T20 INTERNATIONAL CRICKET EDA

Batting Statistics:

The batting statistics provided here were collected from “Cricinfo”, one of the leading sources for cricket-related information. This dataset contains comprehensive batting statistics for various cricket players across different formats of the game. The data includes details such as the player's name, career span, total matches played, innings batted, total runs scored, highest score, batting average, and strike rate. These statistics offer insights into each player's batting performance, including their consistency, scoring ability, and contribution to their team's success.

Records includes the following current or recent matches:

- Qatar v Saudi Arabia at Al Amerat, ACC Men's Premier Cup 15th Match, Apr 16, 2024
- Bahrain v Kuwait at Al Amerat, ACC Men's Premier Cup 14th match, Apr 15, 2024
- Oman v United Arab Emirates at Al Amerat, ACC Men's Premier Cup 13th match, Apr 15, 2024

Bowling Statistics:

The bowling statistics provided here were also collected from “Cricinfo”. This dataset comprises detailed bowling statistics for cricket players from various teams and competitions. The data includes information such as the player's name, career span, total matches played, innings bowled, total balls bowled, runs conceded, wickets taken, best bowling figures in an innings (BBI), bowling average, economy rate, bowling strike rate, and number of maiden overs bowled. These statistics offer insights into each player's bowling prowess, effectiveness, and impact on the game.

Records includes the following recent matches:

- Qatar v Saudi Arabia at Al Amerat, ACC Men's Premier Cup 15th Match, Apr 16, 2024
- Bahrain v Kuwait at Al Amerat, ACC Men's Premier Cup 14th match, Apr 15, 2024
- Oman v United Arab Emirates at Al Amerat, ACC Men's Premier Cup 13th match, Apr 15, 2024

Overview:

In this project, I embark on a comprehensive analysis of cricket performance, leveraging batting and bowling statistics to uncover valuable insights. Through meticulous data collection, cleaning, integration, and analysis, I aim to gain a deeper understanding of player performance, team dynamics, and strategic nuances within the game of cricket.

Project Objectives:

1. **Data Collection and Cleaning:**
    - Gather batting and bowling statistics from reputable sources such as Cricinfo, ensuring data reliability and accuracy.
    - Perform thorough data cleaning to address inconsistencies, missing values, and formatting issues, ensuring data integrity for subsequent analysis.
2. **Data Integration and Preparation:**

- Integrate batting and bowling datasets to create a unified dataset, enabling comprehensive analysis of player performance.
- Organize and structure the data for efficient analysis, including player names, career spans, match details, and statistical metrics.

3. **Statistical Analysis and Insights:**
   - Explore key performance indicators such as batting averages, strike rates, bowling averages, economy rates, and wickets taken to derive meaningful insights.
4. **Visualization and Communication:**
   - Visualize analysis results using charts and graphs to enhance data comprehension and communication.
   - Present insights in a clear, concise manner, making them accessible to stakeholders such as coaches, analysts, and cricket enthusiasts.
5. **Interpretation and Contextualization:**
   - Provide actionable recommendations based on insights gained, guiding strategic decision-making for teams and players.

Codes:

