DA Handling Missing Values and Outliers in a Dataset

Handling Missing Values and Outliers in a Dataset

Task Overview

This exercise focuses on identifying, analyzing, and handling missing values and outliers in a real dataset. It builds on the previous concepts of data integration and summarization but emphasizes data cleaning techniques.

Task Steps:

1. Find a Real Dataset

- Search online for a **real dataset** that contains missing values and potential outliers. Possible topics include:
  - Public health (e.g., patient records, disease statistics)
  - Climate data (e.g., temperature trends, air pollution levels)
  - Finance (e.g., stock market data, cryptocurrency prices)
  - Retail (e.g., product prices, sales performance)
  - Sports (e.g., player statistics, team performance)
  - **Air Quality in Portugal**
- Justify why you chose the dataset and describe its importance.

2. Load and Explore the Dataset

- Download the dataset and **load it into Python using Pandas**.
- Perform initial data exploration using:
  - `df.head()` – Display the first few rows.
  - `df.info()` – Check dataset structure, column types, and missing values.
  - `df.describe()` – Generate summary statistics.
- Identify columns with missing values using:

```
missing_values = df.isnull().sum()
print(missing_values)
```

3. Handling Missing Values

- Determine the percentage of missing values in each column:

```
missing_percentage = (df.isnull().sum() / len(df)) * 100
print(missing_percentage)
```

- Choose an appropriate strategy to handle missing values:
  - **Remove missing values** if they are a small fraction of the dataset:

    ```
    df_cleaned = df.dropna()
    ```

  - **Impute missing values** using:
    - Mean for numerical columns:

      ```
      df["column_name"].fillna(df["column_name"].mean(), inplace=True)
      ```

    - Mode for categorical columns:

      ```
      df["category_column"].fillna(df["category_column"].mode()[0], inplace=True)
      ```

4. Identifying Outliers

- Use the **Interquartile Range (IQR)** method to detect outliers:

```
Q1 = df["column_name"].quantile(0.25)
Q3 = df["column_name"].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = df[(df["column_name"] < lower_bound) | (df["column_name"] > upper_bound)]
print(outliers)
```

- Use a **box plot** to visualize outliers:

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8, 5))
sns.boxplot(x=df["column_name"])
plt.title("Box Plot for Outlier Detection")
plt.show()
```

5. Handling Outliers

- Choose an appropriate strategy to handle outliers:
  - **Remove outliers** if they significantly distort the dataset:

```
df_filtered = df[(df["column_name"] >= lower_bound) & (df["column_name"] <= upper_bound)]
```

- **Transform the data** using log transformation:

```
df["column_name"] = np.log1p(df["column_name"])
```

- **Cap the outliers** to upper/lower thresholds:

```
df["column_name"] = df["column_name"].clip(lower=lower_bound, upper=upper_bound)
```

6. Present Findings

- **Explain dataset characteristics:**
  - The source of the dataset.
  - Number of records and columns.
  - Key observations from missing value and outlier analysis.
- **Summarize cleaning strategies:**
  - What columns had missing values, and how were they handled?
  - What columns contained outliers, and how were they treated?
  - How did cleaning impact data distribution?

Submission Requirements:

- Submit your cleaned dataset (CSV format).
- Include Python code and visualizations in a Jupyter Notebook or a Python script.
- Provide a short summary (one page) explaining missing values, outlier handling, and any challenges faced.

Bonus Task (Optional):

- Perform **advanced imputation** using machine learning models like KNN or regression.
- Compare different outlier removal techniques and discuss their effects on data trends.