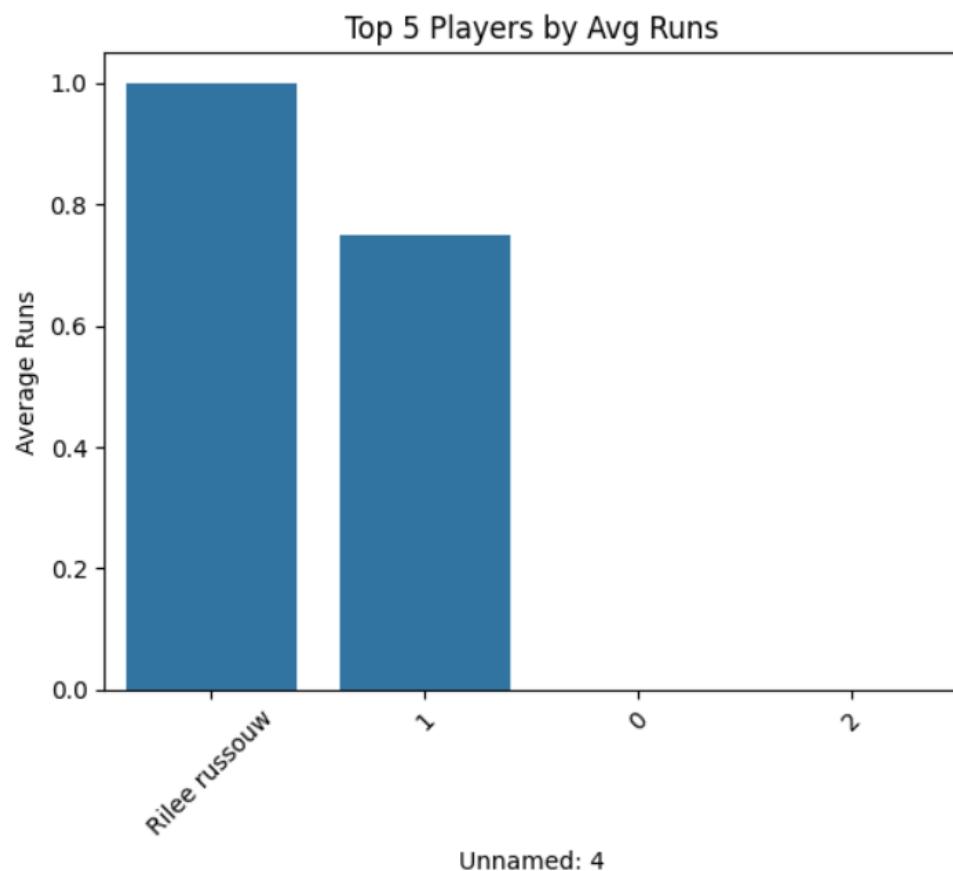
2. Scatter Plot – Runs vs Overcount

```
sns.scatterplot(data=df, x='Unnamed: 10', y='Runs', hue='Unnamed: 3')
plt.title("Scatter Plot with Team Hue")
plt.xlabel("Over Count")
plt.ylabel("Runs")
plt.show()
```



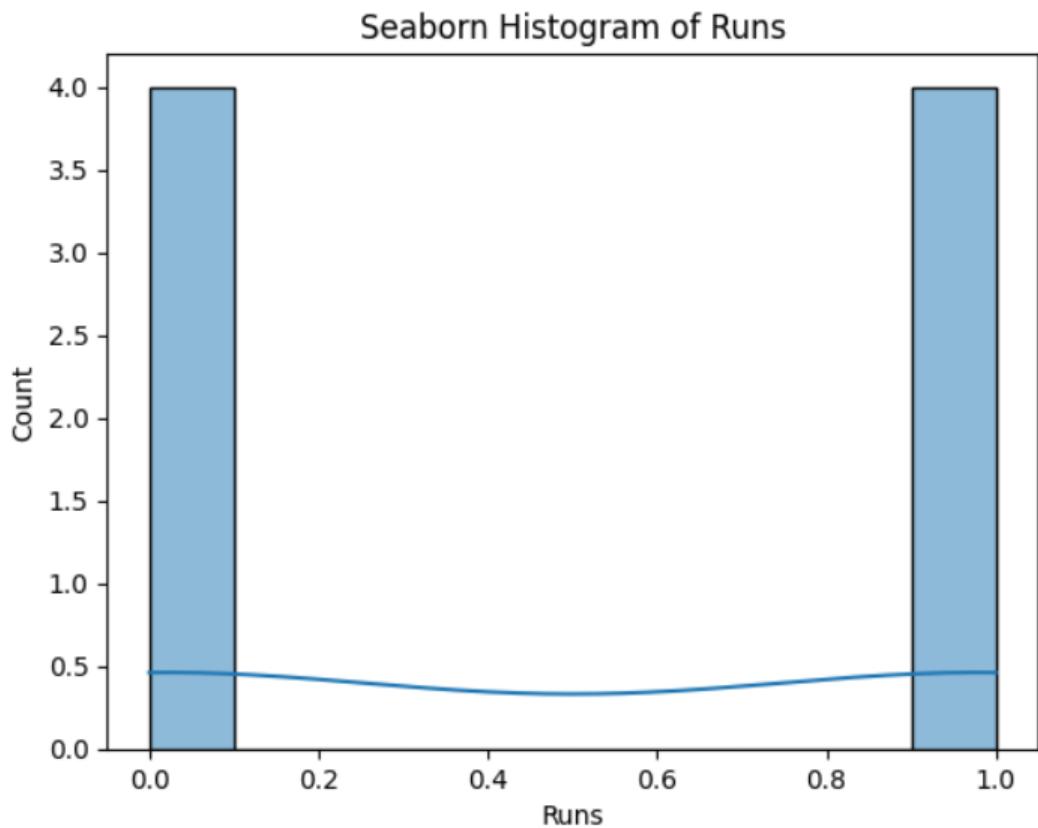
3. Bar Chart – Average Runs per Player (Top 5)

```
avg_runs = df.groupby('Unnamed:  
4')['Runs'].mean().sort_values(ascending=False).head(5)  
sns.barplot(x=avg_runs.index, y=avg_runs.values)  
plt.title("Top 5 Players by Avg Runs")  
plt.ylabel("Average Runs")  
plt.xticks(rotation=45)  
plt.show()
```



4. Histogram – Runs Distribution

```
sns.histplot(df['Runs'], bins=10, kde=True)
plt.title("Seaborn Histogram of Runs")
plt.xlabel("Runs")
plt.show()
```



3. Comparison Table

Feature	Matplotlib	Seaborn
Ease of Use	Medium – requires more setup	Easy – less code, quick results
Customization	Full control (colors, labels, grid, etc.)	Moderate (styled plots, tied to Matplotlib)
Interactivity	Low (static)	Low (static)
Performance	Excellent for all data sizes	Great for medium-sized datasets
Aesthetic Quality	Manual styling needed	Beautiful default themes
Pie Chart Support	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No (use Matplotlib instead)