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
batting_stats = pd.read_csv(r"C:\Users\Rohan\OneDrive\Desktop\Rohan\T20 Int Matches\Batting_Stats.csv")
batting_stats.head()
```

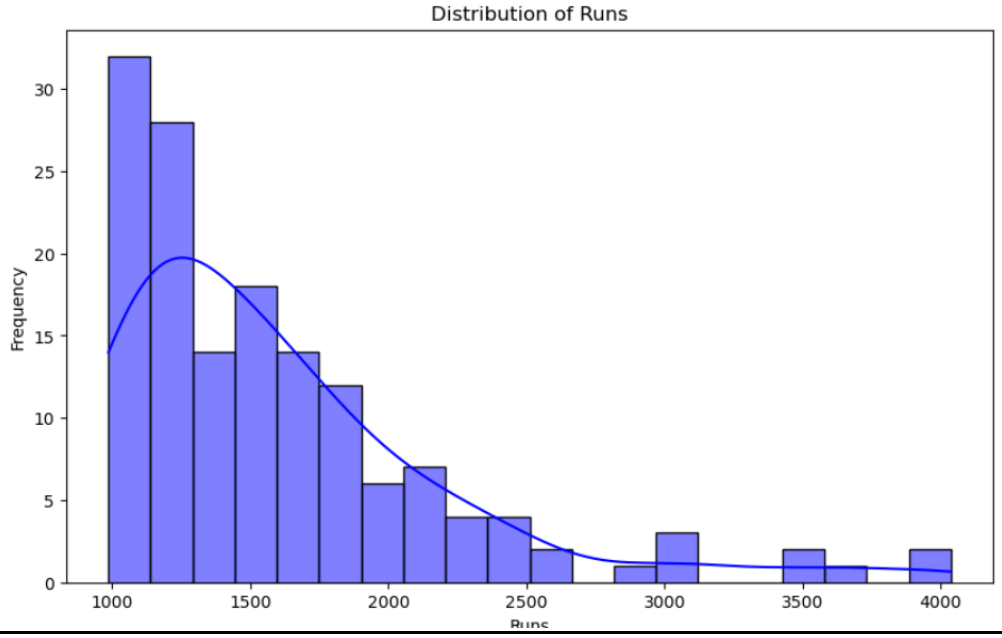
Out[2]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	4s	6s	Runs in Boundary	% Runs in Boundary
0	V Kohli (IND)	2010-2024	117	109	31	4037	122*	51.75	2922	138.15	1	37	5	361	117	2146	53.158286
1	RG Sharma (IND)	2007-2024	151	143	18	3974	121*	31.79	2839	139.97	5	29	12	359	190	2576	64.821339
2	Babar Azam (PAK)	2016-2024	109	103	14	3698	122	41.55	2864	129.12	3	33	5	395	59	1934	52.298540
3	MJ Guptill (NZ)	2009-2022	122	118	7	3531	105	31.81	2602	135.70	2	20	3	309	173	2274	64.401020
4	PR Stirling (IRE)	2009-2024	137	136	11	3491	115*	27.92	2582	135.20	1	23	13	407	124	2372	67.946147

In [6]:
```
print("\nMissing Values in Batting Stats Data:")
print(batting_stats.isnull().sum())
```

```
Missing Values in Batting Stats Data:
Player                0
Span                  0
Mat                   0
Inns                  0
NO                    0
Runs                  0
HS                    0
Ave                   0
BF                    0
SR                    0
100                   0
50                    0
0                     0
4s                    0
6s                    0
Runs in Boundary      0
% Runs in Boundary    0
dtype: int64
```

In [7]:
```
# Distribution of Runs
plt.figure(figsize=(10, 6))
sns.histplot(batting_stats['Runs'], bins=20, kde=True, color='blue')
plt.title('Distribution of Runs')
plt.xlabel('Runs')
plt.ylabel('Frequency')
plt.show()
```



TOP 5 PLAYERS WITH MOST STRIKE RATE IN T20I MATCH

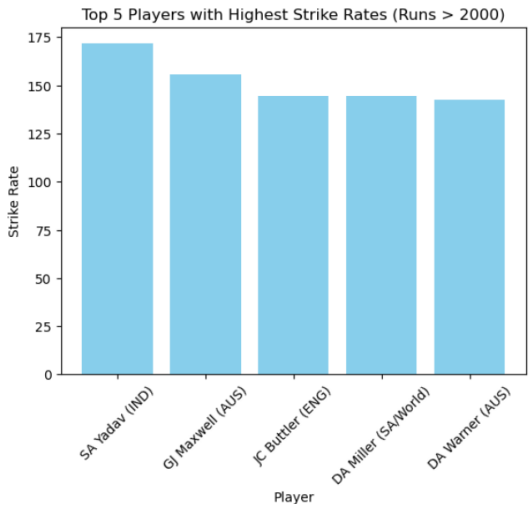
In [17]:
```
# Sort the DataFrame by 'SR' (Strike Rate) in descending order
sorted_df = batting_stats[batting_stats['Runs']>2000].sort_values(by='SR', ascending=False)

