Name - Srushti Satish Sambare **ADS - 3** **D17A - 54**

Aim - Explore data visualization techniques.

Theory -

Data Visualization
Data visualization is the process of representing numerical and categorical data in a visual format such as charts, graphs, and maps. It helps in identifying patterns, trends, correlations, and outliers in datasets.

In Google Colab, we use libraries like:

- **Matplotlib** – for basic plotting
- **Seaborn** – for advanced statistical visualizations
- **Pandas** – for handling and preprocessing data

Data Visualization Techniques Used

1. **Histograms – Distribution of Numerical Data**
   A histogram is used to show the frequency distribution of numerical values. It helps in understanding how values are spread across the dataset.
   - **Purpose:** Identify skewness, central tendency, and spread.
   - **Library Used:** *matplotlib.pyplot*
   - **Function Used:** df.hist(bins=30)

2. **Box Plot – Outlier Detection**
   A box plot (also called a whisker plot) shows the distribution of data based on quartiles and highlights outliers.
   - **Components of a Box Plot:**
     - Q1 (25th percentile) and Q3 (75th percentile)
     - Median (Q2)
     - Whiskers (minimum & maximum values)
     - Outliers (points beyond whiskers)
   - **Purpose:** Detects outliers in numerical columns.
   - **Library Used:** *seaborn*
   - **Function Used:** *sns.boxplot(data=df.iloc[:, 2:], orient="h")*

3. **Scatter Plot – Relationship Between Two Variables**
   A scatter plot is used to visualize relationships between two numerical features. It helps in detecting correlations and clusters.
   - **Purpose:** Understand patterns and trends.
   - **Library Used:** *seaborn*
   - **Function Used:** *sns.scatterplot(x=df["Difference_x"], y=df["Difference_y"], hue=df["Type"])*

4. **Pair Plot – Comparing Multiple Features**
   A pair plot shows the relationships between multiple numerical features using scatter plots and histograms.
   - **Purpose:** Identify feature relationships and clusters.
   - **Library Used:** *seaborn*
   - **Function Used:** *sns.pairplot(df, hue="Type", diag_kind="kde")*

5. **Heatmap – Feature Correlation Matrix**
   A heatmap represents the correlation between numerical features using colors.
   - **Purpose:** Identify highly correlated features (positive or negative).
   - **Library Used:** *seaborn*
   - **Function Used:** *sns.heatmap(numeric_df.corr(), annot=True, cmap="coolwarm")*

   Since the dataset contains non-numeric columns, I used *select_dtypes(include=['number'])* to exclude non-numeric values before computing correlations.

