**Assignment 2**

**Machine Learning**

Name: Jasmine I. Sayyad

PRN: 22510033

Batch: T-2

a. Generate 3 variables, 10000 samples each

i. B: Gaussian mean 5 sd 2

ii. I: Power law

iii. H: Geometric p= 0.005 use scipy.stats.powerlaw.rvs with a= 0.3

**Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy.stats import powerlaw

from sklearn.preprocessing import quantile\_transform

gaussian\_mean = 5

gaussian\_sd = 2

totalsize = 10000

np.random.seed(69)

#Gaussian distribution

B = np.random.normal(gaussian\_mean ,gaussian\_sd , size = totalsize)

#Power Law distribution

a = 0.3

I = powerlaw.rvs(a, size = totalsize)

#Geometric Distribution

p = 0.005

H = np.random.geometric( p, size = totalsize)

plt.boxplot([B,I,H], tick\_labels=["Gaussian","powerlaw","geometric"])

plt.title("Box-plot distribution")

plt.show()

# Divide each variable by max

def max\_normalization(B, I , H):

    Gaussian\_max = B/np.max(B)

    powerlaw\_max = I/np.max(I)

    geometric\_max = H/np.max(H)

    return Gaussian\_max, powerlaw\_max, geometric\_max

Gaussian\_max, powerlaw\_max, geometric\_max = max\_normalization(B, I, H)

plt.boxplot([Gaussian\_max, powerlaw\_max, geometric\_max], tick\_labels=["Gaussian","powerlaw","geometric"])

plt.title("Box-plot distribution max\_normalization")

plt.show()

# Divide each variable by sum of its values

def sum\_normalization(B , I, H):

    Gaussian\_sum = B/np.sum(B)

    powerlaw\_sum = I/np.sum(I)

    geometric\_sum = H/np.sum(H)

    return Gaussian\_sum, powerlaw\_sum, geometric\_sum

Gaussian\_sum, powerlaw\_sum, geometric\_sum = sum\_normalization(B, I, H)

plt.boxplot([Gaussian\_sum, powerlaw\_sum, geometric\_sum], tick\_labels=["Gaussian","powerlaw","geometric"])

plt.title("Box-plot distribution sum\_normalization")

plt.show()

# Convert each variable into z score using respective mean and sd

def zscore\_normalization(B, I, H):

    Gaussian\_zscore = (B-B.mean())/B.std()

    powerlaw\_zscore = (I-I.mean())/I.std()

    geometric\_zscore = (H-H.mean())/H.std()

    return Gaussian\_zscore, powerlaw\_zscore, geometric\_zscore

Gaussian\_zscore, powerlaw\_zscore, geometric\_zscore = zscore\_normalization(B, I,H)

plt.boxplot([Gaussian\_zscore, powerlaw\_zscore, geometric\_zscore], tick\_labels=["Gaussian","powerlaw","geometric"])

plt.title("Box-plot of z-score Normalized Variables")

plt.show()

# For each variable , convert the values in percentiles

def percentile(array):

    np.sort(array)

    curr\_value = np.arange(1,len(array)+1)

    percentile\_value = (curr\_value/totalsize) \* 100

    return percentile\_value

def percentile\_normalization(B, I, H):

    gaussian\_percentile = percentile(B)

    powerlaw\_percentile = percentile(I)

    geometric\_percentile = percentile(H)

    return gaussian\_percentile, powerlaw\_percentile, geometric\_percentile

gaussian\_percentile, powerlaw\_percentile, geometric\_percentile = percentile\_normalization(B, I, H)

plt.boxplot([gaussian\_percentile, powerlaw\_percentile, geometric\_percentile], tick\_labels=["Gaussian","powerlaw","geometric"])

plt.title("Box-plot of percentile normalized variables")

plt.show()

# Make median of all variables same

def same\_median(B, I , H):

    gaussian\_median = np.median(B)

    powerlaw\_median = np.median(I)

    geometric\_median = np.median(H)

    mean\_of\_median = (gaussian\_median + powerlaw\_median + geometric\_median)/3

    multiplier1 = mean\_of\_median/gaussian\_median

    multiplier2 = mean\_of\_median/powerlaw\_median

    multiplier3 = mean\_of\_median/geometric\_median

    new\_B = multiplier1 \* B

    new\_I = multiplier2 \* I

    new\_H = multiplier3 \* H

    return new\_B, new\_I, new\_H

new\_B, new\_I, new\_H = same\_median(B, I, H)

plt.boxplot([new\_B, new\_I, new\_H], tick\_labels=["Gaussian","powerlaw","geometric"])

plt.title("Box-plot after making median same")

plt.show()

#  Quantile normalize the data using off the shelf library function

df = pd.DataFrame({

    "Gaussian": B,

    "Powerlaw": I,

    "Geometric": H

})

df\_quantile\_normalized = pd.DataFrame(

    quantile\_transform(df, output\_distribution='normal', copy=True),

    columns=df.columns

)