# Select the top 5 players
top_5_players = sorted_df.head(5)

# Display the top 5 players with highest strike rates
top_5_players[['Player', 'Runs', 'SR']]
```

Out[17]:

	Player	Runs	SR
21	SA Yadav (IND)	2141	171.55
11	GJ Maxwell (AUS)	2468	155.51
8	JC Buttler (ENG)	2927	144.61
17	DA Miller (SA/World)	2268	144.55
6	DA Warner (AUS)	3099	142.67

TOP 5 PLAYERS WITH MORE THAN 2000 RUNS AND HIGHEST SR

In [51]:
```
# Plot a bar chart for the top 5 players
plt.bar(top_5_players['Player'], top_5_players['SR'], color='skyblue')
plt.xlabel('Player')
plt.ylabel('Strike Rate')
plt.title('Top 5 Players with Highest Strike Rates (Runs > 2000)')
plt.xticks(rotation=45)
plt.show()
```



TOP 5 PLAYERS WITH MOST RUNS IN BOUNDARY (PERCENTAGE)

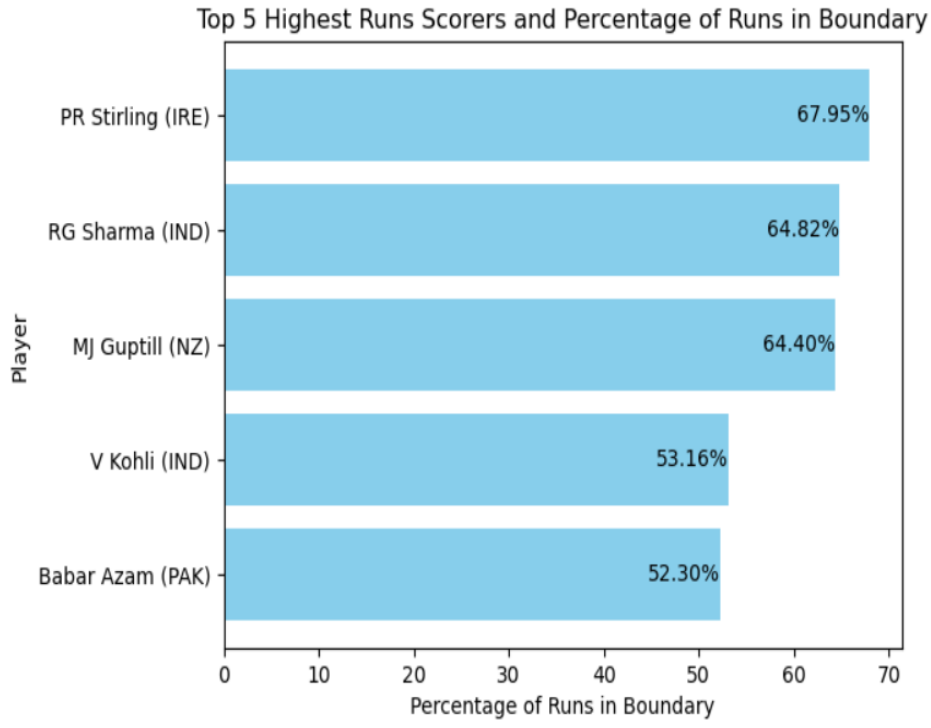
In [34]:
```
# Sorting the DataFrame by 'Runs' in descending order to get the top 5 highest runs scorers
top_5_scorers = batting_stats.sort_values(by='Runs', ascending=False).head(5)

# Calculating the percentage of runs scored in boundary for each player
top_5_scorers['Percentage Runs in Boundary'] = (top_5_scorers['Runs in Boundary'] / top_5_scorers['Runs']) * 100