Implementation -

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Load the cleaned dataset
file_path = "/content/Cleaned_Accumulative_distribution.csv"
df = pd.read_csv(file_path)
# Display basic information about the dataset
print("Dataset Overview:")
print(df.info())
print(df.head())
```

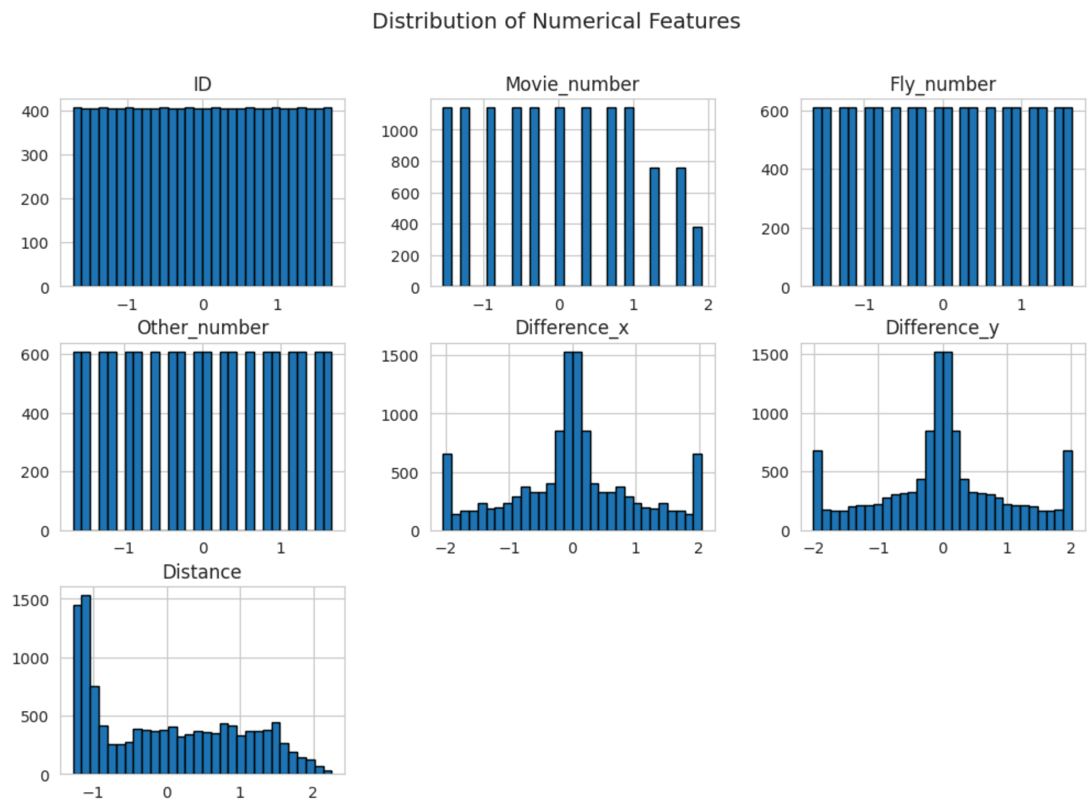
```
Dataset Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12160 entries, 0 to 12159
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   ID             12160 non-null  float64
 1   Type           12160 non-null  object
 2   Movie_number   12160 non-null  float64
 3   Fly_number     12160 non-null  float64
 4   Other_number   12160 non-null  float64
 5   Difference_x   12160 non-null  float64
 6   Difference_y   12160 non-null  float64
 7   Distance       12160 non-null  float64
dtypes: float64(7), object(1)
memory usage: 760.1+ KB
None
```

```
         ID          Type  Movie_number  Fly_number  Other_number  \
0 -1.731908  Dmelanogaster     -1.540178   -1.474087     -1.647509
1 -1.731623  Dmelanogaster     -1.540178   -1.300665     -1.647509
2 -1.731339  Dmelanogaster     -1.540178   -1.127243     -1.647509
3 -1.731054  Dmelanogaster     -1.540178   -0.953821     -1.647509
4 -1.730769  Dmelanogaster     -1.540178   -0.780399     -1.647509

