TASK 2: EXPLORATORY DATA ANALYSIS (EDA) ....

Task 5: Exploratory Data Analysis (EDA)

Objective

Perform a comprehensive Exploratory Data Analysis (EDA) on the Titanic dataset to extract meaningful insights using statistical and visual exploration techniques.

Dataset

- **Name:** Titanic Dataset (as recommended in task description)
- **Source:** Provided by Elevate Labs
- **Description:** Passenger details such as demographics, ticket class, fare, and survival status.

Tools Used

- **Python** – Data analysis and scripting
- **Pandas** – Data manipulation and inspection
- **Matplotlib** – Data visualization
- **Seaborn** – Advanced and aesthetic plotting.

EDA Process

1. Initial Data Inspection

- `.info()` – Data types and missing values
- `.describe()` – Statistical summaries
- `.value_counts()` – Categorical variable distributions

2. Data Visualization

- **Histograms** – Numerical distributions
- **Boxplots** – Outlier detection
- **Scatterplots** – Relationship analysis
- **Heatmap** – Correlation identification
- **Pairplot** – Pairwise relationships and distributions

3. Key Insights

- Females had a significantly higher survival rate than males.
- Passengers in higher classes had better survival rates.
- Younger passengers had slightly better survival chances.
- Higher fares correlated with better survival probability.

Step-by-Step EDA on Global Superstore Dataset

1.Load the Dataset.

```
-> import pandas as pd

df = pd.read_csv('Global_Superstore.csv')

 # Replace with your actual file path

df.head()
```

2. Data Cleaning.

-> Handling missing values........

```
-> df.isnull().sum()

# Check missing values

df.fillna(df.mean(numeric_only=True), inplace=True)

 # Fill numeric columns with mean
```

3. Data Exploration.

```
-> # Check shape
df.shape
# Check data types
df.dtypes
# Check for missing values
df.isnull().sum()
```

4. Data Cleaning.

```
->  Remove duplicates
df = df.drop_duplicates()

# Handle missing values (example: fill with median)
df = df.fillna(df.median(numeric_only=True))

# Detect and handle outliers using IQR
Q1 = df.quantile(0.25, numeric_only=True)
Q3 = df.quantile(0.75, numeric_only=True)
IQR = Q3 - Q1
df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]
```

5. Statistical Analysis.

```
-> # Descriptive statistics
df.describe()

# Correlation matrix
df.corr(numeric_only=True)
```

6. Data Visualization.

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

# Histogram for Sales
plt.figure(figsize=(8,4))
sns.histplot(df['Sales'], bins=30)
plt.title('Sales Distribution')
plt.show()

# Boxplot for Profit
plt.figure(figsize=(8,4))
sns.boxplot(x=df['Profit'])
plt.title('Profit Boxplot')
plt.show()

# Heatmap for correlations
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```