Machine Learning Lab

Experiment 3: Exploratory Data Visualization

For CSE Department, Semester 06

Course Code: U23CM6L2

Compiled by
Mohammed Ufraan

March 24, 2026

Experiment 3

Aim: Use a dataset in a `.csv` file containing information about books to perform Exploratory Data Visualization with the following steps.

a) Importing Libraries

Description: Importing the required Python libraries for data handling and visualization, including `pandas` for dataset operations, `matplotlib` for plotting, and `seaborn` for enhanced visual styling.

> **Dataset Download:** The dataset file `book.csv` used in this experiment is available for download at `https://github.com/ufraaan/ml-lab-experiments`. Datasets for all lab experiments are hosted there.

Input Format:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")
```

Listing 1: Importing required libraries

Algorithm:

1. Import `pandas` as `pd` for loading and manipulating the CSV dataset.
2. Import `matplotlib.pyplot` as `plt` for creating plots and charts.
3. Import `seaborn` as `sns` for enhanced visualization styling.
4. Set the seaborn style to `"whitegrid"` for cleaner plot backgrounds.

Viva Questions:

1. Why do we use `seaborn` in addition to `matplotlib`?
2. What is the role of `pandas` in data visualization?
3. What does `sns.set(style="whitegrid")` do?

---

b) Loading Dataset

Description: Load the CSV file containing book information into a pandas DataFrame and preview its contents.

Input Format:

- CSV file (`book.csv`) with columns such as `Title`, `Price`, `Pages`, `Rating`, `Year`.

Algorithm:

1. Read the CSV file using `pd.read_csv()`.
2. Preview the dataset using `df.head()`.
3. Check dataset structure using `df.info()` and `df.describe()`.

```python
df = pd.read_csv(r’C:\Users\ufraan\Documents\book.csv’)

print("Dataset preview")
print(df.head())
```

Listing 2: Loading the dataset and previewing contents

Expected Output:

- First five rows of the dataset displayed in tabular form.

Viva Questions:

1. What does `df.head()` return by default?
2. How do you check for missing values in a DataFrame?
3. What is the difference between `df.info()` and `df.describe()`?

---

c) **Plot Bar Graph, Scatter Plot, Box Plot, Histogram, Line Graph**

Description: Generate five different types of visualizations from the books dataset to explore relationships and distributions in the data.

PTO

i) Bar Graph — Price of Books

Algorithm:

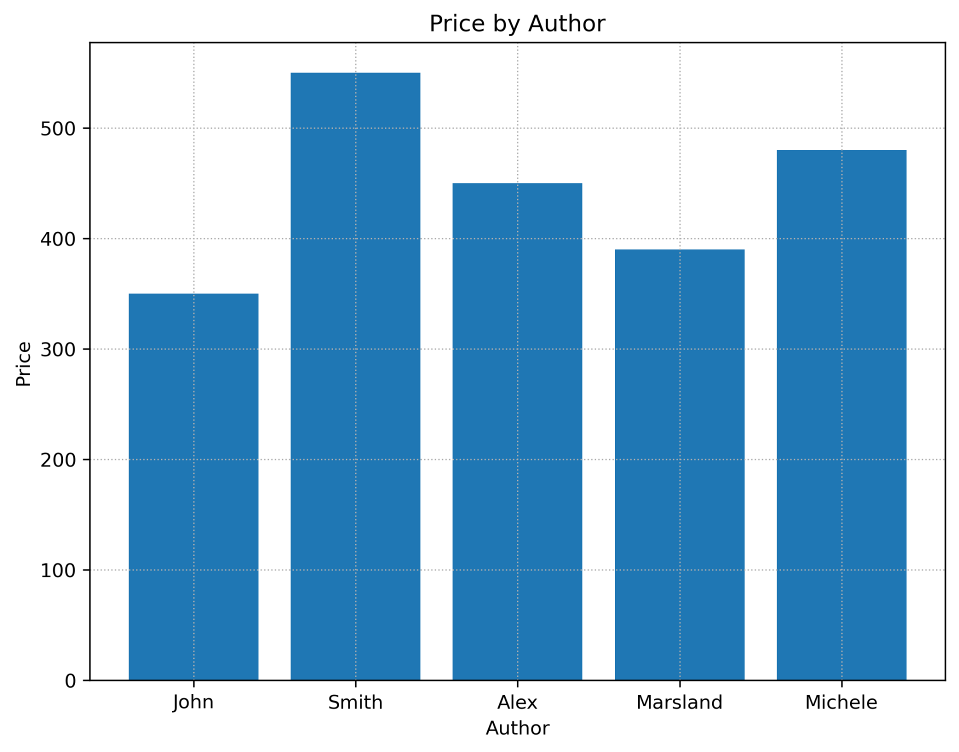
1. Create a new figure using `plt.figure()`.
2. Plot a bar graph of `Title` (x-axis) vs `Price` (y-axis).
3. Set axis labels and title.
4. Rotate x-axis tick labels by 45° for readability.
5. Display the plot using `plt.show()`.

```python
plt.figure()
plt.bar(df['Title'], df['Price'])
plt.xlabel("Book Title")
plt.ylabel("Price")
plt.title("Price of Books")
plt.xticks(rotation=45)
plt.show()
```

Listing 3: Bar Graph: Price of Books

Expected Output:

Output: Bar Graph — Price of Books



[H]

ii) Scatter Plot — Pages vs Price

Algorithm:

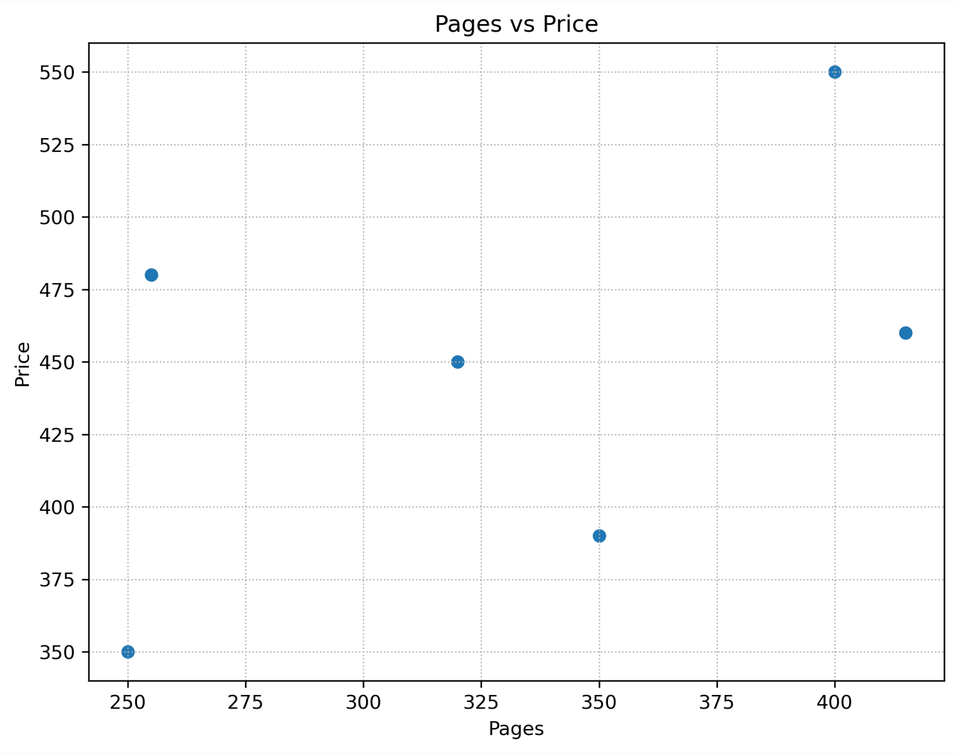
1. Create a new figure using `plt.figure()`.
2. Plot a scatter graph of `Pages` (x-axis) vs `Price` (y-axis).
3. Set axis labels and title.
4. Display the plot using `plt.show()`.

```
plt.figure()
plt.scatter(df['Pages'], df['Price'])
plt.xlabel("Number of pages")
plt.ylabel("Price")
plt.title("Pages vs Price")
plt.show()
```

Listing 4: Scatter Plot: Pages vs Price

Expected Output:

Output: Scatter Plot — Pages vs Price



[H]

iii) Box Plot — Book Ratings

Algorithm:

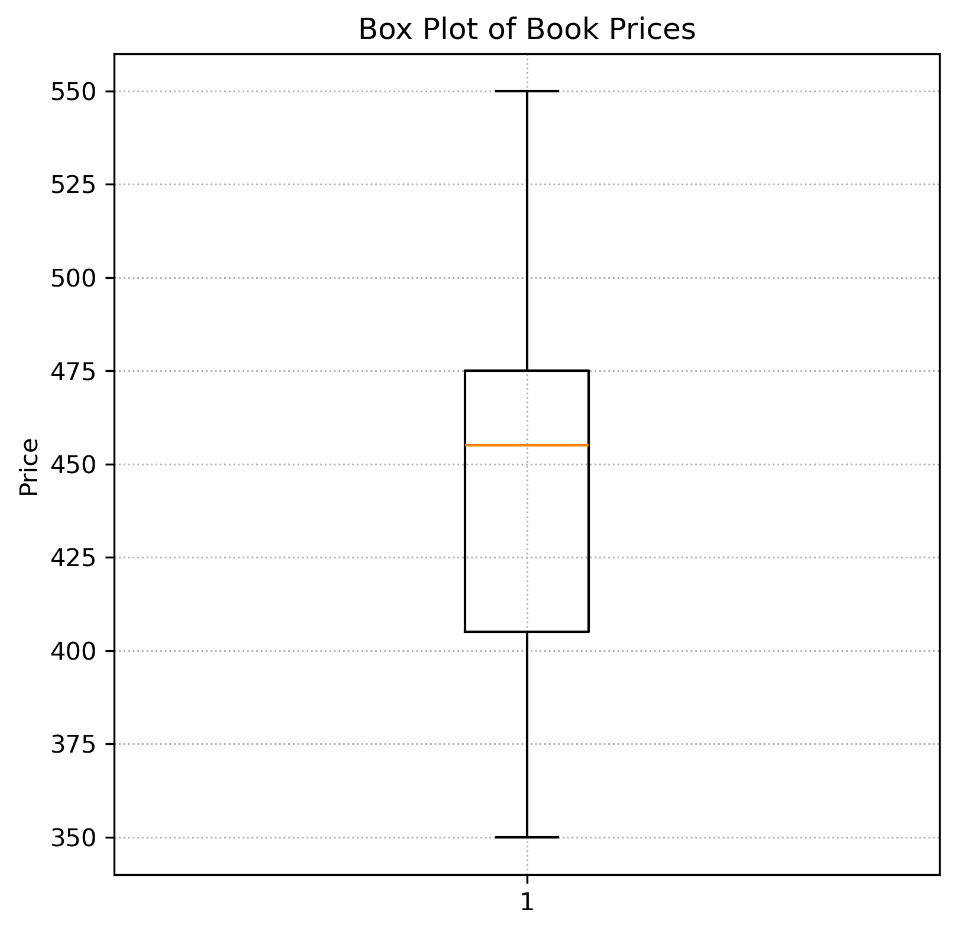
1. Create a new figure using `plt.figure()`.
2. Plot a box plot of the `Rating` column.
3. Set y-axis label and title.
4. Display the plot using `plt.show()`.

```
plt.figure()
plt.boxplot(df['Rating'])
plt.ylabel("Rating")
plt.title("Box plot of Book ratings")
plt.show()
```

Listing 5: Box Plot: Book Ratings

Expected Output:

Output: Box Plot — Book Ratings



[H]

iv) Histogram — Distribution of Pages

Algorithm:

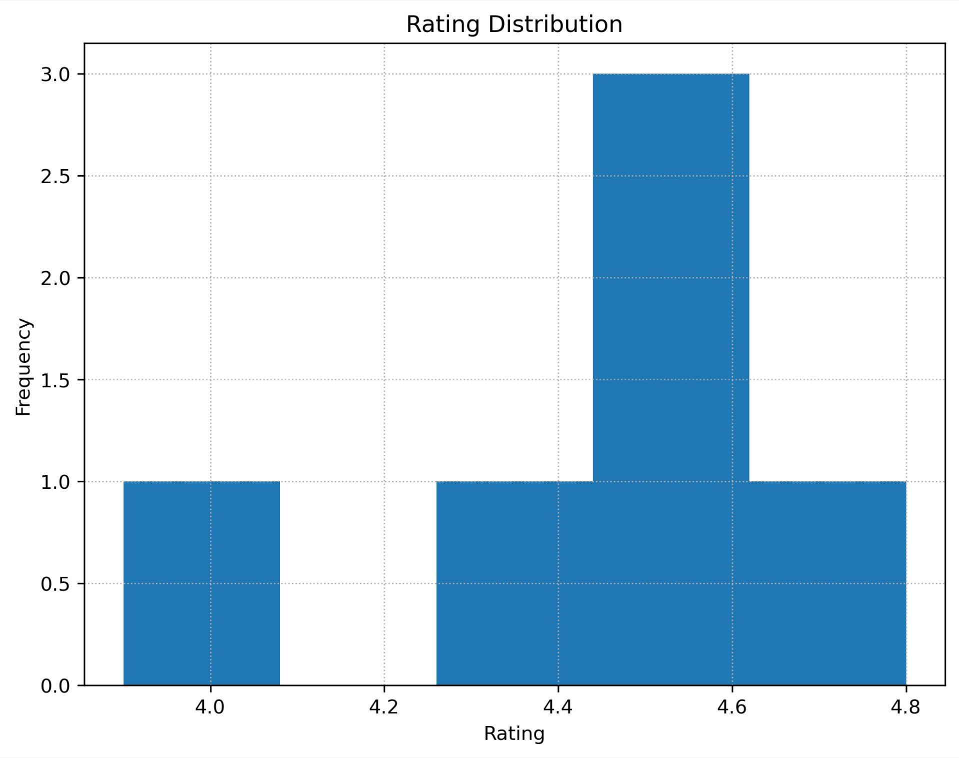
1. Create a new figure using `plt.figure()`.
2. Plot a histogram of the `Pages` column with 5 bins.
3. Set axis labels and title.
4. Display the plot using `plt.show()`.