   Difference_x  Difference_y  Distance
0     -0.291126      0.136903 -0.968219
1     -0.714968     -0.445300 -0.413533
2     -1.780003      1.731326  1.374617
3     -1.871965     -1.020045  0.942211
4     -0.289079      0.247088 -0.897781
```

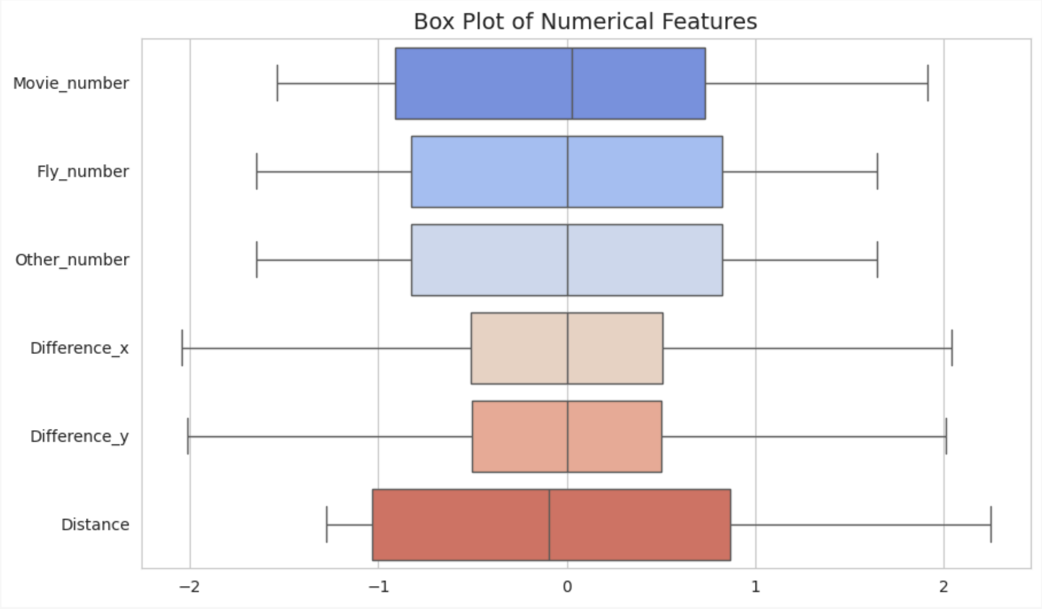
```
# Set a visual style for seaborn
sns.set_style("whitegrid")
# 1. **Distribution of Numerical Features**
plt.figure(figsize=(10, 6))
df.hist(bins=30, figsize=(12, 8), edgecolor='black')
plt.suptitle("Distribution of Numerical Features", fontsize=14)
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```

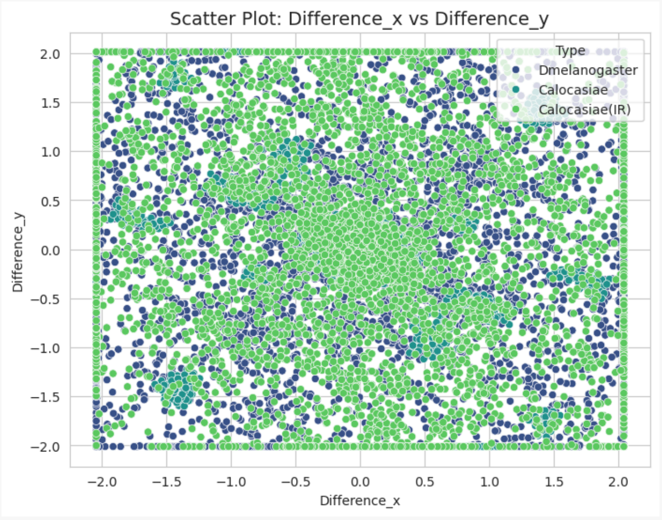



```
# 2. **Box Plot for Outlier Detection**
plt.figure(figsize=(10, 6))
sns.boxplot(data=df.iloc[:, 2:], orient="h", palette="coolwarm")
```

