VISUALIZATION CODE SUMMARY

Here's a summarized explanation of the key approaches and steps in the code:

Data Loading and Initial Inspection:

- The code starts by importing necessary libraries and suppressing warnings.
- It loads a dataset named 'data_indvspak.csv' using Pandas and displays the first few rows using df.head() to get an initial look at the data.
- The shape of the dataset is checked using df.shape to determine the number of rows and columns.

Data Cleaning:

- The code checks for missing values using df.isnull().sum() and prints information about the dataset using df.info() to understand the data types and missing values.
- The 'match_date' column is converted to datetime format using pd.to_datetime.
- The code checks the value counts of 'player_name' to understand the distribution of players.

Filtering Batting Data:

- Rows with batting statistics are filtered based on specific conditions to exclude rows with 'DNB' (Did Not Bat), 'TDNB' (Team Did Not Bat), and 'absent' values in the 'runs_scored' column.
- The asterisks ('*') are removed from the 'runs_scored' column to convert the values to numeric.
- The 'runs_scored' column is converted to an integer data type.

Calculating Batting Statistics:

- Batting statistics for each player are calculated, specifically the average runs scored in the last 7 matches.
- The code groups the data by 'player_id' and 'player_name' and calculates these statistics.

Merging Batting Stats with the Main Dataset:

- The calculated batting statistics are merged with the main dataset 'df' using a left join based on the 'player_name' column.

Defining a Function for Bowling Stats:

- A function called calculate_bowling_stats is defined to handle the calculation of bowling statistics.

- This function filters out rows where the specified column (e.g., 'wickets', 'catches') is not equal to '-' and performs data type conversions.
- It calculates the average value for the specified statistic for each player based on the last 5 matches.

Calculating Bowling Statistics (Wickets, Catches, Stumpings, Runs Conceded):

- Bowling statistics (wickets, catches, stumpings, runs conceded) are calculated using the calculate_bowling_stats function for each player.
- Missing values ('-') are replaced with the calculated averages.

Feature Engineering and Data Transformation:

- Additional features are engineered, such as 'DNB' (Did Not Bat) and 'TDNB' (Team Did Not Bat) columns, which are converted to integers.
- The 'year' column is extracted from the 'match_date' column.
- Player experience (years_of_experience) is calculated as the difference between the maximum and minimum years for each player.

Bowling Average Calculation:

- Bowling averages are calculated for each player and added as a new column named 'bowling_average'.
- The code handles cases where 'wickets' might be missing or '-1'.

Data Visualization:

- Seaborn and Matplotlib are used to create various data visualizations to explore relationships between variables, including bar plots, scatter plots, count plots, and more.

Mean Statistics and Correlation Analysis:

- Mean runs and wickets per year and per opposition are calculated and merged with the main dataset.
- Correlation matrices and heatmaps are generated to visualize correlations between variables.

Exporting Data:

- The cleaned and processed data is exported to a CSV file named 'cricket2.csv' without including the index column.

In summary, the code starts with data loading, cleaning, and feature engineering, followed by the calculation of batting and bowling statistics. It also includes data visualization and correlation analysis to gain insights from the dataset. The final preprocessed dataset is exported for further analysis or modeling.