Name: Srishti Pandey
Class-Roll No.: TY9-40
Batch: B
PRN: 22UF17054CM100

Experiment No. 7

Title: Implementation of Data Discretization (any one) & Visualization (any one).

Aim: To implement Data Discretization & Visualization using Python codes.

Algorithm:

1. Input the continuous data.
2. Choose the number of bins or intervals for discretization.
3. Calculate the width of each bin using the formula: Bin Width=Max Value−Min ValueNumber of Bins\text{Bin Width} = \frac{\text{Max Value} - \text{Min Value}}{\text{Number of Bins}}Bin Width=Number of BinsMax Value−Min Value
4. Divide the data into equal-width intervals (bins) based on the calculated bin width.
5. Assign each data point to its respective bin.
6. Plot the histogram of the original continuous data and the discretized data for visualization.

```
# DWM EXP 7 Ty9-B-40
import numpy as np
import matplotlib.pyplot as plt

# Function to perform equi-depth binning
def equi_depth_binning(data, num_bins):
    sorted_data = sorted(data)
    bin_size = len(data) // num_bins
    bins = [sorted_data[i * bin_size: (i + 1) * bin_size] for i in range(num_bins - 1)]
    bins.append(sorted_data[(num_bins - 1) * bin_size:])
    return bins

# Replace each value in the bin with the bin mean
def bin_means(bins):
    return [[np.mean(bin)] * len(bin) for bin in bins]

# Replace each value in the bin with the closest boundary
def bin_boundaries(bins):
    new_bins = []
    for bin in bins:
        min_val, max_val = min(bin), max(bin)
        new_bins.append([min_val if x < (min_val + max_val) / 2 else max_val for x in bin])
    return new_bins

# Function to plot histograms
def plot_separate_histograms(data, bins, mean_bins, boundary_bins):
    # Flatten the bins for mean and boundary bins
    flattened_means = [val for bin in mean_bins for val in bin]
    flattened_boundaries = [val for bin in boundary_bins for val in bin]

    # Create subplots for three histograms with reduced height
    fig, axs = plt.subplots(3, 1, figsize=(10, 9), sharex=True)

    # Histogram for original data
    axs[0].hist(data, bins=len(bins), color='skyblue', edgecolor='black')
    axs[0].set_title("Histogram of Original Data")
    axs[0].set_ylabel("Frequency")

    # Histogram for bin means
    axs[1].hist(flattened_means, bins=len(bins), color='orange', edgecolor='black')
    axs[1].set_title("Histogram of Bin Means")
    axs[1].set_ylabel("Frequency")

    # Histogram for bin boundaries
    axs[2].hist(flattened_boundaries, bins=len(bins), color='green', edgecolor='black')
    axs[2].set_title("Histogram of Bin Boundaries")
    axs[2].set_xlabel("Value")
    axs[2].set_ylabel("Frequency")

    plt.tight_layout()

    # Bold visualization section title
    print("\n\033[1mVisualization Graph:\033[0m")
    plt.show()

# ====== Main Program ======

# User input
print("\033[1mProgram Input:\033[0m")
data = list(map(float, input("Enter the data values separated by spaces: ").split()))
num_bins = int(input("Enter the number of bins: "))

# Equi-depth bins
bins = equi_depth_binning(data, num_bins)

# Transformed bins
mean_bins = bin_means(bins)
boundary_bins = bin_boundaries(bins)

# Bold program output title
print("\n\033[1mProgram Output:\033[0m")
print("Original Data:", data)
print("Equi-depth Bins:", bins)
print("Bins with Means:", mean_bins)
print("Bins with Boundaries:", boundary_bins)
```

Program Input:

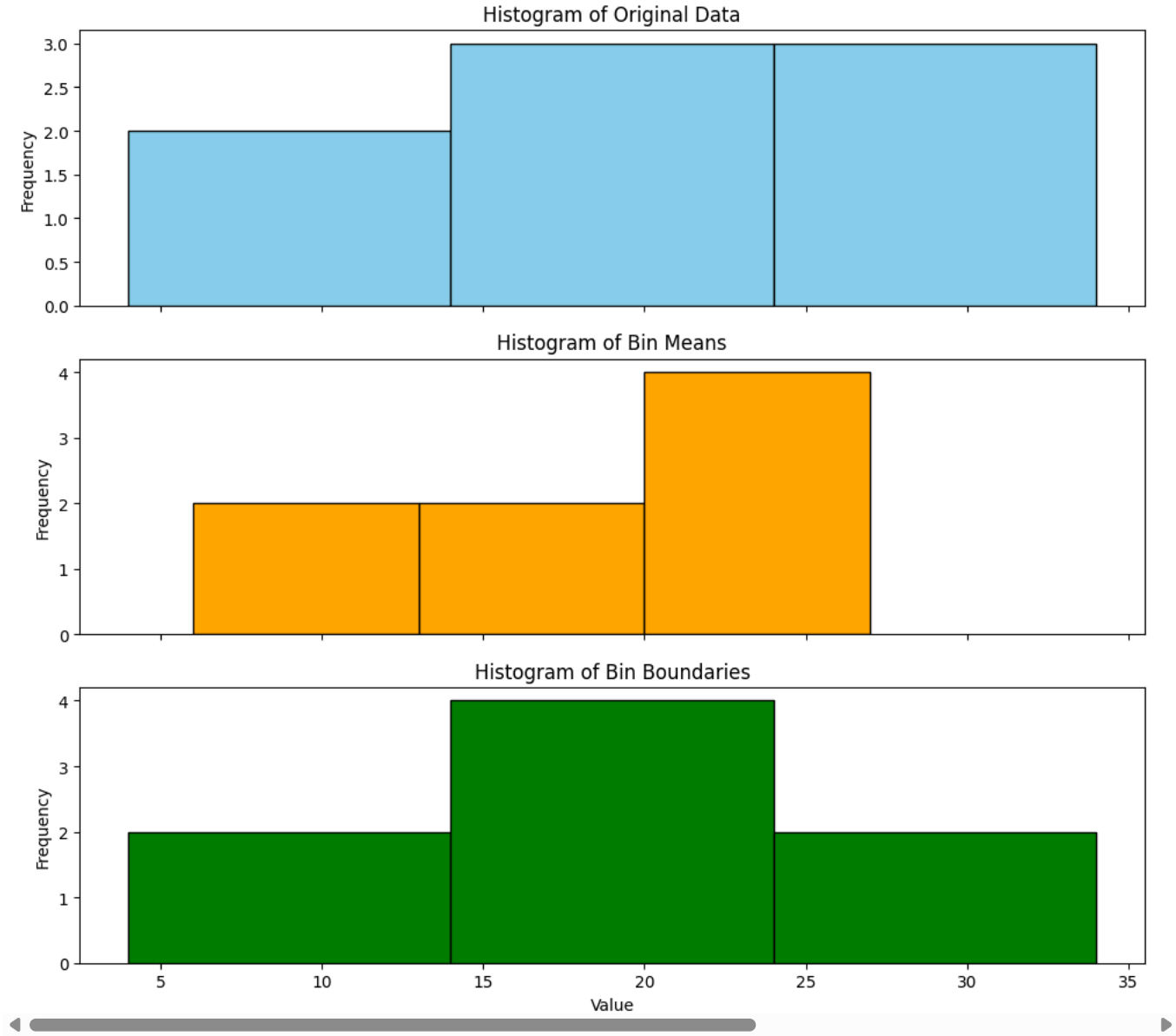
```
Enter the data values separated by spaces: 4 8 15 21 21 25 28 34
Enter the number of bins: 3
```

Program Output:

```
Original Data: [4.0, 8.0, 15.0, 21.0, 21.0, 25.0, 28.0, 34.0]
Equi-depth Bins: [[4.0, 8.0], [15.0, 21.0], [21.0, 25.0, 28.0, 34.0]]
Bins with Means: [[np.float64(6.0), np.float64(6.0)], [np.float64(18.0), np.float64(18.0)], [np.float64(27.0), np.float64(27.0), np
Bins with Boundaries: [[4.0, 8.0], [15.0, 21.0], [21.0, 21.0, 34.0, 34.0]]
```

```
# Plot separate histograms
plot_separate_histograms(data, bins, mean_bins, boundary_bins)
```

Visualization Graph:



Conclusion:- Equal Width Discretization is a simple method for converting continuous data into discrete categories by dividing the data range into equal intervals. It helps simplify complex data but may lose important details if the data is not uniformly distributed. The choice of bin width is crucial: too few bins may oversimplify the data, while too many can lead to overfitting. By comparing histograms of the original and discretized data, we can evaluate how well the process preserves key data features. This technique is useful for simplifying data but may require adjustments depending on the distribution and analysis goals.

Review Questions:

Q1.How does the process of Equal Width Discretization transform continuous data into discrete categories, and what role does the choice of the number of bins play in the outcome?

Ans: Equal Width Discretization transforms continuous data into discrete categories by:

Dividing the range of the data into a predefined number of equal-sized intervals or "bins".

Each data point is then assigned to a bin based on which interval it falls into.

Formula for bin width:

Bin Width = Max Value − Min Value Number of Bins Bin Width= Number of Bins Max Value−Min Value

Each bin is a range, e.g., if min = 0, max = 100, and bins = 5 → bin width = 20:

Bin 1: [0–20), Bin 2: [20–40), ..., Bin 5: [80–100]

Role of the number of bins:

Too few bins: May lead to overgeneralization, losing important data details.

Too many bins: May result in overfitting, noise amplification, or sparsely populated bins.

The number of bins directly affects the granularity and interpretability of the discretized data.

Q2: What insights can be derived from the histogram of the original continuous data, and how does it help in understanding the distribution of the data after discretization?

Ans:A histogram of the original continuous data shows:

Shape of the distribution (e.g., normal, skewed, bimodal)

Frequency of data points in each value range

Presence of outliers or sparse regions

Benefits for understanding discretization:

Helps choose appropriate number of bins (e.g., more bins in high-density areas)

Reveals whether Equal Width Discretization will result in uneven category populations (e.g., if the data is skewed)

Allows for comparison between original and discretized distributions, assessing how well the discretization preserves important patterns.

Q3.What are the potential advantages and limitations of using Equal Width Discretization for continuous data, and how can the choice of bin width affect the analysis of the data?

Ans:Advantages:

Simple to implement

Easy to interpret

Provides a uniform framework for discretization, useful in some machine learning preprocessing.

Limitations: Sensitive to outliers: Can create empty or overloaded bins

Doesn't account for data distribution: May result in bins with very different numbers of observations

Loss of important patterns if bin widths are not chosen appropriately

Effect of bin width: Too wide: Important data variations may be smoothed out.

Too narrow: May increase noise, reduce model generalization.

Ideally, bin width should balance resolution vs. robustness, possibly using data-driven methods or histograms to guide the choice.

Github Link: https://github.com/SrishtiPandey15/DWM-Batch-B-Exps