# Displaying the top 5 highest runs scorers and their corresponding percentage of runs in boundary
top_5_scorers[['Player', 'Runs', 'Percentage Runs in Boundary']]
```

Out[34]:

	Player	Runs	Percentage Runs in Boundary
0	V Kohli (IND)	4037	53.158286
1	RG Sharma (IND)	3974	64.821339
2	Babar Azam (PAK)	3698	52.298540
3	MJ Guptill (NZ)	3531	64.401020
4	PR Stirling (IRE)	3491	67.946147

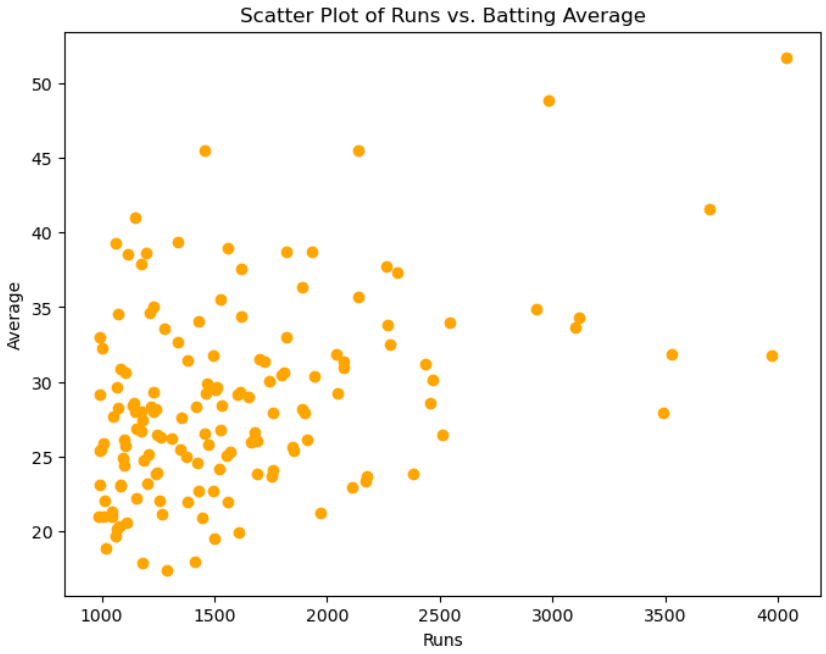
TOP 5 HIGHEST RUN SCORERS IN T20I MATCHES BASED ON THE RUNS SCORED IN BOUNDARIES

In [79]:
```
# Sorting the players and their corresponding percentage of runs in boundary in descending order
sorted_players, sorted_runs_in_boundary = zip(*sorted(zip(players, runs_in_boundary), key=lambda x: x[1], reverse=False))

# Plot a horizontal bar chart for the sorted data
bars = plt.barh(sorted_players, sorted_runs_in_boundary, color='skyblue')
plt.xlabel('Percentage of Runs in Boundary')
plt.ylabel('Player')
plt.title('Top 5 Highest Runs Scorers and Percentage of Runs in Boundary')

# Add exact percentage values on top of each bar
for bar, percentage in zip(bars, sorted_runs_in_boundary):
    plt.text(bar.get_width(), bar.get_y() + bar.get_height()/2, f'{percentage:.2f}%',
             va='center', ha='right', fontsize=10, color='black')
plt.show()
```



SCATTER PLOT B/W RUNS AND BATTING AVERAGE

In [71]:
```
# Scatter Plot for Runs vs. Average
plt.figure(figsize=(8, 6))
plt.scatter(batting_stats['Runs'], batting_stats['Ave'], color='orange')
plt.xlabel('Runs')
plt.ylabel('Average')
plt.title('Scatter Plot of Runs vs. Batting Average')
plt.show()
```



TOP 5 PLAYERS OF EACH COUNTRY WITH HIGHEST RUNS

In [75]:
```
# Extract country from the 'Player' column
batting_stats['Country'] = batting_stats['Player'].str.extract(r'\((.*?)\)')

