Batch T4

Practical No. 2

Title of Assignment :

Normalization Methods

Student Name: Parshwa Herwade

Student PRN: 22510064

a. Generate 3 variables with 10,000 samples each:

1. **B:** Gaussian distribution with mean = 5, standard deviation = 2.
2. **I:** Power law distribution using scipy.stats.powerlaw.rvs with a = 0.3.
3. **H:** Geometric distribution with probability p = 0.005.

b. Compare the above variables in a single box plot.

c. Apply the following normalization methods:

1. **Max Normalization:** Divide each variable by its maximum value.
2. **Sum Normalization:** Divide each variable by the sum of its values.
3. **Z-score Normalization:** Convert each variable into a z-score using its respective mean and standard deviation.
4. **Percentile Transformation:** Convert each variable's values into percentiles.
5. **Median Matching:**
    - Compute the median of each variable.
    - Compute the mean of these medians (m1).
    - Generate a multiplier for each variable so that its median becomes m1.
6. **Quantile Normalization:** Use an off-the-shelf library function to perform quantile normalization.

d. Visualization and Comparison:

1. **Histogram Comparison:** Compare the original distribution with its normalized version in a single histogram for each method.
2. **Box Plot Comparison:** Compare all normalized variables in a single box plot.

CODE:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import powerlaw, geom, zscore, rankdata
from sklearn.preprocessing import quantile_transform

# Function to generate random data
def generate_data(size=10000):
    B = np.random.normal(5, 2, size)
    I = powerlaw.rvs(0.3, size=size)
    H = geom.rvs(0.005, size=size)
    return B, I, H
# Function to plot boxplots with custom colors
def plot_boxplot(data, labels, title):
    plt.figure(figsize=(10, 5))
colors = ['#FFD700', '#90EE90', '#D8BFD8']
    sns.boxplot(data=data, palette=colors, linewidth=2, width=0.6)
    plt.xticks(range(len(labels)), labels, fontsize=12, fontweight='bold')
    plt.title(title, fontsize=14, fontweight='bold', color='black')
    plt.grid(True, linestyle='-.', alpha=0.8)
    plt.show()

# Function to plot histograms
def plot_histogram(original, transformed, title):
    plt.figure(figsize=(10, 5))
    sns.histplot(original, bins=50, color='purple', kde=True, label='Original', alpha=0.5)
```

```
    sns.histplot(transformed, bins=50, color='green', kde=True, label='Transformed', alpha=0.5)
    plt.title(title, fontsize=14, fontweight='bold', color='black')
    plt.legend()
    plt.grid(True, linestyle='-.', alpha=0.8)
    plt.show()

# Function to normalize data
def normalize_data(B, I, H):
    normalizations = {
        "Max": lambda x: x / x.max(),
        "Sum": lambda x: x / x.sum(),
        "Z-Score": zscore,
        "Percentile": lambda x: rankdata(x) / len(x),
        "Median Matching": lambda x, m1: x * (m1 / np.median(x)),
        "Quantile": lambda x: quantile_transform(x.reshape(-1, 1), axis=0, copy=True).flatten()
    }

    medians = np.median([B, I, H], axis=1)
    m1 = np.mean(medians)
    transformed_data = {}
    for name, func in normalizations.items():
        if name == "Median Matching":
            transformed_data[name] = (func(B, m1), func(I, m1), func(H, m1))
        else:
            transformed_data[name] = (func(B), func(I), func(H))

    return transformed_data
```

```
# Generate original data
B, I, H = generate_data()


# Plot the original data distributions using boxplots
plot_boxplot([B, I, H], ['B (Gaussian)', 'I (Power Law)', 'H (Geometric)'], 'Original Variable Distribution')


