```
pip install pandas matplotlib openpyxl
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: openpyxl in /usr/local/lib/python3.11/dist-packages (3.1.5)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.11/dist-packages (from openpyxl) (2.0.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

```
import pandas as pd

# Sample data
data = {
    'Marks': [12, 10, 8, 7, 13, 5, 14, 9, 6, 11]
}

# Create DataFrame
df = pd.DataFrame(data)

# Save as Excel file in Colab's file system
df.to_excel("Case_Study_5_B.xlsx", index=False)

print("Excel file created!")
```

```
Excel file created!
```

```
from google.colab import files
files.download("Case_Study_5_B.xlsx")
```

```
from google.colab import files
uploaded = files.upload()
```

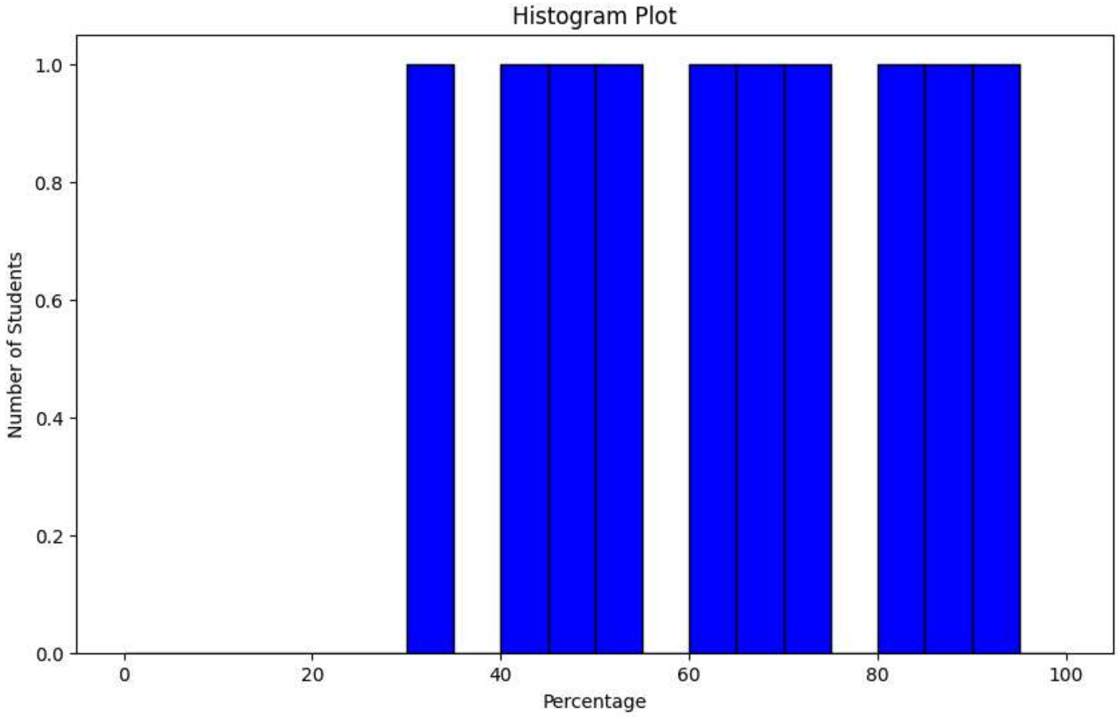
Choose Files Case_Study_5_B.xlsx

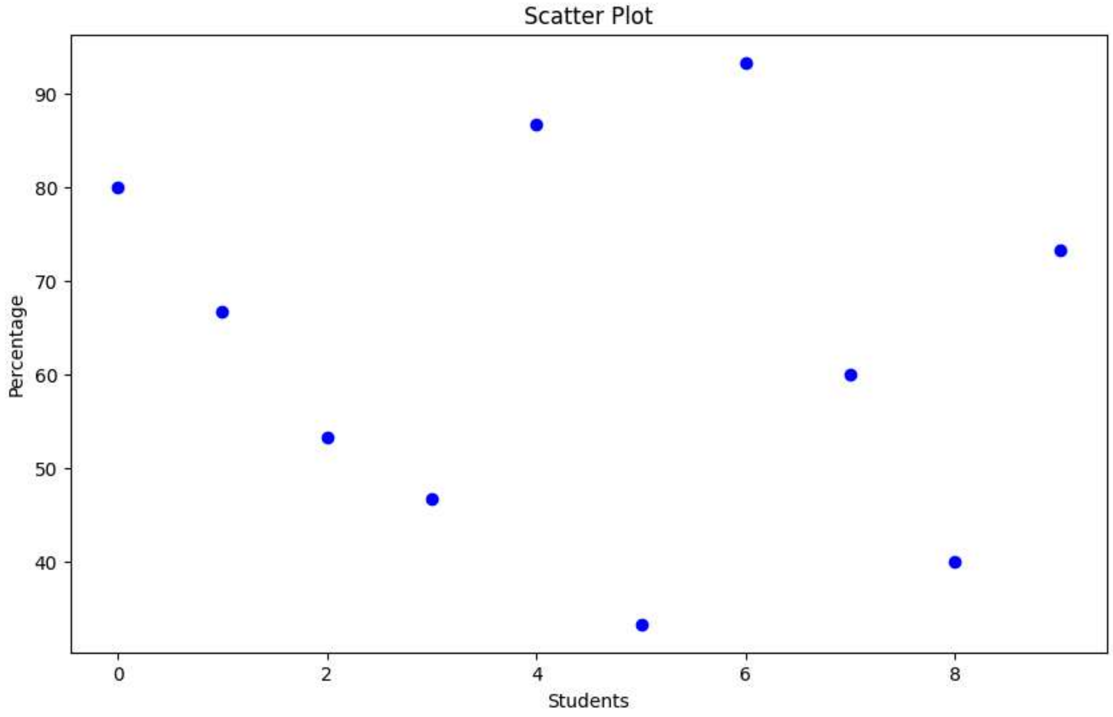
- **Case_Study_5_B.xlsx**(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 5009 bytes, last modified: 5/6/2025 - 100% done