plt.boxplot([df\_quantile\_normalized["Gaussian"],

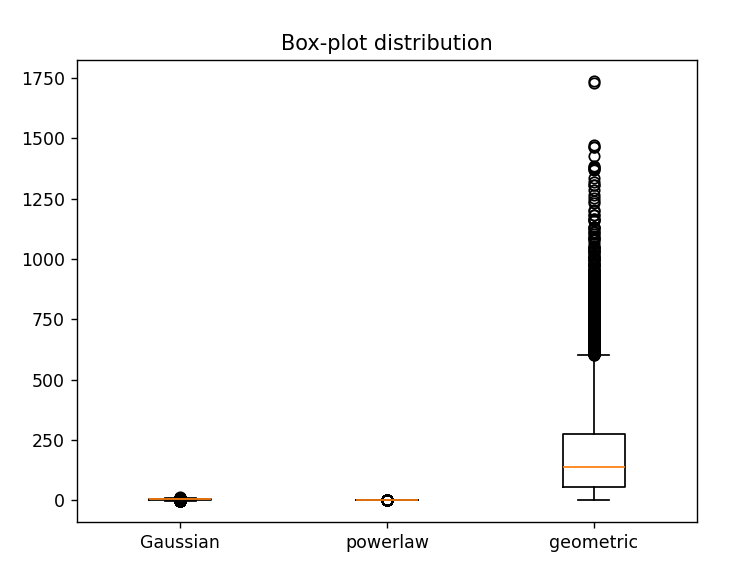
             df\_quantile\_normalized["Powerlaw"],

             df\_quantile\_normalized["Geometric"]],

            labels=df.columns)

plt.title("Box-plot of Quantile Normalized Variables (Using Off-the-Shelf Library)")

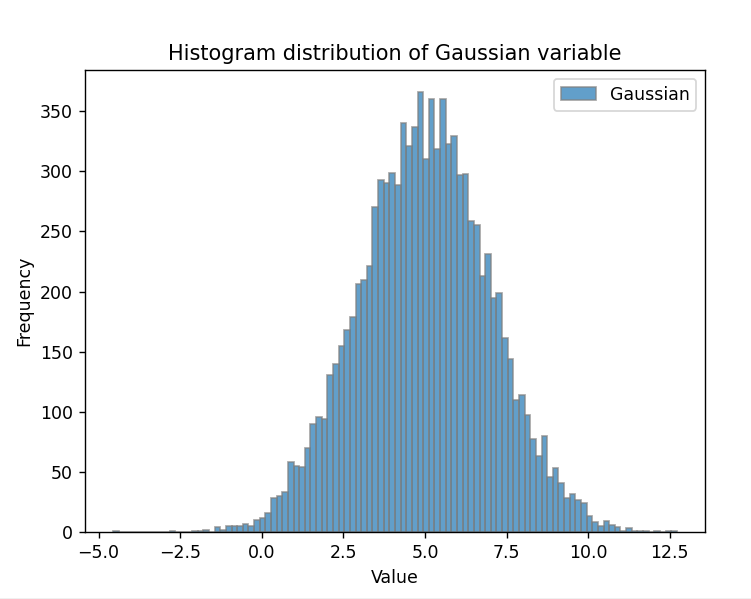
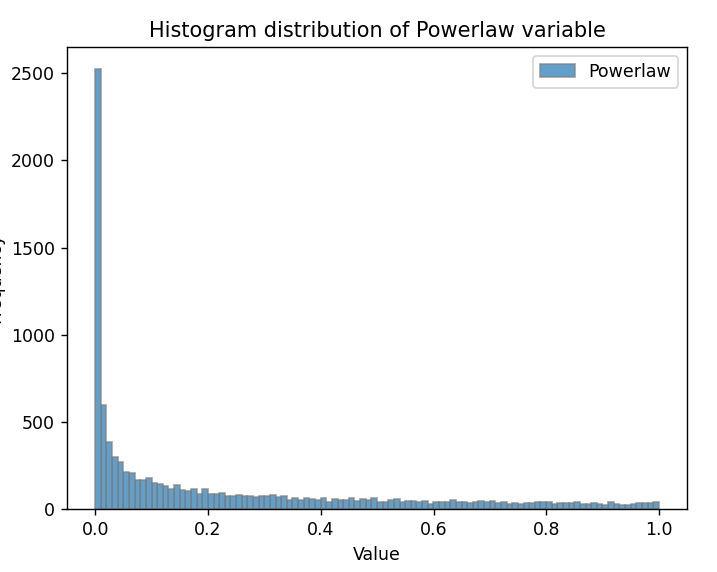
plt.show()

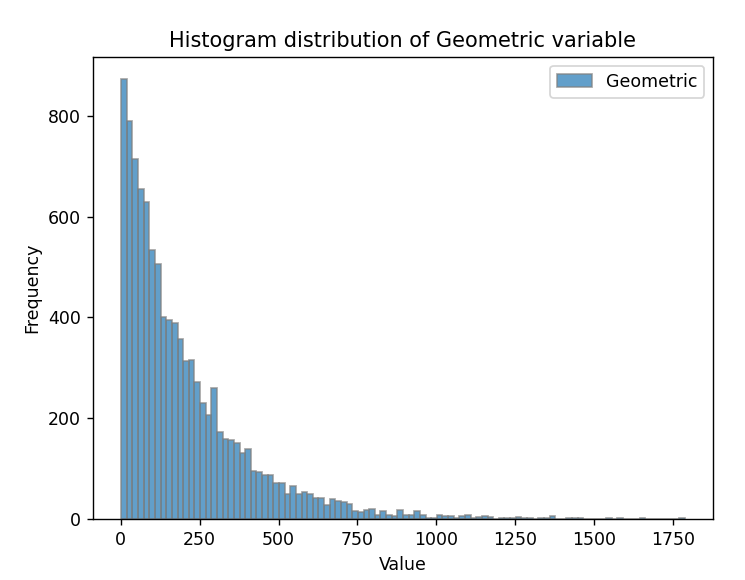


Observation:

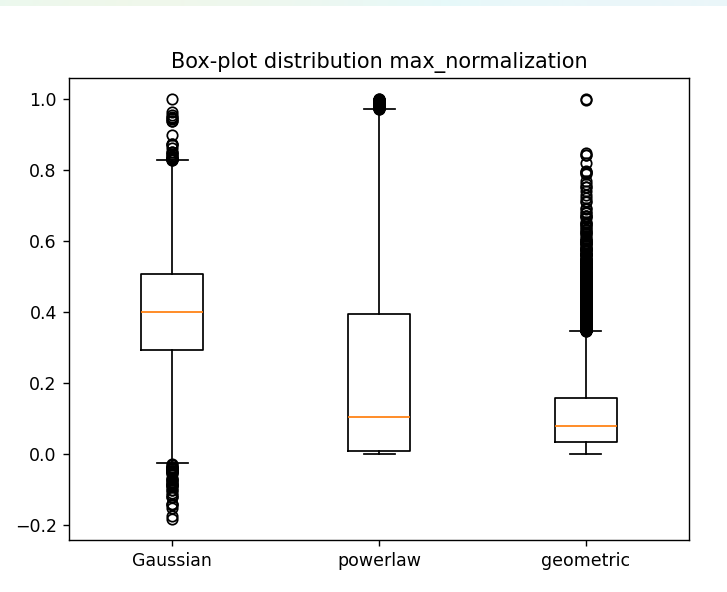
1) In Gaussian distribution, the median will approximately lie close to 5 since mean = 5 and as SD = 2, 50% of data will lie inside the IQR (3 to 7 here), very few outliers

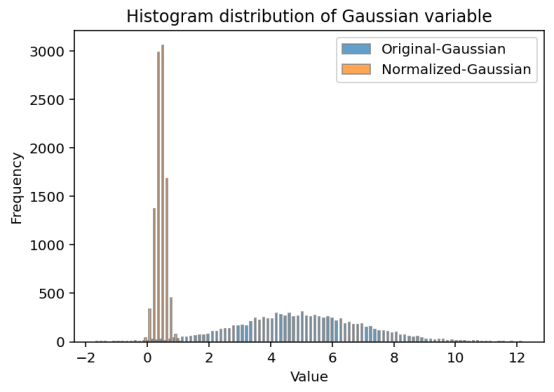
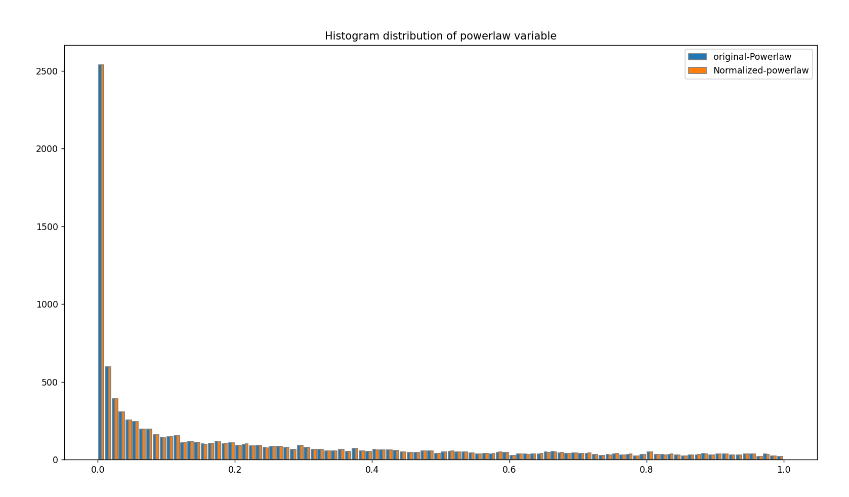
2) In power law distribution, the median will be near to 0 because this distribution generates more smaller values than larger one. Most of the outliers will be lying beyond the upper whisker, IQR is very small and near to 0

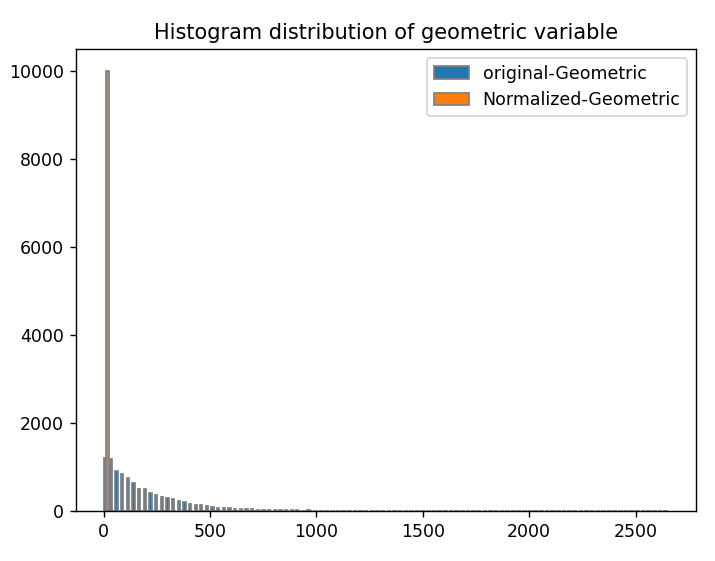
3) As p= 0.005 which is very low, the no. of trials required for first success is large may be around 200. Therefore, median is high, also most of the outliers lie at the upper whisker



