Projects Summary:

1. Command Line Calculator

Project Overview

- **Content**:
    - Simple calculator that supports addition, subtraction, multiplication, division, square root, and exponentiation.
    - User selects the operation and enters the numbers.

Code Explanation

- **Content**:
    - Functions for each operation (Add, Subtract, Multiply, Divide, Square Root, Power).
    - Example: add(x, y) adds two numbers, divide(x, y) handles division with error checking for zero.

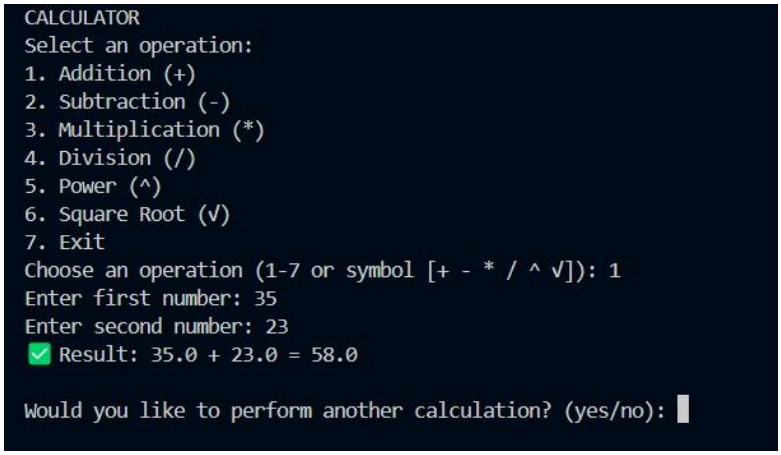
How It Works

- **Content**:
    - User selects the operation.
    - Enters numbers based on the operation selected.
    - The program performs the operation and displays the result.

Tools and Technologies Used:

- Python

Result Screenshots:



2. CSV File Analysis

Project Overview

This project loads a CSV file, analyzes the data, and provides basic statistics (mean, median, max, min). Users can filter data based on their input for customized analysis.

Code Explanation

- **Data Loading**: CSV file is read using pandas.
- **Statistics**: Calculates mean, median, max, and min for selected columns.
- **Filtering**: Allows data filtering based on user input.
- **Visualization**: Generates basic plots using matplotlib.

How It Works

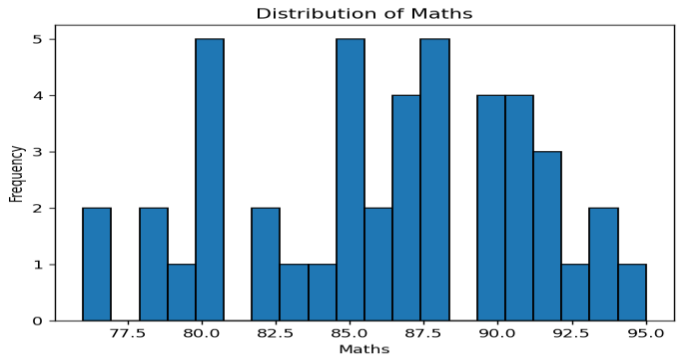
1. Load CSV into a DataFrame.
2. User selects a column or filter condition.
3. Program calculates and displays statistics.
4. Optional visualizations are generated.

Result Screenshots:

```
PS C:\Users\ashwi\OneDrive\Desktop\Data Softis\SecondProject> python ./analyze.py
Enter the path to the CSV file: C:\Users\ashwi\OneDrive\Desktop\Data Softis\SecondProject\marks_data.csv

Available columns: ['StudentID', 'Name', 'Maths', 'English', 'Science', 'History', 'Total Marks']
Enter the column name to analyze: Maths
Available columns: ['StudentID', 'Name', 'Maths', 'English', 'Science', 'History', 'Total Marks']
Available columns: ['StudentID', 'Name', 'Maths', 'English', 'Science', 'History', 'Total Marks']
Enter the column name to analyze: Maths
Statistics for column 'Maths':
Average: 86.2
Maximum: 95
Minimum: 76

Would you like to visualize the data? (yes/no):
```



3. Web Scraping

Project Overview

This project scrapes data from a website using requests and BeautifulSoup, extracting relevant information like article titles, prices, or descriptions. The data is then saved in a structured format (CSV or JSON).

Code Explanation

- **Requests**: Used to fetch the webpage content.
- **BeautifulSoup**: Extracts specific data from the HTML.
- **Error Handling**: Manages missing elements and failed requests.

How It Works

1. Fetch webpage content with requests.
2. Parse the HTML to extract data using BeautifulSoup.
3. Save the scraped data in CSV or JSON format.

Result Screenshots: