> Name: Suyash Katkam

> Class: TE9-B-25

> Subject: DWM

> 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 Data:**
>   - Take a list of continuous numerical values.
>   - Specify the number of bins for discretization.
> 2. **Sort the Data:**
>   - Arrange the input data in ascending order.
> 3. **Perform Equi-Depth Binning:**
>   - Divide the sorted data into bins with approximately equal number of elements.
> 4. **Apply Bin Transformations:**
>   - **Bin Mean**: Replace all values in each bin with the bin's mean.
>   - **Bin Boundaries**: Replace values with the nearest boundary (min or max of the bin).
> 5. **Display Results:**
>   - Print original data, equi-depth bins, bin means, and bin boundaries.
> 6. **Visualization:**
>   - Plot histograms of:
>   - Original Data
>   - Bin Means
>   - Bin Boundaries

> **Program Code:**

```
import numpy as np
import matplotlib.pyplot as plt

def equi_depth_binning(data, num_bins):
    data = sorted(data)
    size = len(data) // num_bins
    return [data[i*size:(i+1)*size] for i in range(num_bins-1)] + [data[(num_bins-1)*size:]]

def bin_means(bins):
    return [[np.mean(bin)] * len(bin) for bin in bins]

def bin_boundaries(bins):
    return [[min(bin) if x < (min(bin) + max(bin)) / 2 else max(bin) for x in bin] for bin in bins]

def plot_histograms(original, bins, means, boundaries):
    flat_means = [v for bin in means for v in bin]
    flat_bounds = [v for bin in boundaries for v in bin]
    titles = ["Original Data", "Bin Means", "Bin Boundaries"]
    colors = ['skyblue', 'orange', 'green']
    data_sets = [original, flat_means, flat_bounds]

    fig, axs = plt.subplots(3, 1, figsize=(6, 5), sharex=True)
    for ax, data, title, color in zip(axs, data_sets, titles, colors):
        ax.hist(data, bins=len(bins), color=color, edgecolor='black')
        ax.set_title(f"Histogram of {title}")
        if title != "Bin Means":
```

```
            ax.set_ylabel("Frequency")
        if title == "Bin Boundaries":
            ax.set_xlabel("Value")

    plt.tight_layout()
    print("\n\033[1mVisualization Graph:\033[0m")
    plt.show()

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

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: "))

bins = equi_depth_binning(data, num_bins)
mean_bins = bin_means(bins)
boundary_bins = bin_boundaries(bins)

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)

plot_histograms(data, bins, mean_bins, boundary_bins)
```

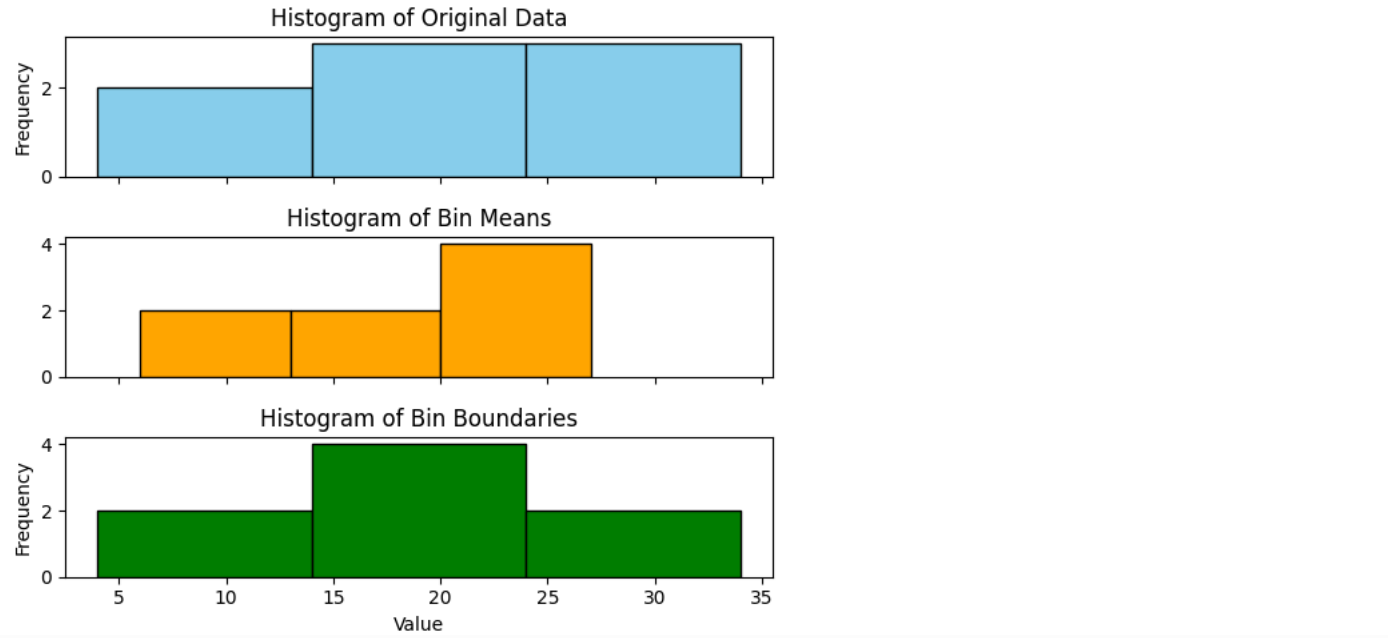
Program Input:

```
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.floa
Bins with Boundaries: [[4.0, 8.0], [15.0, 21.0], [21.0, 21.0, 34.0, 34.0]]
```

Visualization Graph:



Conclusion: Data discretization simplifies continuous data by grouping values into bins for easier pattern analysis. Equi-depth binning ensures each bin has an equal number of data points, providing balanced representation. This approach enables effective conversion of continuous variables into categorical ones for analysis. Visualizations of bin means and boundaries help in understanding the distribution and transformation of the data.

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:

1. **Transformation Process:**
    - Divide the entire range of continuous values into equal-width intervals.
    - Assign each data value to a bin according to its range.
2. **Role of Number of Bins:**
    - More bins lead to finer and detailed categorization.
    - Fewer bins result in broader groupings but may lose important patterns.

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:

1. **Insights from Histogram:**
    - Shows the overall distribution (e.g., skewed, symmetric, uniform).
    - Identifies clusters, gaps, and outliers in the data.
2. **Importance for Discretization:**
    - Helps decide the appropriate number of bins based on data spread.
    - Aids in preserving important features during discretization.

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:

1. **Advantages:**
    - Easy to understand and implement.
    - Good for uniformly distributed data.
2. **Limitations:**
    - May result in empty or crowded bins for skewed data.
    - Sensitive to extreme values (outliers).
3. **Effect of Bin Width:**
    - Smaller bin width increases granularity but may cause noise.
    - Larger bin width reduces details and may hide significant variations.

GitHub Link: https://github.com/suyashkatkam/DWM.git