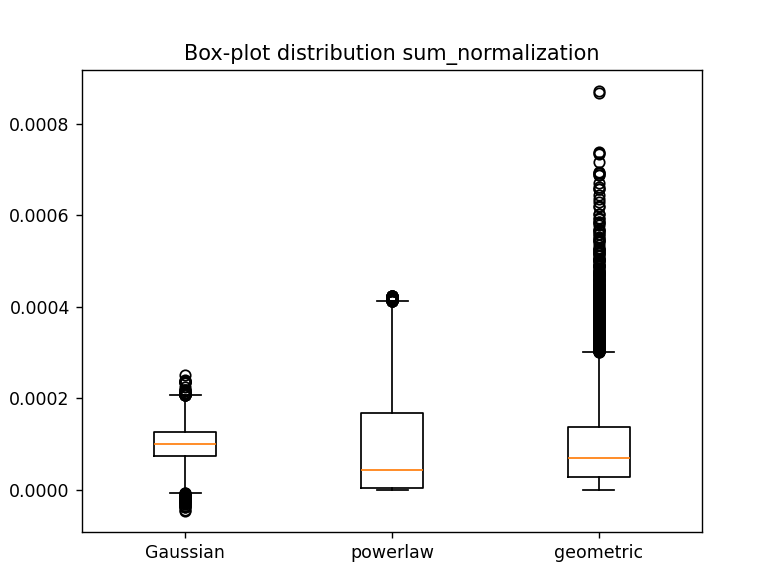
* Min-Max Normalization:



1. Normalizes the datasets to fall in range 0 to 1
2. In case of gaussian the data will be compresses but symmetric. Minimal spread and no significant outliers.

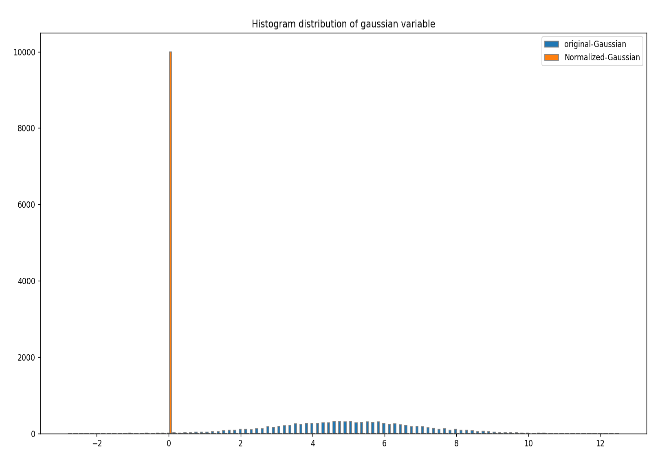
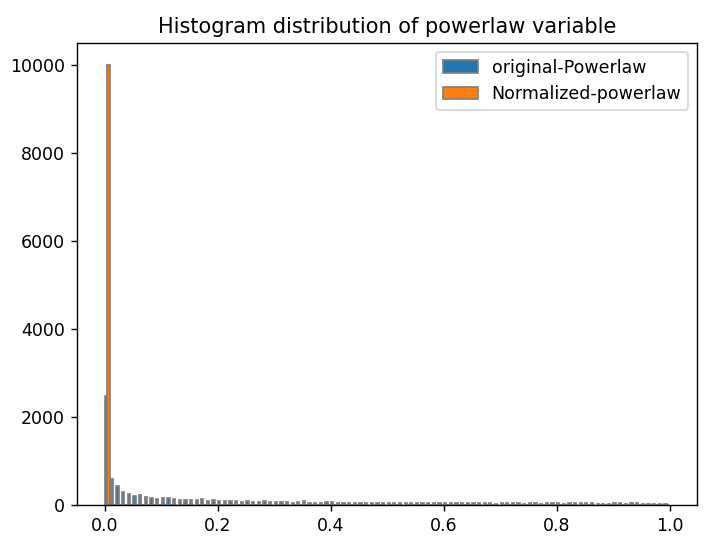