# Normalize the data using various methods
transformed_data = normalize_data(B, I, H)


# Plot histograms and boxplots for each transformed data
for name, (B_new, I_new, H_new) in transformed_data.items():
    plot_histogram(B, B_new, 'B - ' + name + ' Normalization')
    plot_histogram(I, I_new, 'I - ' + name + ' Normalization')
    plot_histogram(H, H_new, 'H - ' + name + ' Normalization')
    plot_boxplot([B_new, I_new, H_new], ['B', 'I', 'H'], 'Box Plot - ' + name + ' Normalization')
```

OBSERVATIONS:

Gaussian Distribution (B - Normal Distribution):

Also called a bell curve, it is symmetrical around the mean.The probability of values further from the mean decreases exponentially.Many natural phenomena (e.g., heights, IQ scores) follow this distribution.

Power Law Distribution (I):

This distribution has a heavy tail, meaning a few extreme values dominate.Common in wealth distribution, internet traffic, and city populations.Unlike Gaussian, the mean and variance may be infinite in extreme cases.

Geometric Distribution (H):

Models the number of trials before the first success in repeated experiments.Highly right-skewed, meaning many small values and few large values.Used in modeling failure rates, waiting times, and biological mutations.Key Normalization Concepts

Quantiles:

Divide data into equal-sized groups (e.g., quartiles, percentiles).

Used in quantile normalization to adjust distributions to a common scale.

Z-Score:

Measures how far a value is from the mean in units of standard deviation.

Standardizes data to have mean = 0 and standard deviation = 1.

Ranking & Percentiles:

Ranks each value compared to the dataset (e.g., 90th percentile = top 10%).

Removes absolute values and only considers relative positioning.

1. Max Normalization (x / x.max())

Boxplot Observations:

All values are scaled between 0 and 1.

B (Gaussian): Shape remains, but spread compresses.

I (Power Law): Still dominated by large values.

H (Geometric): Right skew remains.

Histogram Observations:

Shapes remain the same, only rescaled. Sensitive to extreme outliers, as one large value affects all others.

2. Sum Normalization (x / x.sum())

Boxplot Observations:

All values shrink significantly.

Relative proportions stay intact.

Histogram Observations:

Shapes are preserved but on a much smaller scale.Works well for proportional comparisons (e.g., probability distributions).Sensitive to dataset size—different sizes lead to different scaling.

3. Z-Score Normalization ((x - mean) / std)

Boxplot Observations:

All distributions are centered at 0.

Outliers become more pronounced.

Histogram Observations:

B becomes a standard normal distribution.

I and H retain skew, but outliers become extreme.Handles different scales well, making variables comparable.Assumes Gaussian-like distribution, so skewed data may still be problematic.

4. Percentile Normalization (rank(x) / len(x))

Boxplot Observations:

All distributions become uniform.

Histogram Observations:

B loses bell shape, becoming uniform.

I and H spread evenly.Removes influence of outliers but loses original distribution properties.

5. Median Adjusted Normalization (x * (mean(medians) / median(x)))

Boxplot Observations:

Distributions align in median.

Histogram Observations:

Rescales while keeping overall shape intact.More robust than mean-based scaling.

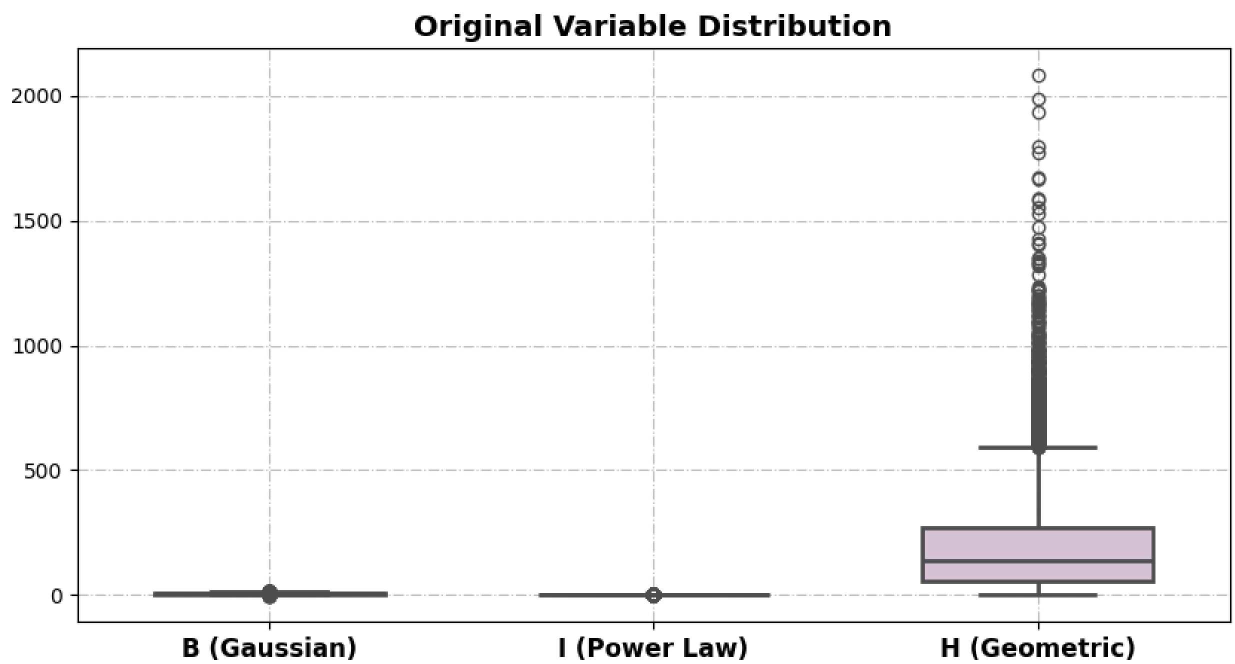
6. Quantile Normalization

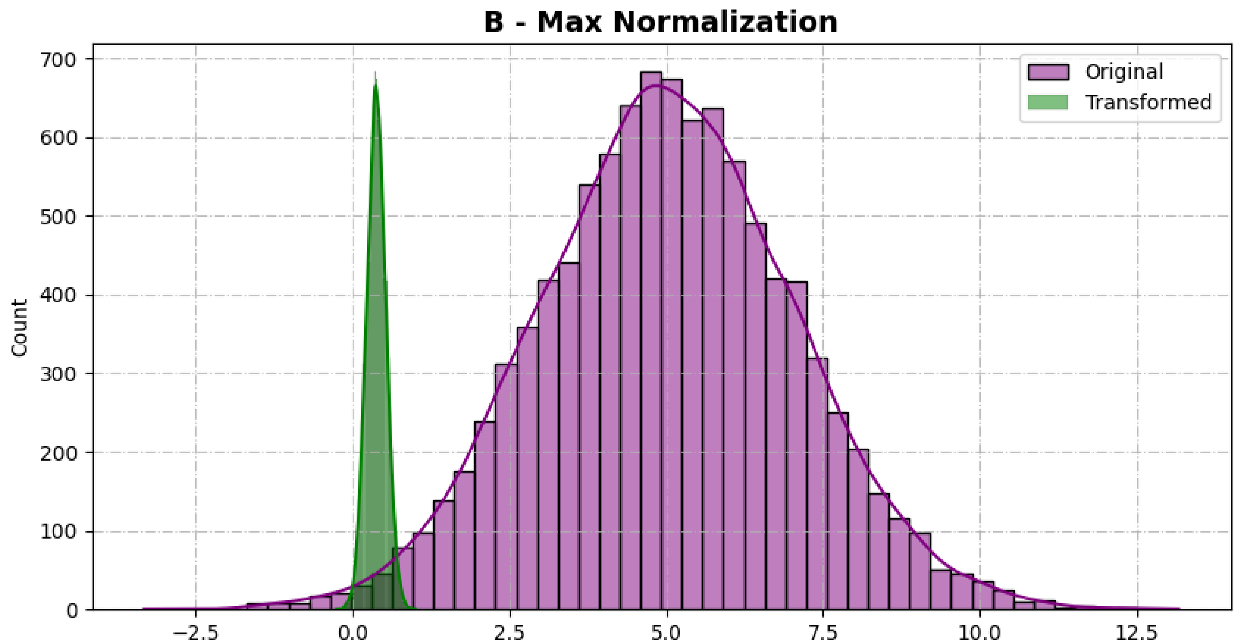
Boxplot Observations:

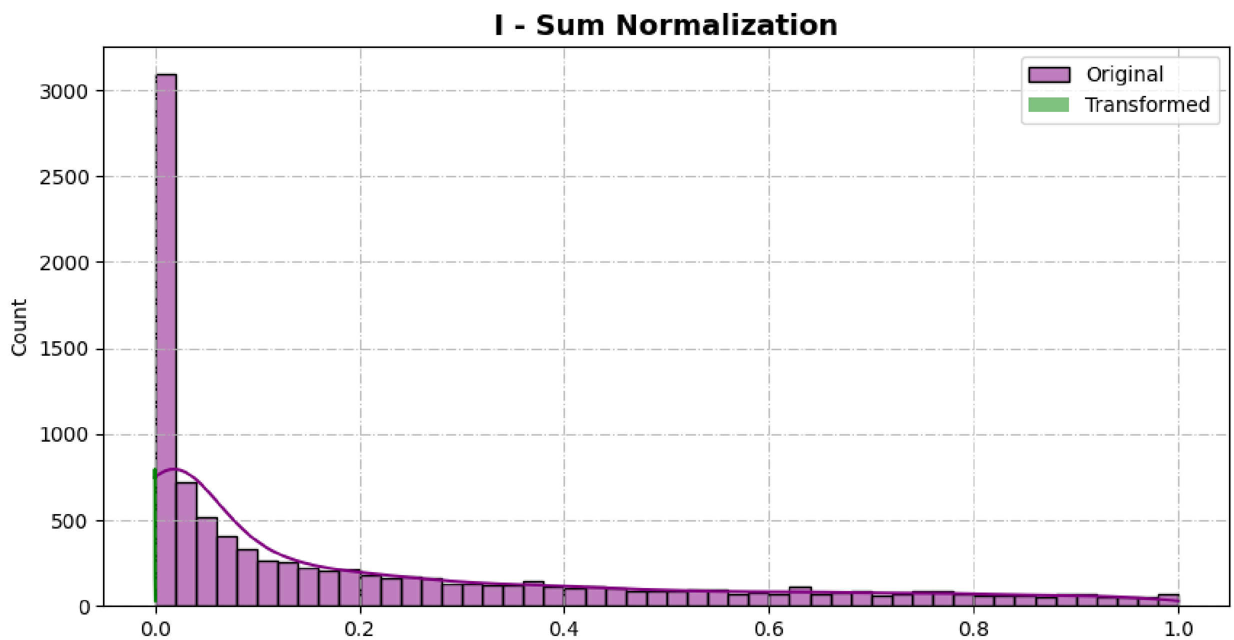
All distributions look similar.

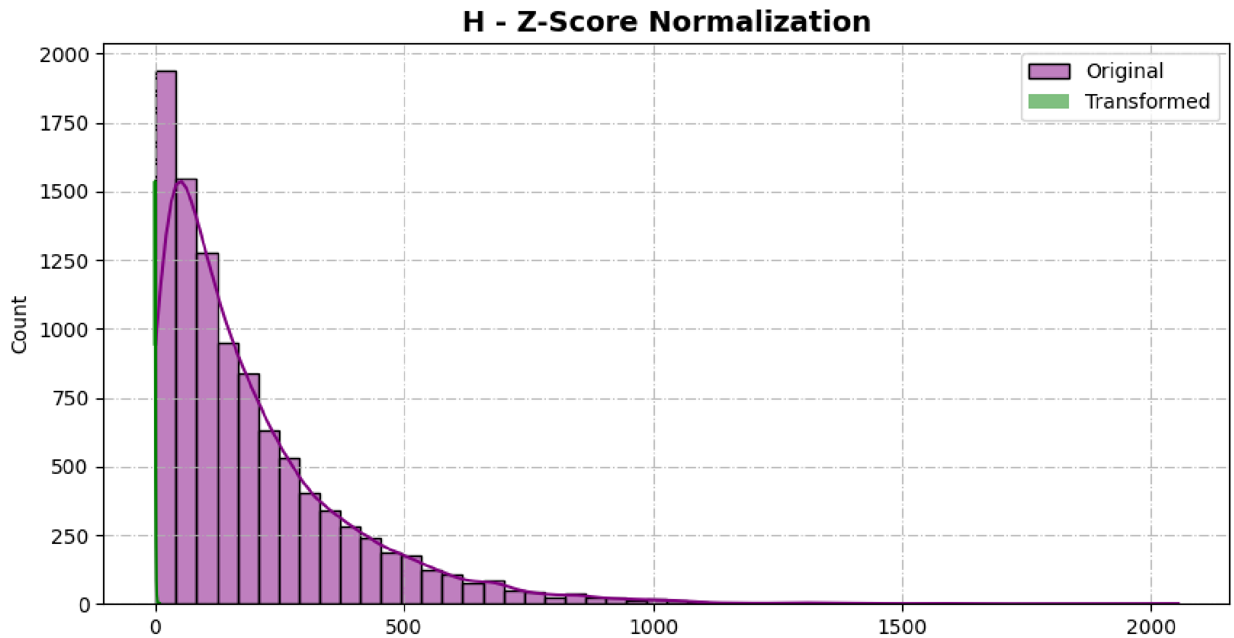
Histogram Observations:F orces distributions into the same shape.

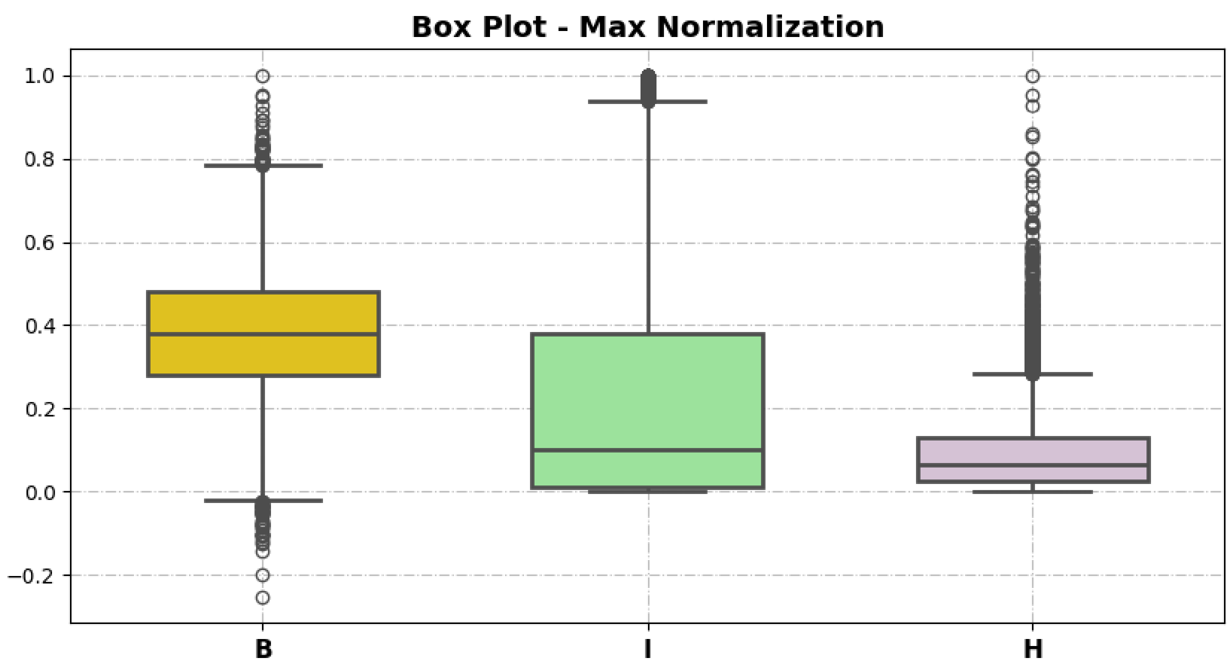
Best for making different datasets comparable.Loses original data structure.

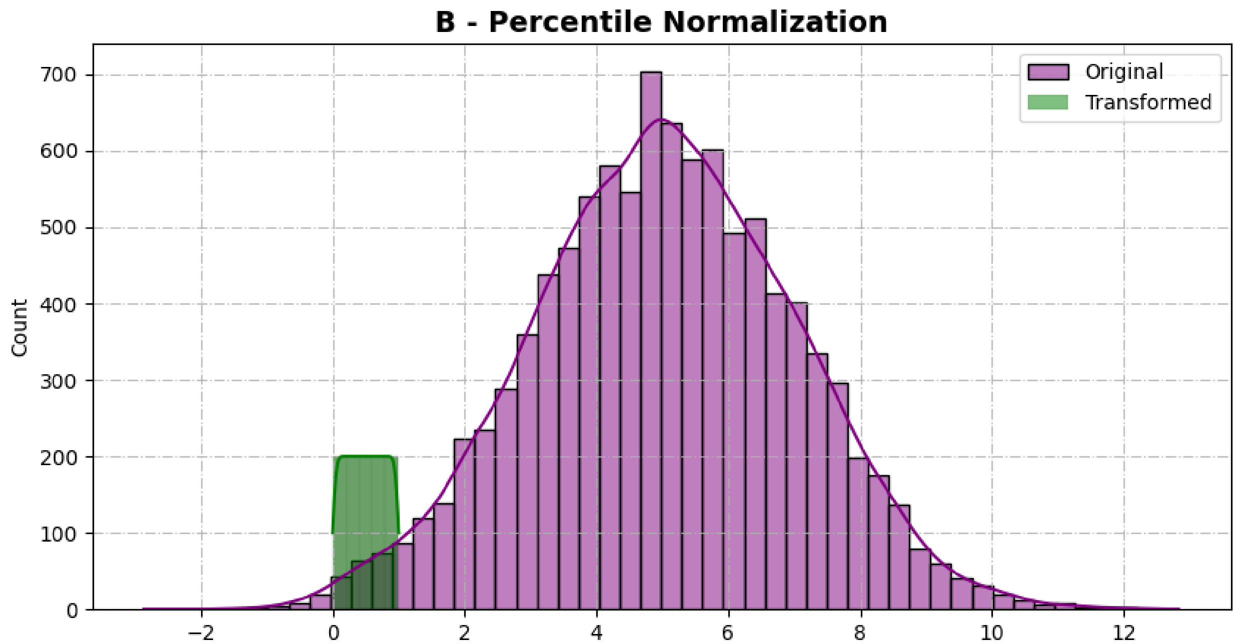