# Grouping the data by country and sorting the runs within each group
top_batsmen_by_country = batting_stats.groupby('Country').apply(lambda x: x.nlargest(5, 'Runs')).reset_index(drop=True)
top_batsmen_by_country[['Player', 'Runs', 'Country']]
```

Out[75]:

	Player	Runs	Country
0	Mohammad Nabi (AFG)	2109	AFG
1	Mohammad Shahzad (AFG)	2048	AFG
2	Najibullah Zadran (AFG)	1808	AFG
3	Asghar Afghan (AFG)	1382	AFG
4	Rahmanullah Gurbaz (AFG)	1376	AFG
...	...	...	...
111	Sikandar Raza (ZIM)	1854	ZIM
112	SC Williams (ZIM)	1691	ZIM
113	H Masakadza (ZIM)	1662	ZIM
114	CR Ervine (ZIM)	1429	ZIM
115	RP Burl (ZIM)	1156	ZIM

116 rows × 3 columns

TOP 5 INDIAN BATSMEN WITH MOST RUNS IN T20I MATCHES

In [76]:
```
# Filtering the DataFrame for rows where the country is "IND" (India)
indian_batsmen = batting_stats[batting_stats['Country'] == 'IND']

# Sorting the runs in descending order to find the top 5 batsmen for India
top_5_indian_batsmen = indian_batsmen.nlargest(5, 'Runs')
top_5_indian_batsmen[['Player', 'Runs', 'Country']]
```

Out[76]:

	Player	Runs	Country
0	V Kohli (IND)	4037	IND
1	RG Sharma (IND)	3974	IND
18	KL Rahul (IND)	2265	IND
21	SA Yadav (IND)	2141	IND
41	S Dhawan (IND)	1759	IND

MOST CENTURIES SCORED BY A COUNTRY IN T20 INTERNATION MATCHES

In [86]:
```
# Converting the '100' column to numeric type
batting_stats['100'] = pd.to_numeric(batting_stats['100'], errors='coerce')

# Grouping the dataset by 'Country' and sum the centuries scored by each player
centuries_by_country = batting_stats.groupby('Country')['100'].sum()

# Find the country with the maximum centuries scored
most_centuries_country = centuries_by_country.idxmax()
most_centuries = centuries_by_country.max()

# Display the country with the most centuries and the number of centuries
print(f"Country with the most centuries: {most_centuries_country}")
print(f"Number of centuries: {most_centuries}")
```

```
Country with the most centuries: IND
Number of centuries: 13.0
```

BREAKDOWN OF MOST CENTURIES IN T20I MATCHES BY INDIAN PLAYERS

In [95]:
```
# Filter the dataset for players from India
indian_players = batting_stats[batting_stats['Country'] == 'IND']

# Group the dataset by player and count the number of centuries scored by each player
centuries_by_player = indian_players.groupby('Player')['100'].sum().reset_index()
centuries_by_player.sort_values(by='100',ascending=False)
```

Out[95]:

	Player	100
3	RG Sharma (IND)	5.0
5	SA Yadav (IND)	4.0
1	KL Rahul (IND)	2.0
6	SK Raina (IND)	1.0
8	V Kohli (IND)	1.0
0	HH Pandya (IND)	0.0
2	MS Dhoni (IND)	0.0
4	S Dhawan (IND)	0.0
7	SS Iyer (IND)	0.0
9	Yuvraj Singh (IND)	0.0

MOST NUMBER OF 50s BY BATTERS ACROSS T20I FORMAT

In [120]:
```
# Group the dataset by country and find the player with the maximum number of 50s in each group
batsmen_with_most_50s = batting_stats.loc[batting_stats.groupby('Country')['50'].idxmax()]
batsmen_with_most_50s[['Player', 'Country', '50']].sort_values(by='50',ascending=False).head(10)
```

Out[120]:

	Player	Country	50
0	V Kohli (IND)	IND	37
2	Babar Azam (PAK)	PAK	33
6	DA Warner (AUS)	AUS	26
4	PR Stirling (IRE)	IRE	23
8	JC Buttler (ENG)	ENG	22
3	MJ Guptill (NZ)	NZ	20
25	S Ssesazi (UGA)	UGA	16
15	Virandeep Singh (MAS)	MAS	16
32	CH Gayle (WI)	WI	14
37	Muhammad Waseem (UAE)	UAE	14

DISPLAYING CAREER SPAN OF THE BATSMEN

In [131]:
```
def display_career_stats(player_name):
    # Check if the player exists in the dataset
    if player_name in batting_stats['Player'].values:
        # Retrieve player's data
        player_data = batting_stats[batting_stats['Player'] == player_name].iloc[0]