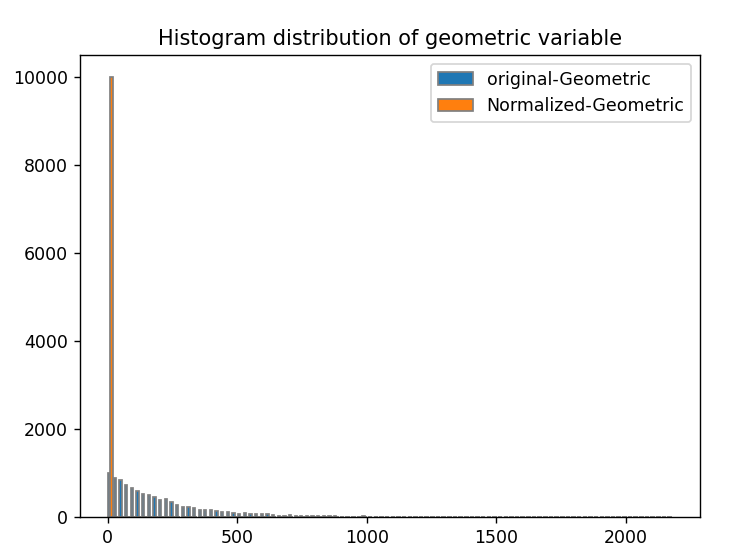
* Sum Normalization



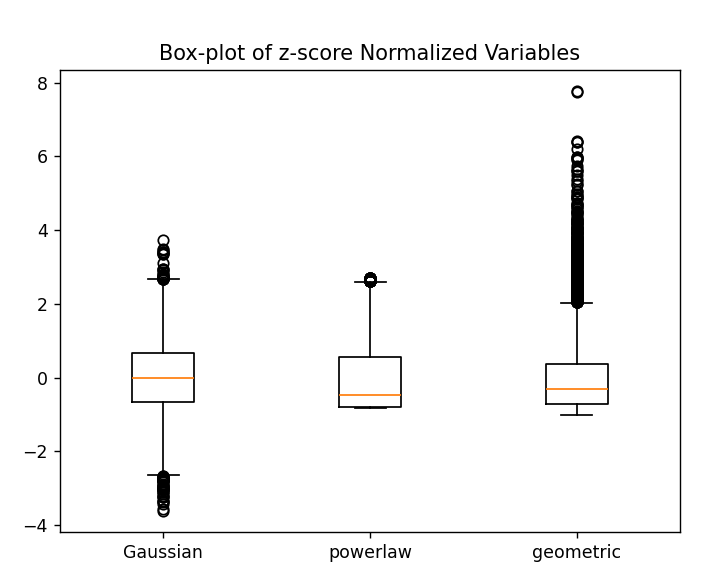
**Gaussian Distribution**: Symmetric with a balanced spread around the median, few outliers, and moderate variability.

**Power-law Distribution**: Skewed with a heavy upper tail, large variability, and several outliers, reflecting rare extreme values.

**Geometric Distribution**: Concentrated at lower values with a narrow spread and many outliers,



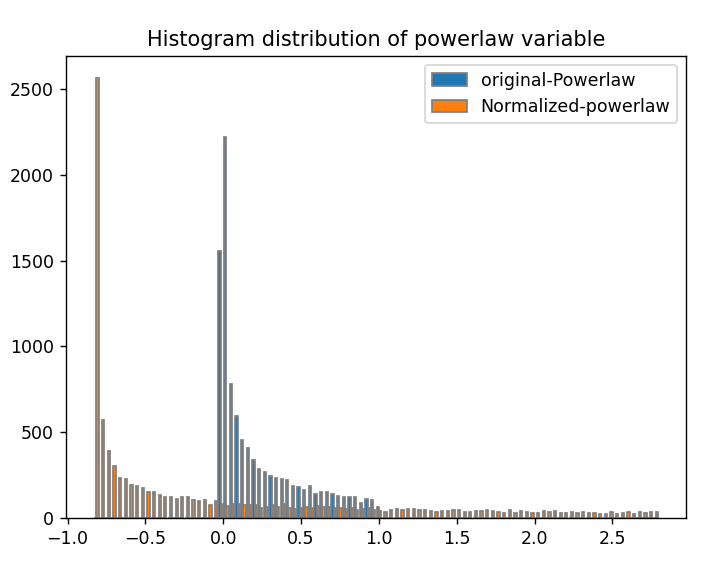
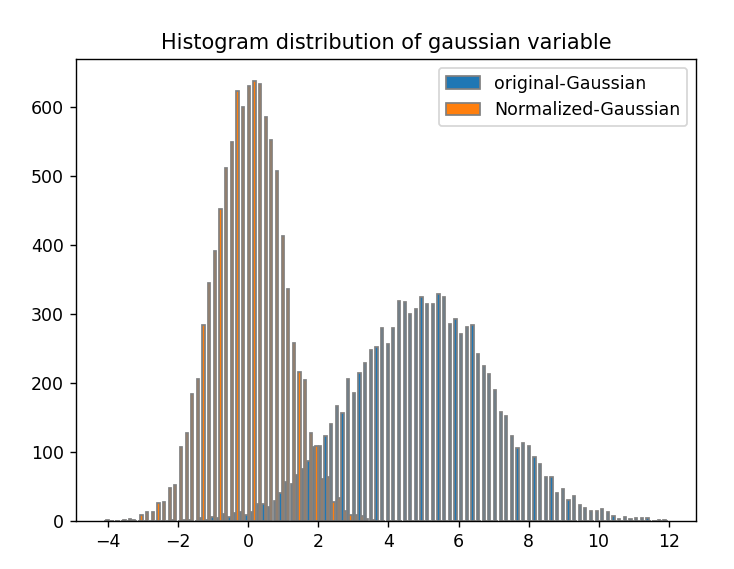
* Z-score normalization