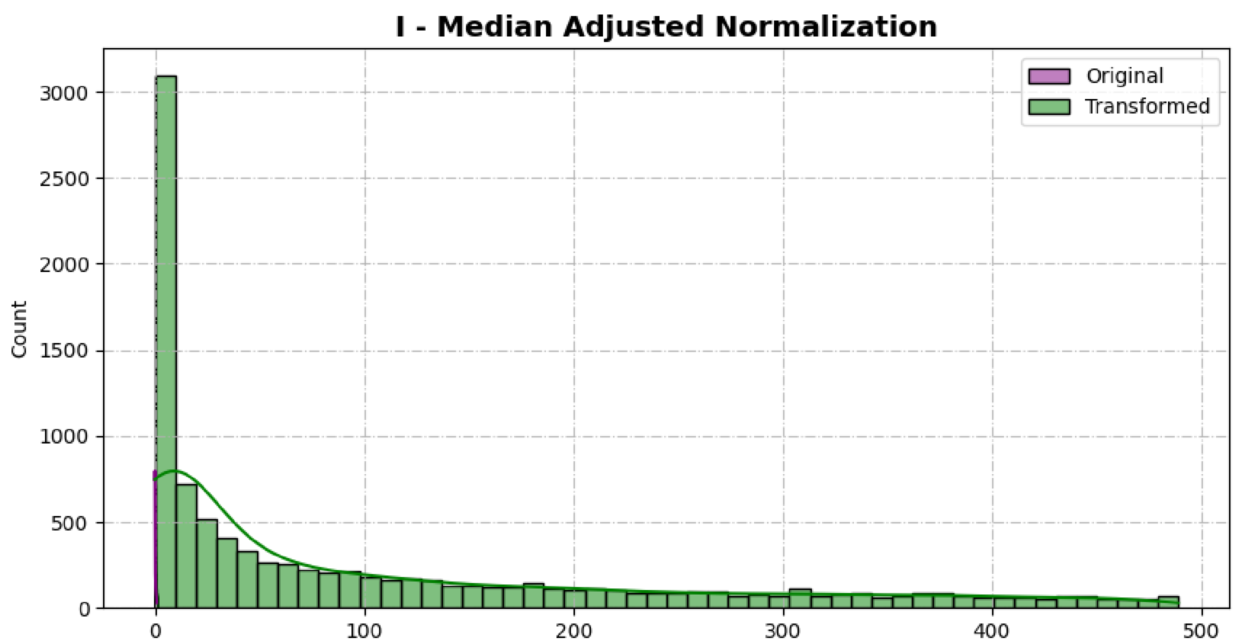

B - Max Normalization
Original
Transformed
Count
700
600
500
400
300
200
100
0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5

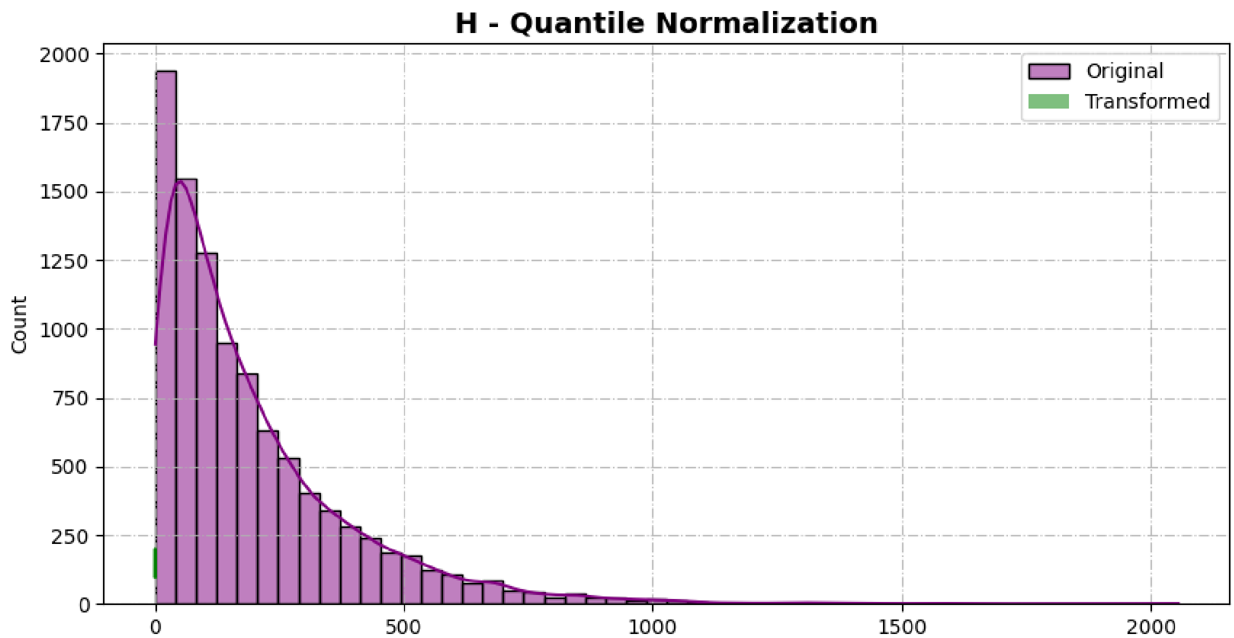

I - Sum Normalization
Original
Transformed
Count
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0


H - Z-Score Normalization
Original
Transformed
Count
2000
1750
1500
1250
1000
750
500
250
0
0
500
1000
1500
2000


Box Plot - Max Normalization
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
B
I
H


B - Percentile Normalization
Original
Transformed
Count
700
600
500
400
300
200
100
0
−2
0
2
4
6
8
10
12


I - Median Adjusted Normalization
Original
Transformed
Count
3000
2500
2000
1500
1000
500
0
0
100
200
300
400
500



Similarly others….