        # Print career statistics
        print(f"Player: {player_data['Player']}")
        print(f"Career Span: {player_data['Span']}")
        print(f"Matches Played: {player_data['Mat']}")
        print(f"Total Runs: {player_data['Runs']}")
        print(f"Highest Score: {player_data['HS']}")
        print(f"Strike Rate: {player_data['SR']}")
        print(f"Batting Average: {player_data['Ave']}")
    else:
        print("Player not found in the dataset")

# Take input from the user for player's name
player_name = input("Enter the player's name: ")

# Display career statistics for the selected palayer
display_career_stats(player_name)
```

```
Enter the player's name: RG Sharma (IND)
Player: RG Sharma (IND)
Career Span: 2007-2024
Matches Played: 151
Total Runs: 3974
Highest Score: 121*
Strike Rate: 139.97
Batting Average: 31.79
```

Bowling Stats:

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
bowling_stats = pd.read_csv(r"C:\Users\Rohan\OneDrive\Desktop\Rohan\T20 Int Matches\Bowling_Stats.csv")
bowling_stats.head()
```

Out[2]:

	Player	Span	Mat	Inns	Balls	Overs	Mdns	Runs	Wkts	BBI	Ave	Econ	SR	4	5
0	TG Southee (NZ)	2008-2024	123	120	2681	446.5	6	3635	157	43221	23.15	8.13	17.07	2	2
1	Shakib Al Hasan (BAN)	2006-2023	117	115	2535	422.3	3	2869	140	43952	20.49	6.79	18.10	5	2
2	Rashid Khan (AFG/ICC)	2015-2024	85	85	1946	324.2	1	1970	138	45356	14.27	6.07	14.10	5	2
3	IS Sodhi (NZ)	2014-2024	111	107	2279	379.5	-	3048	132	46844	23.09	8.02	17.26	3	-
4	MJ Santner (NZ)	2015-2024	100	98	2072	345.2	2	2457	111	45600	22.13	7.11	18.66	3	-

In [5]:
```
print("\nMissing Values in Bowling Stats Data:")
print(bowling_stats.isnull().sum())
```

```
Missing Values in Bowling Stats Data:
Player    0
Span      0
Mat       0
Inns      0
Balls     0
Overs     0
Mdns      0
Runs      0
Wkts      0
BBI       0
Ave       0
Econ      0
SR        0
4         0
5         0
dtype: int64
```

TOP 5 BOWLERS WITH BEST ECONOMY IN T20I MATCHES WITH MORE THAN 85 GAMES PLAYED

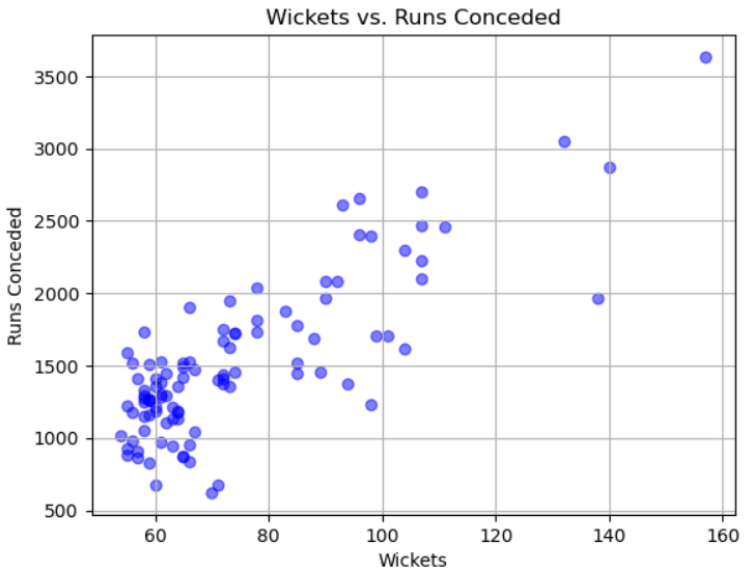
In [13]:
```
filtered_stats = bowling_stats[bowling_stats['Mat'] > 85]
sorted_stats = filtered_stats.sort_values(by='Econ', ascending=True)
top_5_bowlers = sorted_stats.head(5)
top_5_bowlers[['Player', 'Econ']]
```