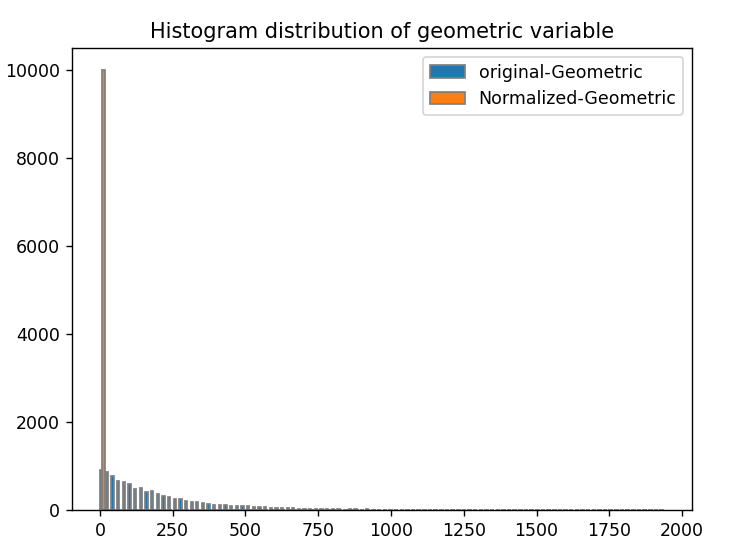


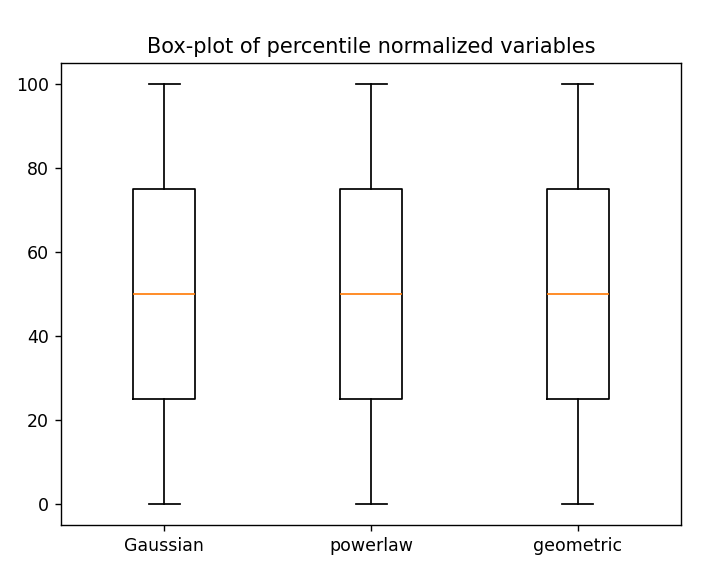
**Gaussian:** The data will look balanced and symmetrical on the boxplot, with the middle value (mean) at 0. The spread of the data will be consistent and standardized.

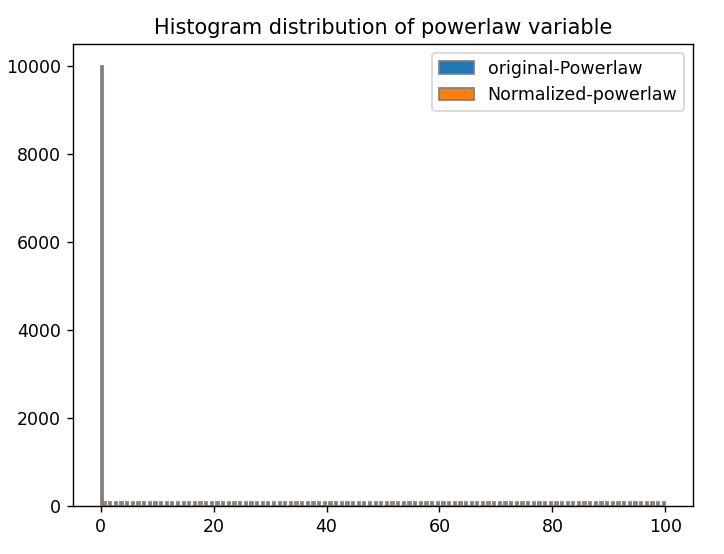
**Power-law Distribution:** You might see some extreme values (outliers) that are still present, but they might not be as far from the main part of the data. The boxplot will still show a heavy skew due to the long tail of extreme values.

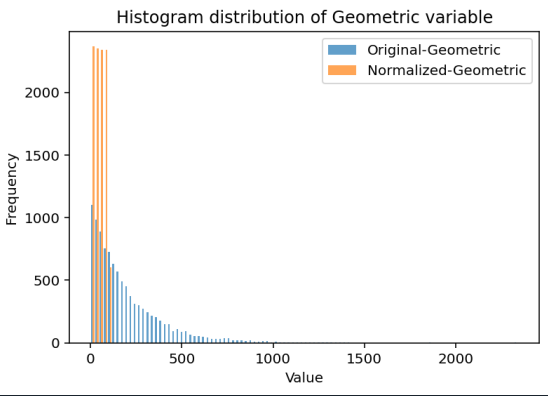
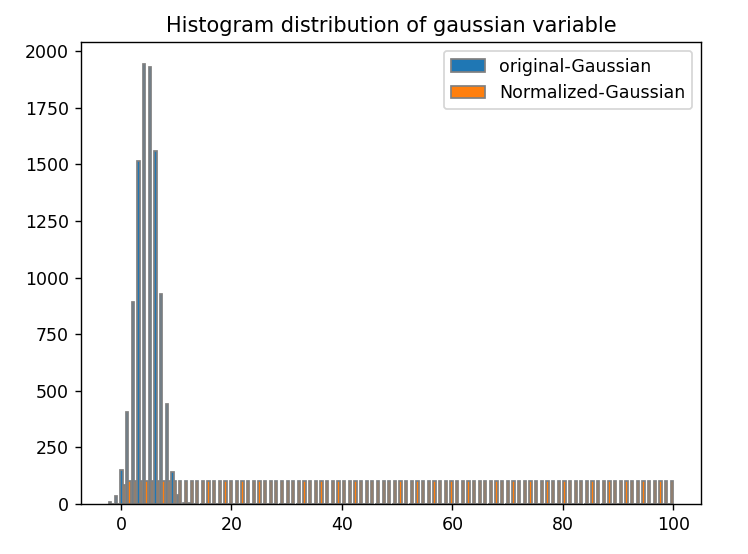
**Geometric Distribution:** The data points will be more evenly spread out near 0, but there will still be more data concentrated around the lower values.

