# Python Lab: Exploratory Data Analysis (EDA)

## Objective

This lab will guide you through the process of performing Exploratory Data Analysis (EDA) using a sample dataset. You will learn how to clean data, apply basic transformations, conduct univariate and bivariate analysis, and perform correlation analysis using visualizations.

## Prerequisites

Ensure you have the following libraries installed using pip:  
```bash  
pip install pandas numpy seaborn matplotlib  
```

## Step 1: Load the Dataset

Download the dataset (`sample\_data.csv`) and load it using Pandas. First, view the first few rows using the `.head()` method.

```python  
import pandas as pd  
  
# Load the dataset  
df = pd.read\_csv('sample\_data.csv')  
  
# Display the first few rows  
print(df.head())  
```

## Step 2: Data Cleaning

Data cleaning is essential to handle missing or incorrect data. First, check for missing values using `.isnull().sum()`.

```python  
# Check for missing values  
print('Missing Values Before Cleaning:')  
print(df.isnull().sum())  
  
# Fill missing values with median  
df['Income'].fillna(df['Income'].median(), inplace=True)  
df['SpendingScore'].fillna(df['SpendingScore'].median(), inplace=True)  
  
# Verify missing values  
print('Missing Values After Cleaning:')  
print(df.isnull().sum())  
```

## Step 3: Basic Transformations

Some datasets have categorical values that need to be converted to numeric values for further analysis. Here, convert the 'Gender' column to numeric using `map()`.

```python  
df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})  
print('Converted Gender to Numeric:')  
print(df.head())  
```

## Step 4: Univariate Analysis

Univariate analysis involves analyzing individual variables using visualizations like histograms and count plots.

```python  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
fig, axes = plt.subplots(2, 2, figsize=(12, 10))  
  
sns.histplot(df['Age'], ax=axes[0, 0], kde=True)  
axes[0, 0].set\_title('Age Distribution')  
  
sns.histplot(df['Income'], ax=axes[0, 1], kde=True)  
axes[0, 1].set\_title('Income Distribution')  
  
sns.histplot(df['SpendingScore'], ax=axes[1, 0], kde=True)  
axes[1, 0].set\_title('Spending Score Distribution')  
  
sns.countplot(x='EducationYears', data=df, ax=axes[1, 1])  
axes[1, 1].set\_title('Education Years Count')  
  
plt.tight\_layout()  
plt.show()  
```

## Step 5: Bivariate Analysis

Bivariate analysis is used to explore relationships between two variables. Visualizations like scatter plots help identify patterns.

```python  
plt.figure(figsize=(8, 6))  
sns.scatterplot(x='Income', y='SpendingScore', hue='Gender', data=df)  
plt.title('Income vs Spending Score')  
plt.show()  
```

## Step 6: Correlation Analysis

Correlation analysis measures the relationship between numerical variables using correlation coefficients.

```python  
# Calculate correlation matrix  
correlation\_matrix = df.corr()  
print(correlation\_matrix)  
  
# Visualize correlation using a heatmap  
plt.figure(figsize=(8, 6))  
sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')  
plt.title('Correlation Matrix')  
plt.show()  
```

## Conclusion

In this lab, you have performed data cleaning, basic transformations, univariate and bivariate analysis, and correlation analysis using visualizations. These steps are essential for understanding data characteristics before applying machine learning algorithms.