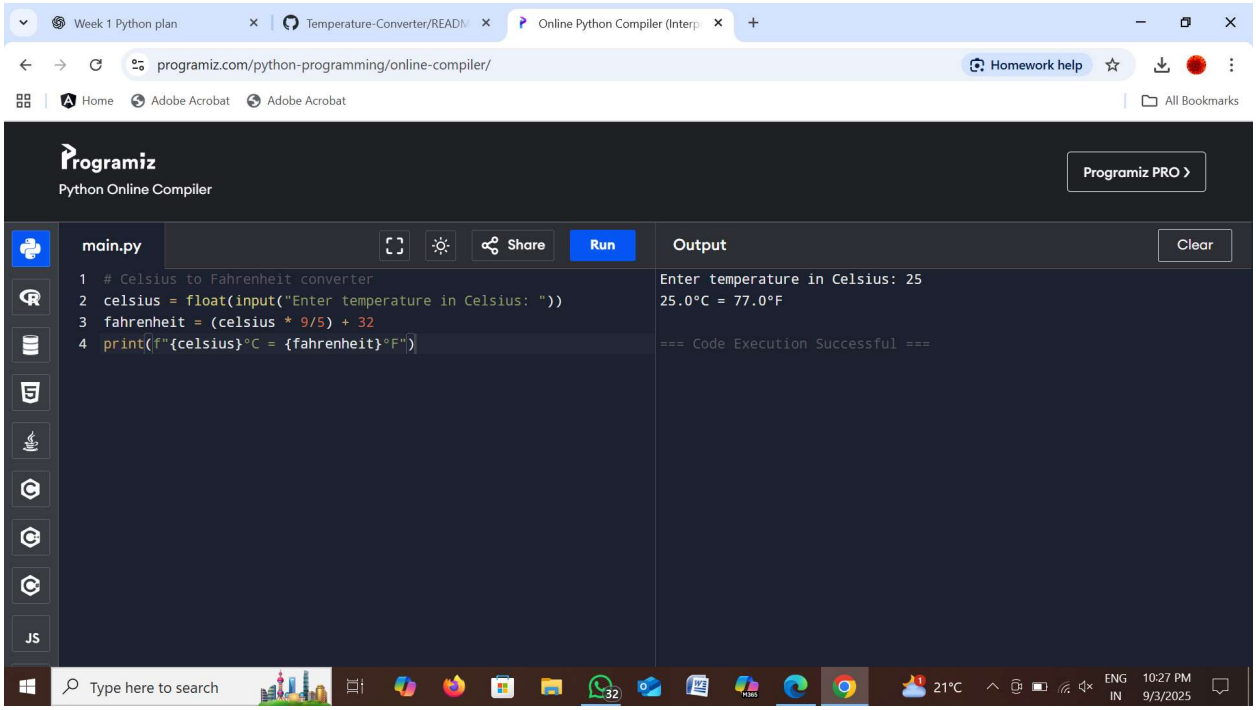
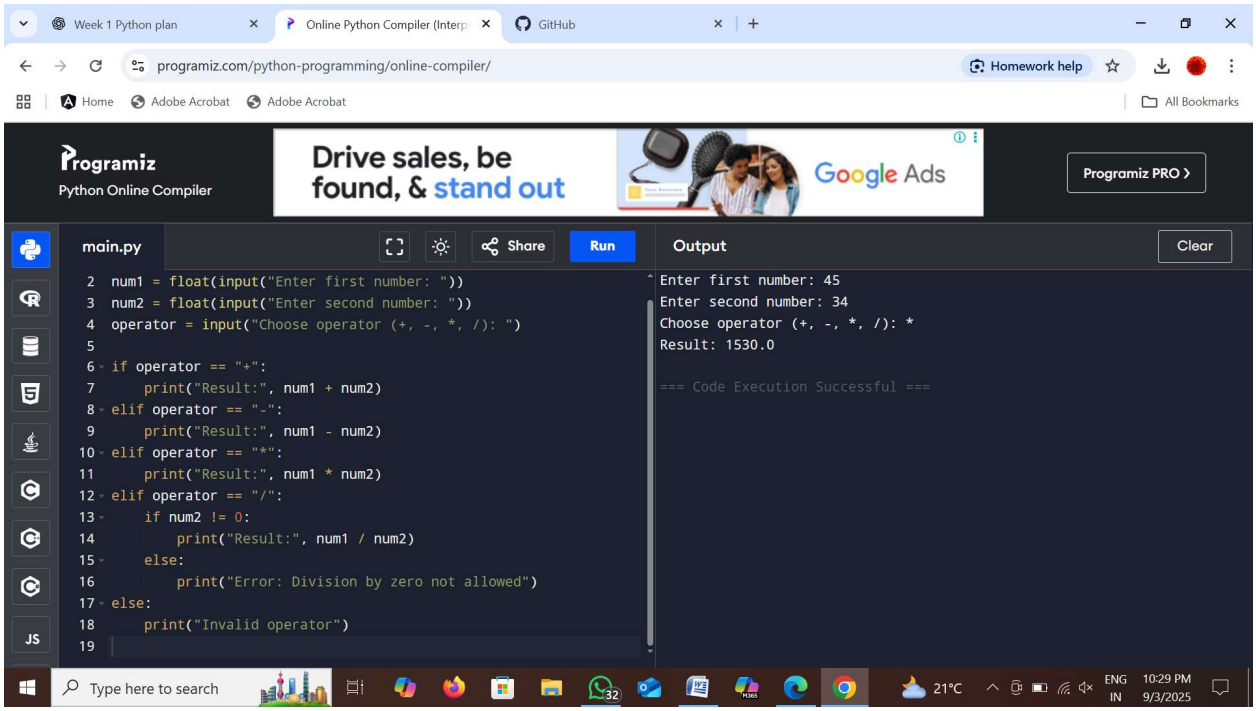
Python Programming

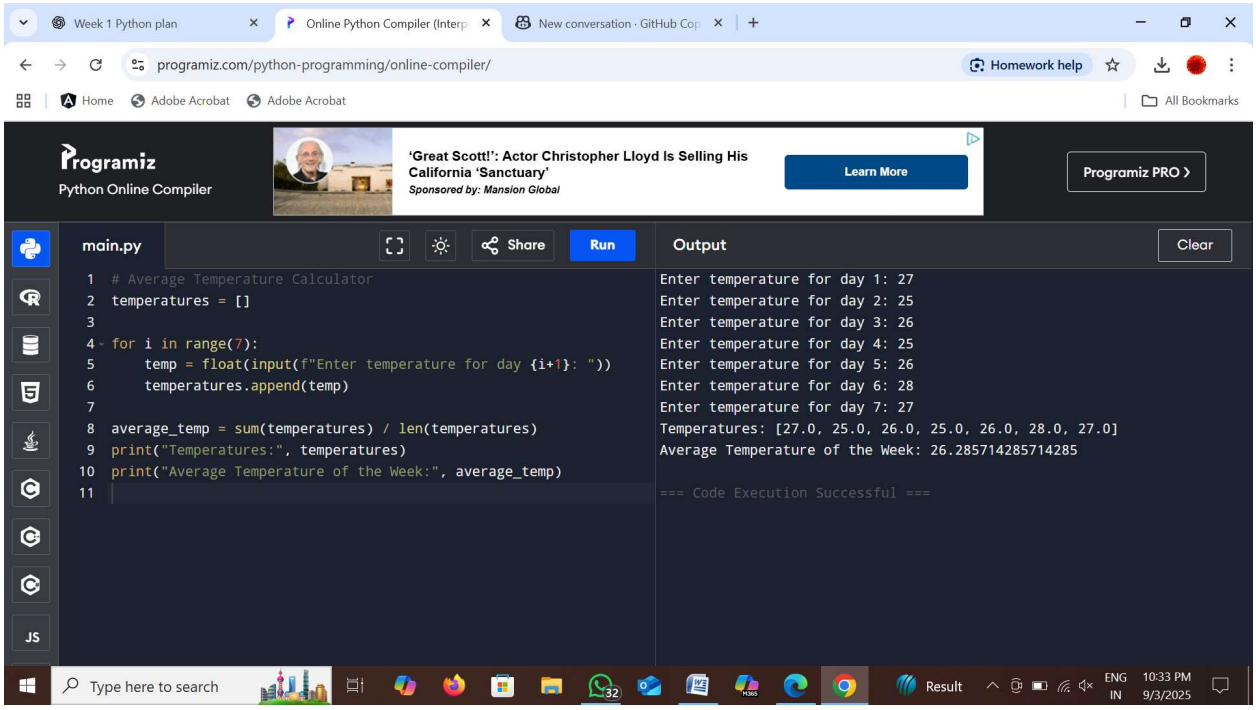
Week 1: Introduction to Python Programming

Hands-On: Write basic Python programs (e.g., temperature converter, calculator).



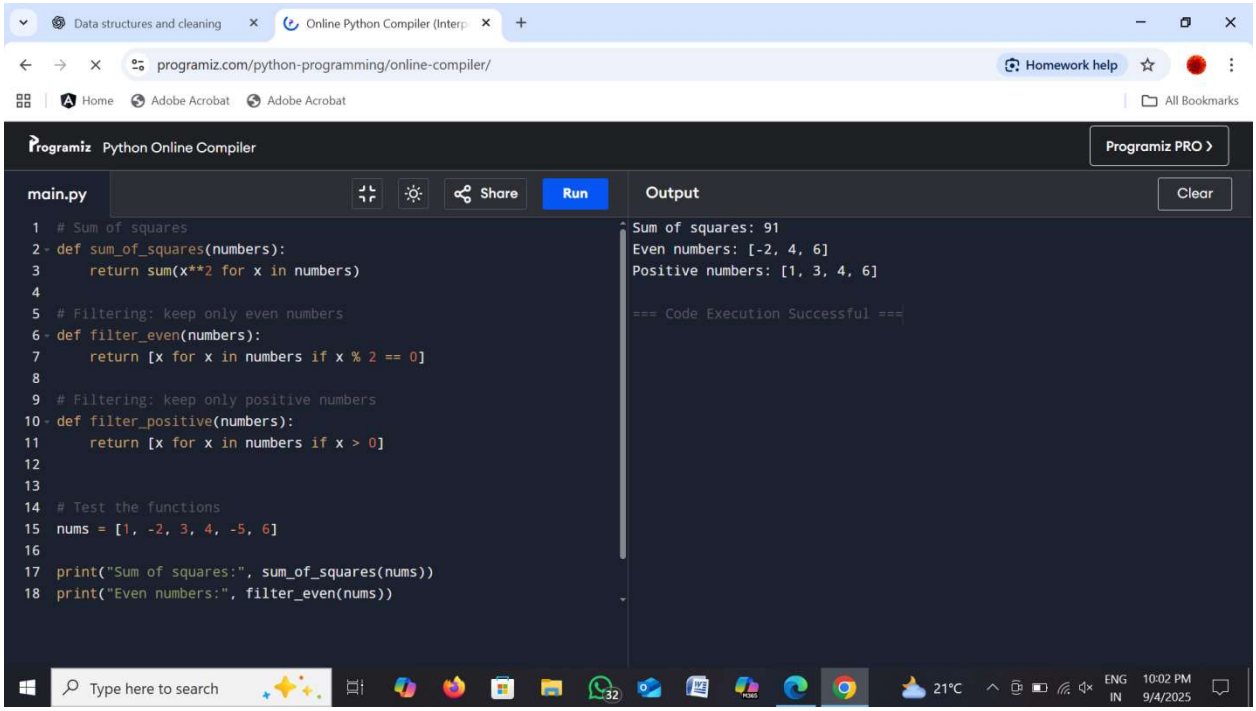


Client Project: Create a basic data processing script (e.g., calculating the average temperature).

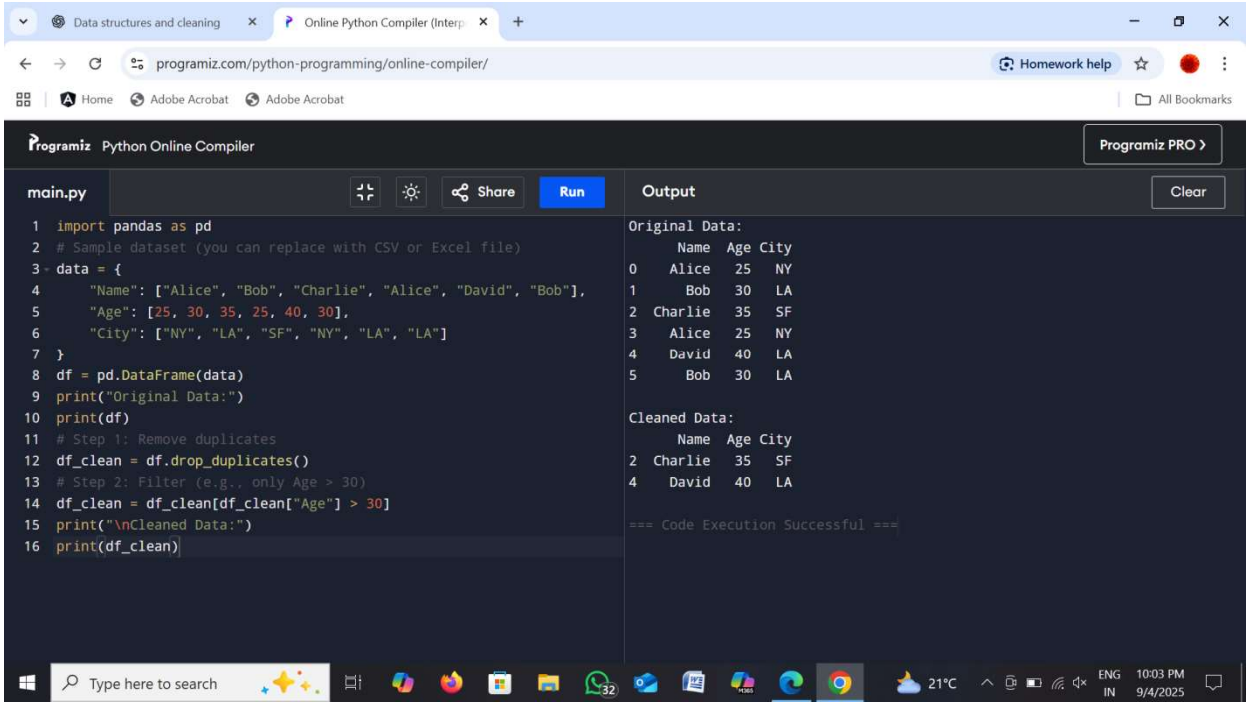


Week 2: Data Structures and Functions

Hands-On: Work with data structures and write functions for data transformations (e.g., sum of squares, filtering).

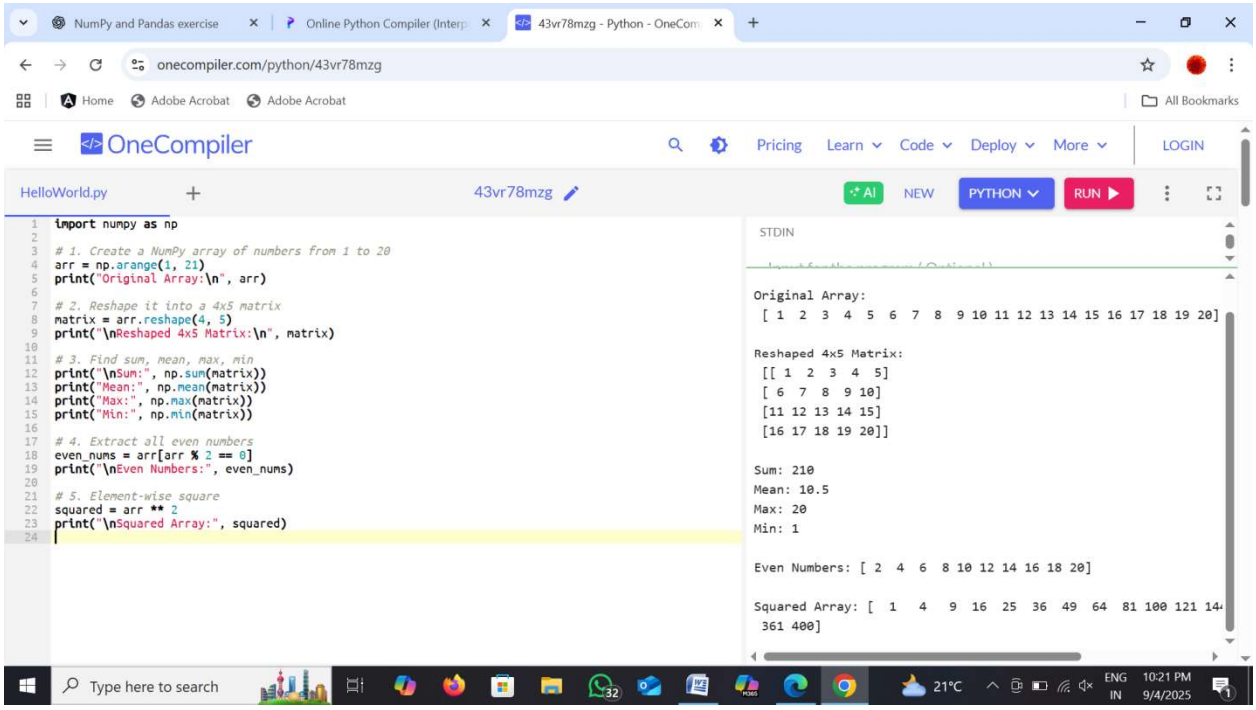


Client Project: Write a script for data cleaning (e.g., remove duplicates, filter data).

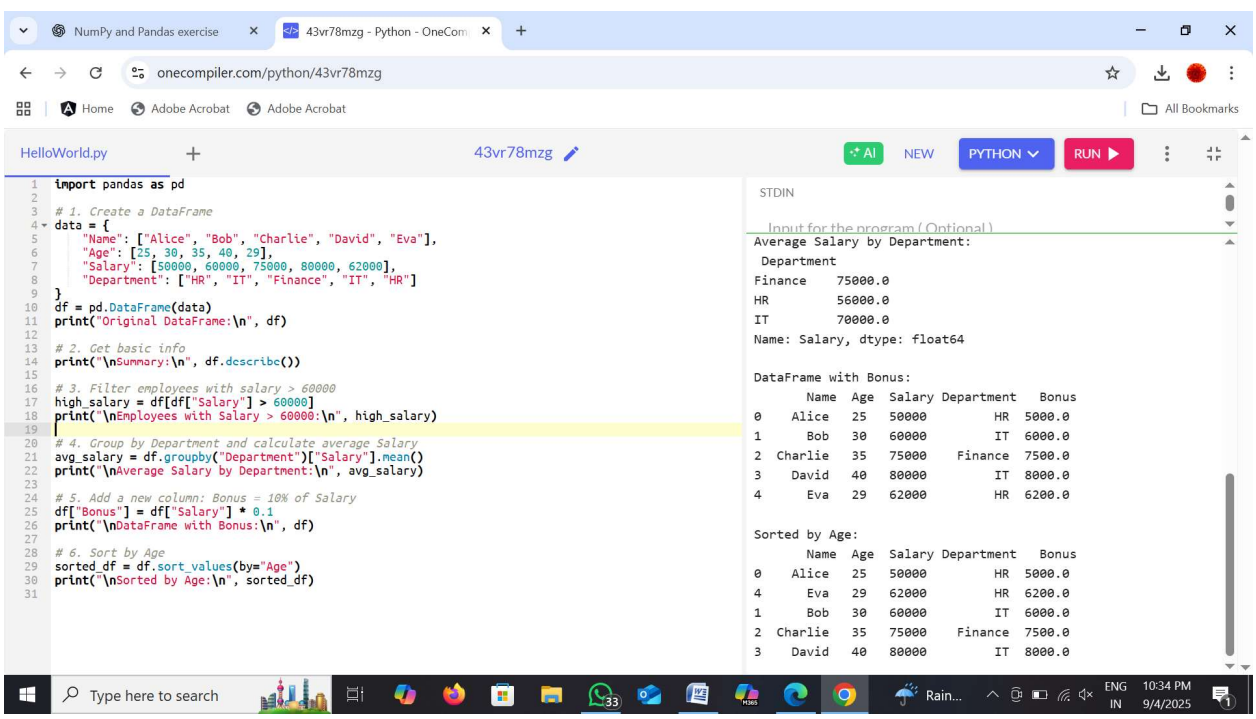


Week 3: NumPy and Pandas for Data

Hands-On: Perform operations with NumPy and manipulate datasets with Pandas.



Client Project: Clean and aggregate a dataset (e.g., remove missing values, calculate averages).



Output:

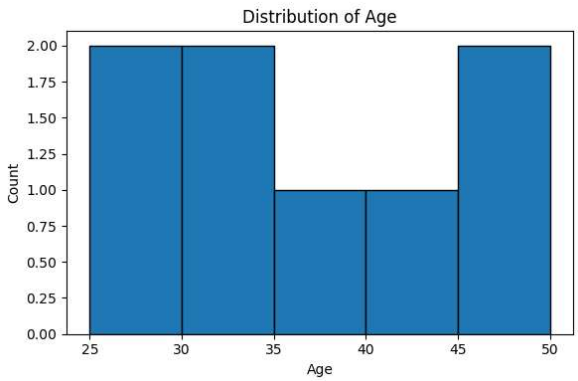
```
Original DataFrame:
       Name  Age  Salary Department
0    Alice   25   50000         HR
1      Bob   30   60000         IT
2  Charlie   35   75000    Finance
3    David   40   80000         IT
4      Eva   29   62000         HR

Summary:
             Age        Salary
count   5.00000      5.000000
mean   31.80000  65400.000000
std     5.80517  12074.767078
min    25.00000  50000.000000
25%    29.00000  60000.000000
50%    30.00000  62000.000000
75%    35.00000  75000.000000
max    40.00000  80000.000000

Employees with Salary > 60000:
       Name  Age  Salary Department
2  Charlie   35   75000    Finance
3    David   40   80000         IT
4      Eva   29   62000         HR

Average Salary by Department:
 Department
Finance    75000.0
HR         56000.0
IT         70000.0
Name: Salary, dtype: float64

DataFrame with Bonus:
       Name  Age  Salary Department   Bonus
0    Alice   25   50000         HR  5000.0
1      Bob   30   60000         IT  6000.0
2  Charlie   35   75000    Finance  7500.0
3    David   40   80000         IT  8000.0
4      Eva   29   62000         HR  6200.0

Sorted by Age:
       Name  Age  Salary Department   Bonus
0    Alice   25   50000         HR  5000.0
4      Eva   29   62000         HR  6200.0
1      Bob   30   60000         IT  6000.0
2  Charlie   35   75000    Finance  7500.0
3    David   40   80000         IT  8000.0
```

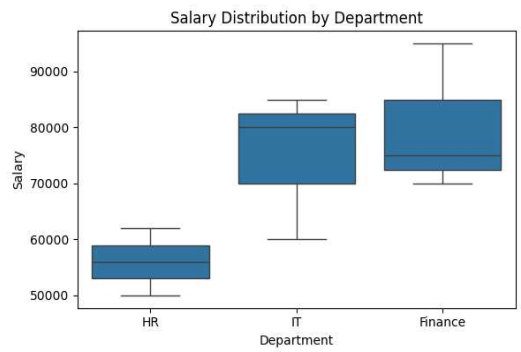
Week 4: Data Visualization with Matplotlib and Seaborn

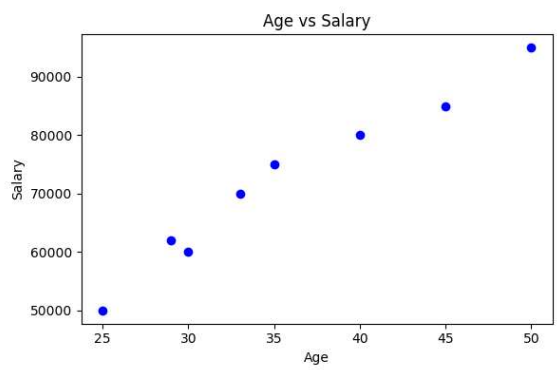
Hands-On: Create visualizations for dataset analysis.

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns


# Sample dataset

data = {

    "Age": [25, 30, 35, 40, 29, 50, 45, 33],

    "Salary": [50000, 60000, 75000, 80000, 62000, 95000, 85000, 70000],

    "Department": ["HR", "IT", "Finance", "IT", "HR", "Finance", "IT", "Finance"]

}

df = pd.DataFrame(data)


# 1. Histogram of Age

plt.figure(figsize=(6,4))

plt.hist(df["Age"], bins=5, edgecolor="black")

plt.title("Distribution of Age")

plt.xlabel("Age")

plt.ylabel("Count")

plt.show()


# 2. Scatter Plot (Age vs Salary)

plt.figure(figsize=(6,4))
```

```
plt.scatter(df["Age"], df["Salary"], c="blue")

plt.title("Age vs Salary")

plt.xlabel("Age")

plt.ylabel("Salary")

plt.show()


# 3. Boxplot of Salary by Department

plt.figure(figsize=(6,4))

sns.boxplot(x="Department", y="Salary", data=df)

plt.title("Salary Distribution by Department")

plt.show()
```







Client Project: Create a dashboard for visualizing relationships between features in a dataset (e.g., scatter plots, histograms).

```
import pandas as pd

import plotly.express as px

import plotly.io as pio


# Sample dataset

data = {

    "Age": [25, 30, 35, 40, 29, 50, 45, 33],

    "Salary": [50000, 60000, 75000, 80000, 62000, 95000, 85000, 70000],

    "Department": ["HR", "IT", "Finance", "IT", "HR", "Finance", "IT", "Finance"]

}

df = pd.DataFrame(data)


# Scatter plot (Age vs Salary with Department color)

fig1 = px.scatter(df, x="Age", y="Salary", color="Department", size="Salary",

                  title="Age vs Salary by Department")

fig1.show()


# Histogram of Salary

fig2 = px.histogram(df, x="Salary", nbins=6, color="Department",

                    title="Salary Distribution")

fig2.show()
```

```
# Boxplot Salary by Department

fig3 = px.box(df, x="Department", y="Salary", color="Department",

            title="Salary Distribution by Department")

fig3.show()
```