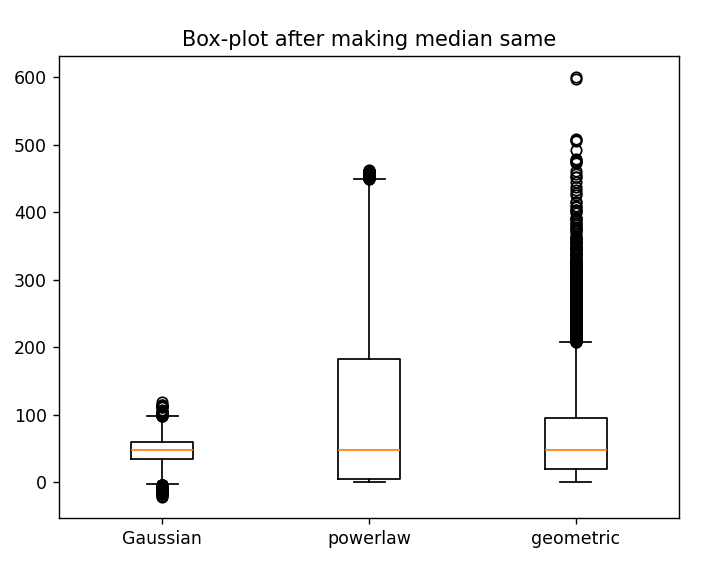






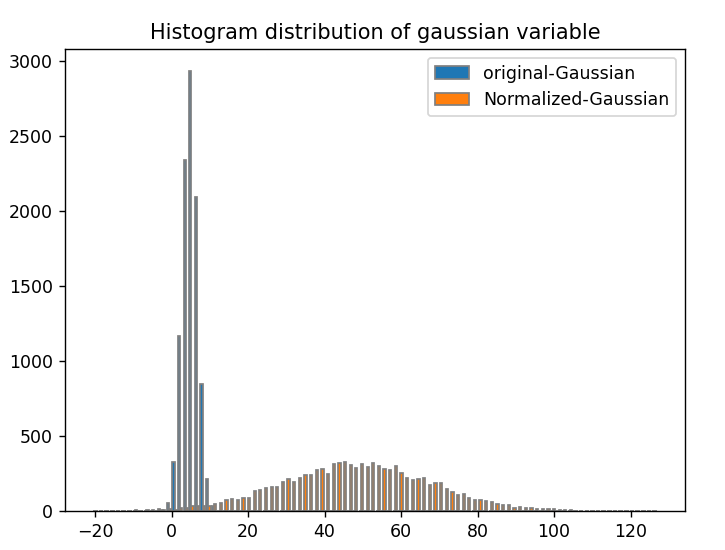
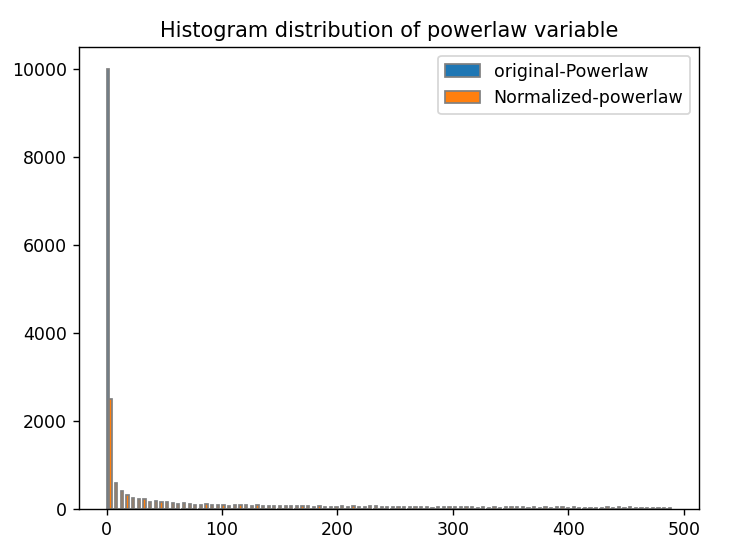
The boxplot appears the same for all three distributions (Gaussian, Power-law, and Geometric) when using percentile normalization because the data is being adjusted based on the relative position of values (percentiles) rather than the raw values themselves. This normalization removes differences in scale and spread.