```
Saving Case_Study_5_B.xlsx to Case_Study_5_B (1).xlsx
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Read the uploaded Excel file
data = pd.read_excel("Case_Study_5_B.xlsx")

# Calculate percentage (Max marks = 15)
data['Percentage'] = (data['Marks'] / 15) * 100

# Categorize students
high_achievers = data[data['Percentage'] > 75]
average_students = data[(data['Percentage'] >= 60) & (data['Percentage'] <= 75)]
low_achievers = data[data['Percentage'] < 60]

# Print categories
print("Students with >75%:\n", high_achievers)
print("\nStudents with 60%-75%:\n", average_students)
print("\nStudents with <60%:\n", low_achievers)

# Plot Histogram (blue bars, width=5, black border)
plt.figure(figsize=(10, 6))
plt.hist(data['Percentage'], bins=range(0, 101, 5), color='blue', edgecolor='black')
plt.xlabel("Percentage")
plt.ylabel("Number of Students")
plt.title("Histogram Plot")
plt.show()

# Plot Scatter Plot
plt.figure(figsize=(10, 6))
plt.scatter(data.index, data['Percentage'], color='blue')
plt.xlabel("Students")
plt.ylabel("Percentage")
plt.title("Scatter Plot")
plt.show()
```

```
Students with >75%:
    Marks  Percentage
0      12  80.000000
4      13  86.666667
6      14  93.333333

Students with 60%-75%:
    Marks  Percentage
1      10  66.666667
7       9  60.000000
9      11  73.333333

Students with <60%:
    Marks  Percentage
2       8  53.333333
3       7  46.666667
5       5  33.333333
8       6  40.000000
```





Start coding or generate with AI.