Out[13]:

	Player	Econ
69	Mohammad Hafeez (PAK)	6.60
14	Shahid Afridi (ICC/PAK)	6.63
1	Shakib Al Hasan (BAN)	6.79
21	B Kumar (IND)	6.96
10	Shadab Khan (PAK)	7.07

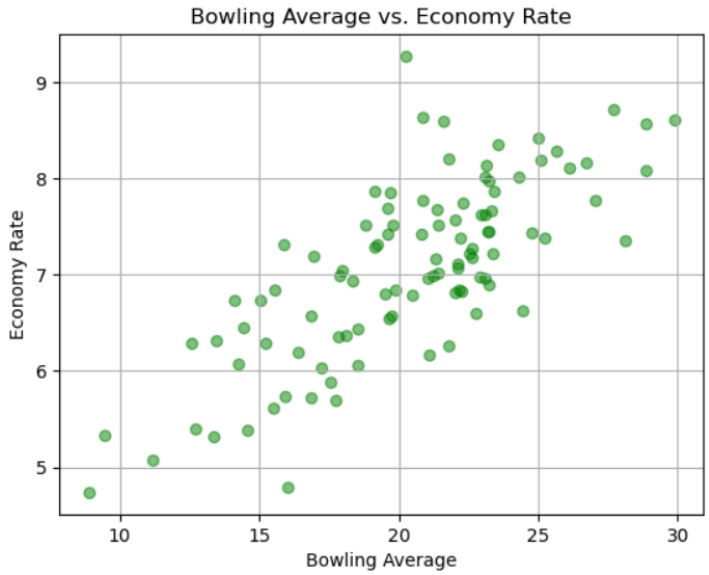
SCATTER PLOT B/W WICKETS AND RUNS CONCEDED

In [19]:
```
plt.scatter(bowling_stats['Wkts'], bowling_stats['Runs'], color='blue', alpha=0.5)
plt.title('Wickets vs. Runs Conceded')
plt.xlabel('Wickets')
plt.ylabel('Runs Conceded')
plt.grid(True)
plt.show()
```



SCATTER PLOT B/W BOWLING AVERAGE AND ECONOMY RATE

In [20]:
```
plt.scatter(bowling_stats['Ave'], bowling_stats['Econ'], color='green', alpha=0.5)
plt.title('Bowling Average vs. Economy Rate')
plt.xlabel('Bowling Average')
plt.ylabel('Economy Rate')
plt.grid(True)
plt.show()
```



TOP 5 BOWLERS OF EACH COUNTRY WITH HIGHEST WICKETS

In [24]:
```
# Extract country from the 'Player' column
bowling_stats['Country'] = bowling_stats['Player'].str.extract(r'\((.*?)\)')

# Grouping the data by country and sorting the runs within each group
top_bowlers_by_country = bowling_stats.groupby('Country').apply(lambda x: x.nlargest(5, 'Wkts')).reset_index(drop=True)
top_bowlers_by_country[['Player', 'Wkts', 'Country']]
```

Out[24]:

	Player	Wkts	Country
0	Mohammad Nabi (AFG)	93	AFG
1	Mujeeb Ur Rahman (AFG)	58	AFG
2	Rashid Khan (AFG/ICC)	138	AFG/ICC
3	A Zampa (AUS)	92	AUS
4	MA Starc (AUS)	74	AUS
...	...	...	...
87	S Badree (WI/World)	56	WI/World
88	TL Chatara (ZIM)	62	ZIM
89	R Ngarava (ZIM)	59	ZIM
90	LM Jongwe (ZIM)	58	ZIM
91	Sikandar Raza (ZIM)	58	ZIM