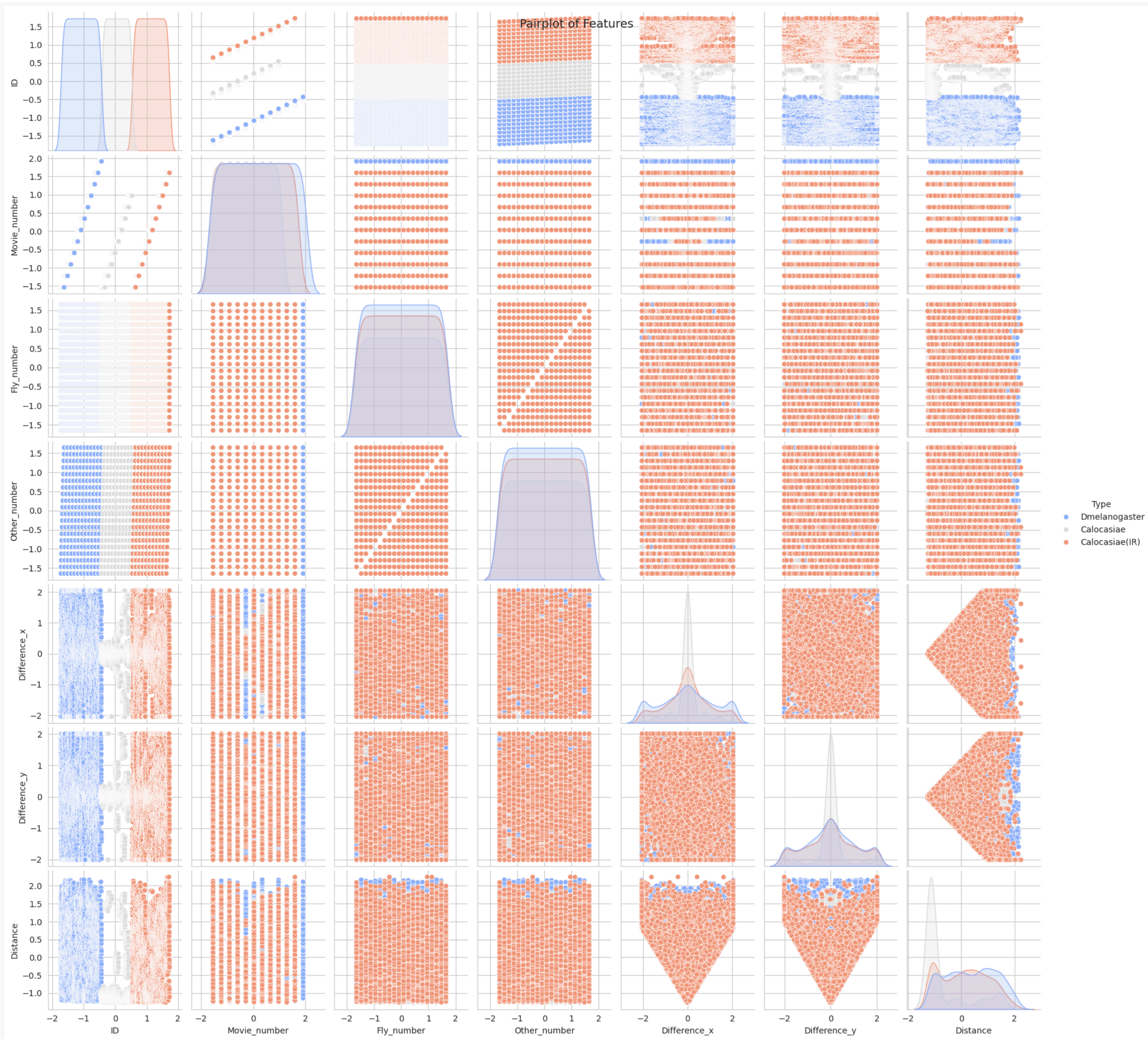
```
plt.title("Box Plot of Numerical Features", fontsize=14)
plt.show()
```




```
# 3. **Scatter Plot (Difference_x vs Difference_y)**
plt.figure(figsize=(8, 6))
sns.scatterplot(x=df["Difference_x"], y=df["Difference_y"], hue=df["Type"], palette="viridis")
plt.title("Scatter Plot: Difference_x vs Difference_y", fontsize=14)
plt.xlabel("Difference_x")
plt.ylabel("Difference_y")
plt.legend(title="Type")
plt.show()
```



