# Python Lab: Visualizations with Pandas, Matplotlib, and Seaborn

## Objective

This lab will guide you through creating different types of visualizations using Python. You'll use a sample dataset containing sales, profit, customer count, and satisfaction scores for different product categories over five months.

## Prerequisites

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

## Loading the Dataset

First, download the sample dataset (`visualization\_data.csv`) and load it using Pandas. You can display the first few rows using the `.head()` method.

```python  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# Load the dataset  
df = pd.read\_csv('visualization\_data.csv')  
print(df.head())  
```

## 1. Bar Chart - Visualizing Sales by Category

A bar chart is useful for comparing categorical data. In this example, we'll visualize the total sales for each category.

```python  
plt.figure(figsize=(8,6))  
sns.barplot(x='Category', y='Sales', data=df)  
plt.title('Sales by Category')  
plt.show()  
```

## 2. Pie Chart - Visualizing Profit Distribution

A pie chart is used to show the proportion of each category’s profit in the total profit.

```python  
plt.figure(figsize=(8,6))  
plt.pie(df['Profit'], labels=df['Category'], autopct='%1.1f%%', startangle=90)  
plt.title('Profit Distribution by Category')  
plt.axis('equal')  
plt.show()  
```

## 3. Line Chart - Visualizing Customer Growth Over Months

A line chart is ideal for observing trends over time. Here, we visualize the growth in customer count.

```python  
plt.figure(figsize=(8,6))  
sns.lineplot(x='Month', y='Customer\_Count', data=df, marker='o')  
plt.title('Customer Growth Over Months')  
plt.show()  
```

## 4. Histogram - Understanding Sales Distribution

A histogram helps in understanding the distribution of numerical data. In this example, we visualize the sales distribution.

```python  
plt.figure(figsize=(8,6))  
sns.histplot(df['Sales'], kde=True)  
plt.title('Sales Distribution')  
plt.show()  
```

## 5. Box Plot - Detecting Outliers in Profits

Box plots are used to identify outliers and visualize the spread of data. Here, we analyze the profit distribution.

```python  
plt.figure(figsize=(8,6))  
sns.boxplot(y='Profit', data=df)  
plt.title('Profit Box Plot')  
plt.show()  
```

## 6. Scatter Plot - Visualizing Sales vs. Profit

A scatter plot is used to observe the relationship between two numerical variables. We’ll plot sales vs. profit to identify any correlation.

```python  
plt.figure(figsize=(8,6))  
sns.scatterplot(x='Sales', y='Profit', data=df)  
plt.title('Sales vs Profit')  
plt.show()  
```

## 7. Heatmap - Correlation Analysis

A heatmap visualizes the correlation between numeric variables. It helps in identifying strong or weak relationships.

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

## 8. Joint Plot - Detailed Bivariate Analysis

A joint plot provides both a scatter plot and distribution of variables. It is useful for understanding variable relationships and data spread.

```python  
sns.jointplot(x='Sales', y='Profit', data=df, kind='reg')  
plt.show()  
```

## Conclusion

In this lab, you have learned how to create various visualizations using Python libraries. These visualizations will help you gain insights into data, detect patterns, and support decision-making in real-world scenarios.