Cybersecurity Threats Data Analytics Project

This project focuses on analyzing a global cybersecurity threats dataset (2015-2024). The goal is to extract actionable insights and detect patterns that can help in understanding the landscape of cybersecurity attacks.

Key Data Analytics Tasks:

1. **Cleaning and Handling Missing Values:**
    - Checked for null values and handled them using median/mode imputation.
    - Ensured no missing values that could bias the results.
2. **Feature Selection and Engineering:**
    - Selected relevant features such as country, attack type, and financial loss.
    - Engineered a new feature: Loss per User ($), providing granular financial impact analysis.
3. **Ensuring Data Integrity and Consistency:**
    - Removed duplicate records.
    - Standardized categorical values for consistency in analysis.
4. **Summary Statistics and Insights:**
    - Generated summary statistics for numerical and categorical data.
    - Identified the most frequent attack types and countries with high average financial loss.
5. **Identifying Patterns, Trends, and Anomalies:**
    - Analyzed trends in financial losses and affected users over time.
    - Highlighted anomalies with unusually high financial losses per user.
6. **Handling Outliers and Data Transformations:**
    - Detected and removed outliers using the IQR method.
    - Applied log transformation on skewed columns to normalize data.
7. **Initial Visual Representation of Key Findings:**
    - Plotted the top attack types.
    - Created a correlation heatmap.
    - Visualized yearly financial losses through a time series line plot

Technology Used

1. **Python**
    - Primary programming language for data analysis and visualization.
    - Offers a vast ecosystem of libraries for data cleaning, transformation, and insights.
2. **Pandas**
    - Used for data manipulation and analysis.
    - Helps in handling missing values, filtering data, and generating summary statistics.
3. **NumPy**
    - Supports numerical operations and data transformations.
    - Essential for mathematical computations and handling arrays.
4. **Matplotlib & Seaborn**
    - Libraries used for data visualization.
    - Enabled generation of charts like bar plots, line plots, and heatmaps to visually interpret trends and correlations.

5. **FPDF**
    - A Python library for generating PDF reports.
    - Used to create a structured project description in a shareable format.
6. **Jupyter Notebook / Python Script**
    - Development environment for running and iterating over code interactively.
    - Facilitates a notebook-based approach to testing, exploring, and documenting the analysis process.
7. **CSV (Comma-Separated Values) Format**
    - The dataset was provided in CSV format, which is commonly used for storing tabular data.