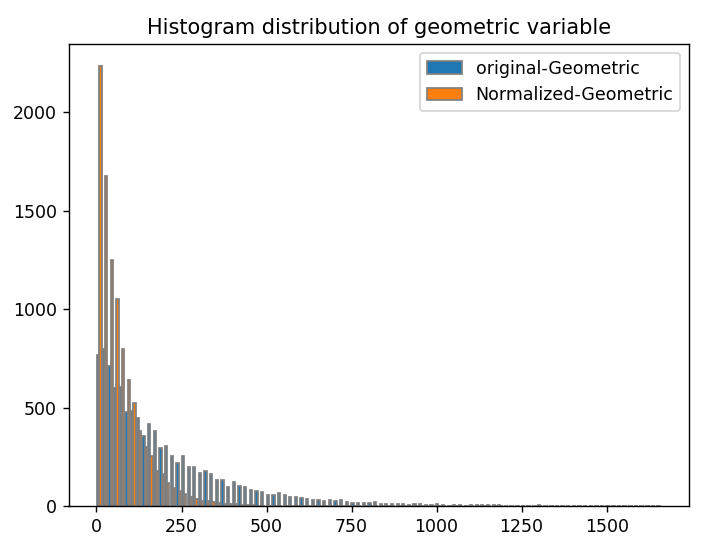
 

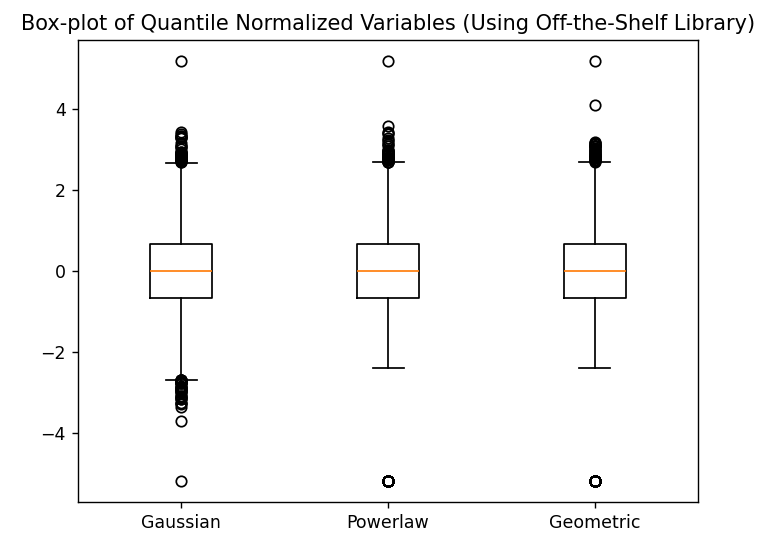


**Gaussian distribution:** The boxplot will look symmetric with a consistent spread. Low IQR

**Power-law distribution:** The boxplot may still show a skewed shape with outliers, even though the median is the same.

**Geometric distribution:** The boxplot still have a concentration of data towards lower values, showing a less uniform spread.

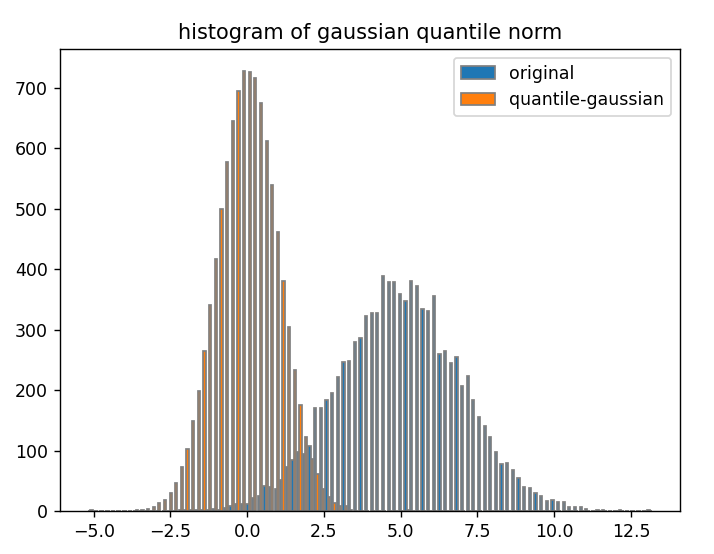
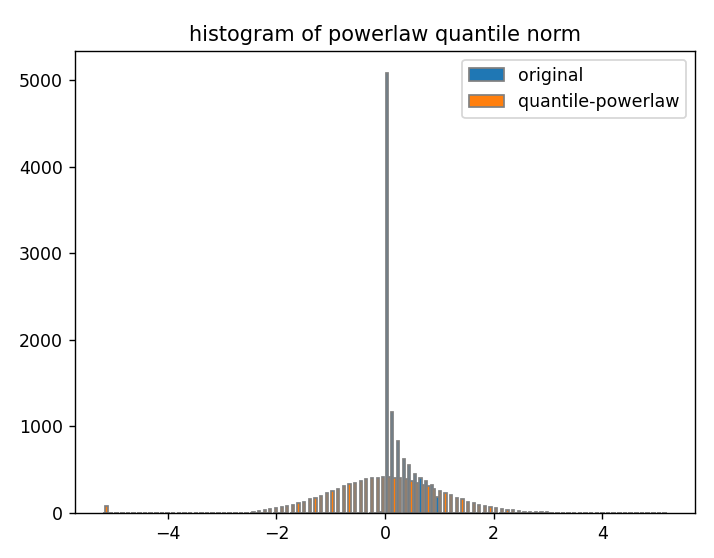


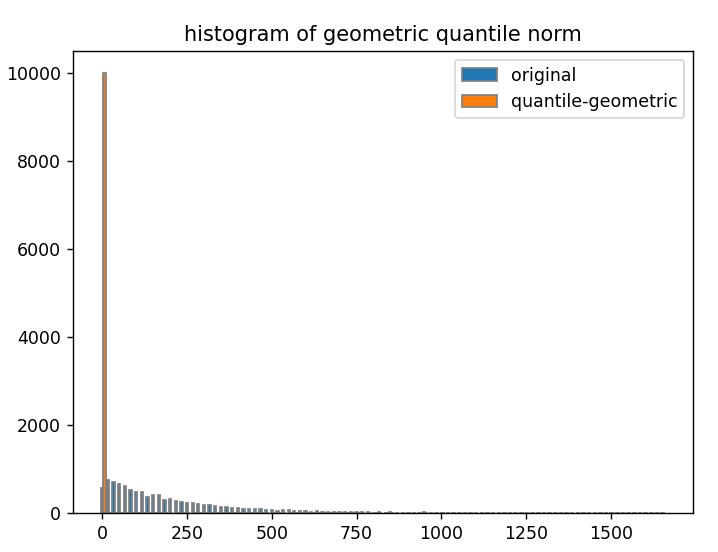


**Gaussian distribution:** The boxplot will appear symmetric, with uniform spread since the data will be adjusted to match the quantiles of the other distributions.

**Power-law distribution:** Same median, Same IQR. Median = 0

**Geometric distribution:** Scale is adjusted to match other two normalizations with same median and IQR.



****

**Observations:**

Quantile normalization is the most effective in making all variables comparable.

Z-score and Percentile normalization are useful for reducing the effects of skewed data.

Max and Sum normalization help in bringing data to a comparable range but do not fix distribution shape differences.