92 rows × 3 columns

TOP 5 INDIAN BOWLERS WITH MOST NUMBER OF WICKETS IN T20I MATCHES

In [27]:
```
# Filtering the DataFrame for rows where the country is "IND" (India)
indian_bowler = bowling_stats[bowling_stats['Country'] == 'IND']

# Sorting the wickets in descending order to find the top 5 bowlers for India
top_5_indian_bowlers = indian_bowler.nlargest(5, 'Runs')
top_5_indian_bowlers[['Player','Mat','Wkts', 'Country']]
```

Out[27]:

	Player	Mat	Wkts	Country
15	YS Chahal (IND)	80	96	IND
21	B Kumar (IND)	87	90	IND
36	HH Pandya (IND)	92	73	IND
40	R Ashwin (IND)	65	72	IND
31	JJ Bumrah (IND)	62	74	IND

TOP 5 COUNTRIES WITH MOST NUMBER OF WICKETS IN T20I

In [34]:
```
wickets_by_country = bowling_stats.groupby('Country')['Wkts'].sum().reset_index()
sorted_wickets = wickets_by_country.sort_values(by='Wkts', ascending=False)

top_5_countries = sorted_wickets.head(5)
for index, row in top_5_countries.iterrows():
    country = row['Country']
    total_wickets = row['Wkts']
top_5_countries[['Country','Wkts']]
```

Out[34]:

	Country	Wkts
24	PAK	682
22	NZ	647
11	IND	526
30	SL	398
12	IRE	391

DISPLAYING CAREER SPAN OF THE BOWLER

In [42]:
```python
def display_bowling_career_stats(player_name):
    # Check if the player exists in the dataset
    if player_name in bowling_stats['Player'].values:
        # Retrieve player's data
        player_data = bowling_stats[bowling_stats['Player'] == player_name].iloc[0]

        # Print career statistics
        print(f"Player: {player_data['Player']}")
        print(f"Career Span: {player_data['Span']}")
        print(f"Matches Played: {player_data['Mat']}")
        print(f"Total Wickets: {player_data['Wkts']}")
        print(f"Bowling Average: {player_data['Ave']}")
        print(f"Economy: {player_data['Econ']}")
        print(f"Maidens Bowled: {player_data['Mdns']}")
    else:
        print("Player not found in the dataset")

# Take input from the user for player's name
player_name = input("Enter the player's name: ")

# Display career statistics for the selected player
display_bowling_career_stats(player_name)
```

```
Enter the player's name: YS Chahal (IND)
Player: YS Chahal (IND)
Career Span: 2016-2023
Matches Played: 80
Total Wickets: 96
Bowling Average: 25.09
Economy: 8.19
Maidens Bowled: 2
```

Conclusion:

In wrapping up our analysis of batting statistics, we've uncovered a wealth of insights into player performance and team dynamics in cricket. By delving into key metrics like runs scored, batting average, highest score, and strike rate, we've gained a nuanced understanding of each player's batting prowess and their impact on matches.

Through rigorous statistical analysis, we've identified trends, patterns, and correlations in batting performance, shedding light on the factors that contribute to player success. Visualizations such as charts and graphs have aided in interpreting the data, facilitating clear communication of our findings.

As we conclude our exploration of bowling statistics, we've unravelled compelling insights into player performance and tactical nuances in cricket. By scrutinizing key metrics such as wickets taken, bowling average, economy rate, and bowling strike rate, we've gained a comprehensive understanding of each player's bowling prowess and their influence on match dynamics.

Our in-depth statistical analysis has unearthed trends, patterns, and correlations in bowling performance, illuminating the factors that underpin player success. Visual representations of the data have facilitated interpretation, enabling us to distil complex information into actionable insights.

Bowling statistics are instrumental in gauging player effectiveness, wicket-taking ability, and control over opposition batsmen. They offer invaluable insights into bowling partnerships, strategic variations, and game-changing spells that dictate match outcomes.