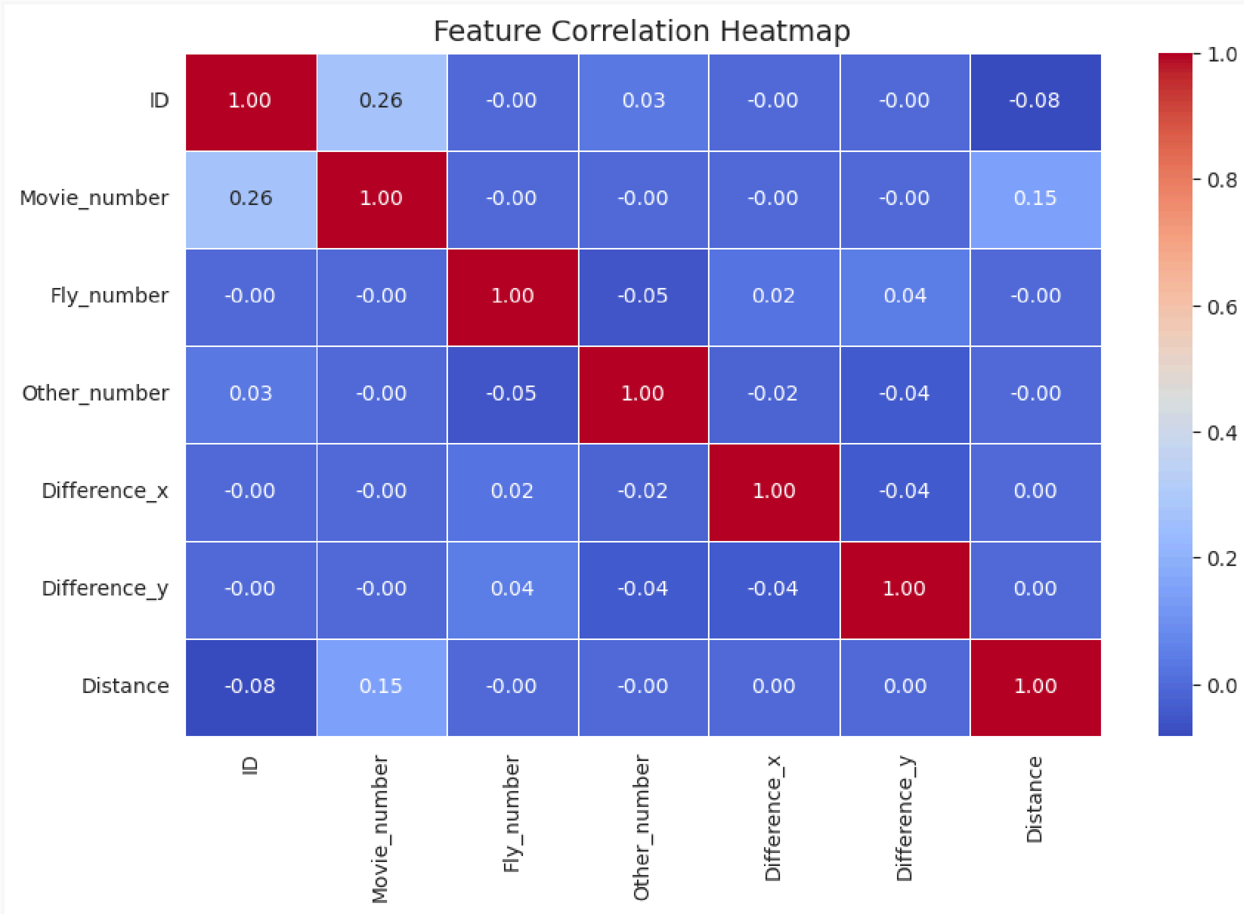

```
# 4. **Pairplot to Visualize Feature Relationships**
sns.pairplot(df, hue="Type", diag_kind="kde", palette="coolwarm")
plt.suptitle("Pairplot of Features", fontsize=14)
plt.show()
```




```
# 5. **Heatmap for Correlation (Only Numeric Columns)**
plt.figure(figsize=(10, 6))

# Select only numeric columns
numeric_df = df.select_dtypes(include=['number'])

sns.heatmap(numeric_df.corr(), annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Feature Correlation Heatmap", fontsize=14)
plt.show()
```



Dataset - Insects Flight Dynamics 

Conclusion -
Hence, data visualization is an essential step in Exploratory Data Analysis (EDA) allowing us to
✔ Detect patterns and anomalies
✔ Identify relationships between features
✔ Understand distributions and data spread
By using Matplotlib and Seaborn, data visualization was performed on the cleaned dataset.