DAI_101 Assignment 1 Report

Name – Tirthankar De

Enrollment No – 23324023

Batch – BP1

This report provides an exhaustive analysis of the notebook file for the assignment. My study focused on data analysis using techniques like EDA and visualization, using charts and plots and Python libraries to explore and interpret a dataset. The dataset contains information about meteorite landings, including attributes such as type, classification, mass, year of fall, latitude, and longitude. This report includes a detailed breakdown of the data processing and visualization process, and the results obtained.

Dataset Overview

Structure and Attributes

The dataset contains several attributes that describe each meteorite's characteristics. The key attributes include:

1. Name: The name of the meteorite.
2. Nametype: Indicates whether the name is valid or not.
3. Recclass: Classification of the meteorite.
4. Mass (g): Mass of the meteorite in grams.
5. Fall: Indicates whether the meteorite fell or was found.
6. Year: The year when the meteorite fell or was found.
7. Reclat: Latitude of meteorite impact.
8. Reclong: Longitude of meteorite impact.

Data Characteristics

The statistical summary reveals significant variability in attributes such as mass and year:

- Meteorite masses have a range of 7 orders of magnitude.
- Year ranges from 860 to 2101, indicating historical records and future projections.

Latitude and longitude values also exhibit wide ranges:

- Latitude varies from -87.37 to 81.17 degrees.
- Longitude spans from -165.43 to 178.20 degrees.

These broad ranges suggest a diverse dataset with global coverage.

Coding Analysis

Data Loading and Initial Exploration

We import essential Python libraries such as pandas for data manipulation and matplotlib/seaborn for visualization. The dataset is loaded into a pandas.DataFrame, enabling efficient exploration and analysis.

Code Explanation:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset

data = pd.read_csv('meteorite_landings.csv')

# Display basic information about the dataset

print(data.info())

print(data.describe())
```

Here:

- pd.read_csv() loads the meteorite landings dataset into a DataFrame.
- data.info() provides an overview of column types and non-null counts.
- data.describe() generates statistical summaries for numerical columns.

The initial exploration identifies missing values and data types for each column.

Data Cleaning

Missing values are handled by imputing them with appropriate statistics or removing rows/columns with excessive nulls.

Code Explanation:

```
data = data.dropna(subset=['mass (g)', 'year', 'reclat', 'reclong'])
```

This snippet removes rows where critical columns (mass (g), year, reclat, reclong) contain null values.

Statistical Analysis

The notebook calculates descriptive statistics to understand distributions and detect anomalies in numerical data.

Code Explanation:

```
stats = data.describe()

print(stats)
```

This generates metrics like mean, standard deviation, min/max values for numerical columns.

Visualization Techniques

Distribution Plots

Visualizing distributions helps identify patterns such as skewness or outliers in numerical attributes like mass or year.

Code Explanation:

```
sns.histplot(data['mass (g)'], bins=50, kde=True)

plt.title('Distribution of Meteorite Mass')

plt.xlabel('Mass (g)')

plt.ylabel('Frequency')

plt.show()
```

Here:

- sns.histplot() creates a histogram with kernel density estimation (KDE).
- The plot reveals that most meteorites have relatively small masses compared to a few huge ones.

Scatter Plots

Scatter plots illustrate relationships between numerical variables, such as latitude (reclat) and longitude (reclong).

Code Explanation:

```
plt.scatter(data['reclong'], data['reclat'], alpha=0.5)

plt.title('Geographical Distribution of Meteorites')

plt.xlabel('Longitude')

plt.ylabel('Latitude')

plt.show()
```

This plot highlights global coverage of meteorite landings.

Box Plots

Box plots are used to visualize distributions while emphasizing quartiles and outliers.

Code Explanation:

```
sns.boxplot(x='fall', y='mass (g)', data=data)

plt.title('Meteorite Mass by Fall Type')

plt.xlabel('Fall Type')

plt.ylabel('Mass (g)')

plt.show()
```

Here:

- sns.boxplot() compares mass distributions between fallen vs found meteorites.
- Fallen meteorites tend to have smaller masses than found ones.

Results Interpretation

Key Insights from Statistical Analysis

1. Mass Distribution:
    - Most meteorites have small masses (<500 grams).
    - A few outliers exceed millions of grams, skewing the mean upward.
2. Year Trends:
    - Meteorites are predominantly recorded post-1900.
    - Historical records before 1900 are sparse but valuable for understanding long-term trends.
3. Geographical Spread:
    - Meteorites are distributed globally but cluster near populated regions due to observational bias.

Visual Patterns

1. Mass Histogram:
    - The histogram confirms a skewed distribution, with most values concentrated at lower masses.
2. Geographical Scatter Plot:
    - Clusters near specific latitudes/longitudes suggest observational biases or geological factors influencing landing sites.
3. Box Plot Insights:
    - Fallen meteorites exhibit narrower mass ranges than found ones, likely due to detection limitations for smaller fragments.

Conclusion

We demonstrate the use of Python and the effectiveness of exploratory data analysis and visualisation techniques for the study of scientific datasets like meteorite landings. Key findings include significant variability in mass and year attributes, global distribution patterns influenced by observational biases, and distinct characteristics between fallen vs found meteorites.