```python
plt.figure()
plt.hist(df['Pages'], bins=5)
plt.xlabel("Pages")
plt.ylabel("Frequency")
plt.title("Histogram of pages")
plt.show()
```

Listing 6: Histogram: Distribution of Pages

Expected Output:

Output: Histogram — Distribution of Pages



[H]

v) Line Graph — Year vs Rating

Algorithm:

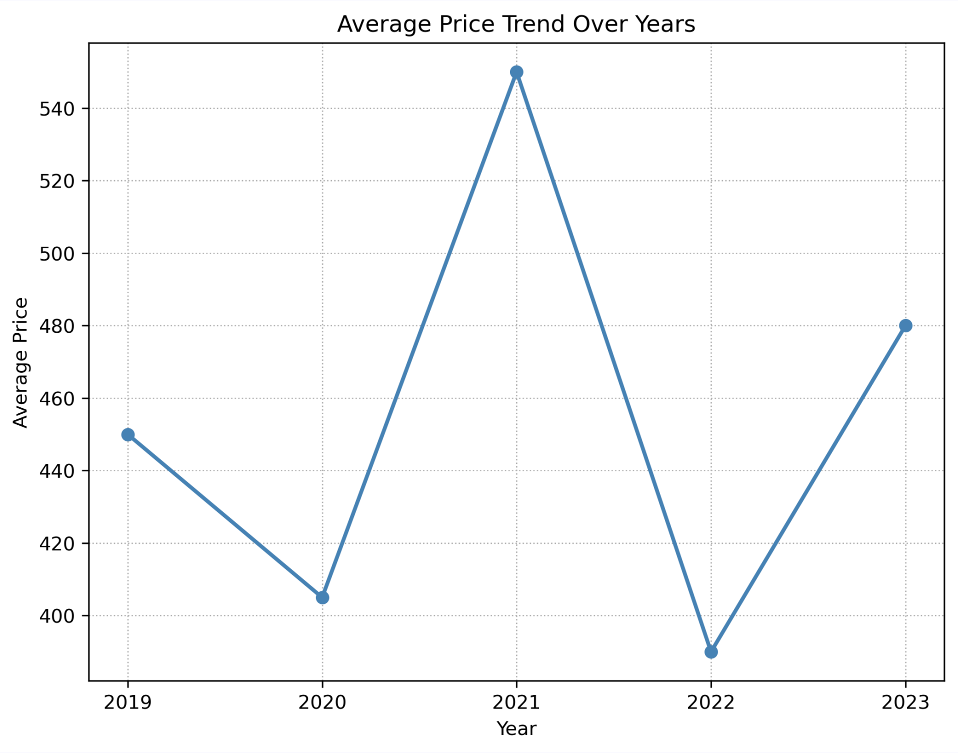
1. Create a new figure using `plt.figure()`.
2. Plot a line graph of `Year` (x-axis) vs `Rating` (y-axis) with marker `'o'`.
3. Set axis labels and title.
4. Display the plot using `plt.show()`.

```
plt.figure()
plt.plot(df['Year'], df['Rating'], marker='o')
plt.xlabel("Years")
plt.ylabel("Rating")
plt.title("Year vs Rating")
plt.show()
```

Listing 7: Line Graph: Year vs Rating

Expected Output:

Output: Line Graph — Year vs Rating



[H]

Viva Questions:

1. What is the difference between a bar graph and a histogram?
2. When would you prefer a scatter plot over a line graph?
3. What information does a box plot convey that a histogram does not?
4. What does the `bins` parameter control in `plt.hist()`?
5. How does a line graph help in identifying trends over time?