Assignment 2 -Normalization methods

Name: Jasmine Sayyad

PRN: 22510033

Batch: T2

Code:

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import powerlaw

from scipy.stats import rankdata

gaussian\_mean = 5

gaussian\_sd = 2

totalsize = 10000

np.random.seed(45)

# 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)

unique\_gaussian = len(np.unique(B))

unique\_powerlaw = len(np.unique(I))

unique\_geometric = len(np.unique(H))

print(f"no. of unique values in Gaussian: {unique\_gaussian}")

print(f"no. of unique values in Powerlaw: {unique\_powerlaw}")

print(f"no. of unique values in Geometric: {unique\_geometric}")

def plot\_histogram(data\_list, labels, title):

plt.hist(data\_list, bins=100, label=labels, alpha=0.7)

plt.xlabel("Value")

plt.ylabel("Frequency")

plt.title(title)

plt.legend(loc='upper right')

plt.show()

plt.boxplot([B, I, H], labels=["Gaussian", "Powerlaw", "Geometric"])

plt.title("Box-plot distribution")

plt.show()

plot\_histogram([B], ["Gaussian"], "Histogram distribution of Gaussian variable")

plot\_histogram([I], ["Powerlaw"], "Histogram distribution of Powerlaw variable")

plot\_histogram([H], ["Geometric"], "Histogram distribution of Geometric variable")

# Percentile Normalization - tie breaking methods:

def percentile(array, method\_used):

curr\_value = rankdata(array, method=method\_used)

percentile\_value = ((curr\_value-1) / (totalsize-1)) \* 100

return percentile\_value

tie\_methods = ["average","min","max","ordinal","dense"]

for method in tie\_methods:

gaussian\_percentile = percentile(B, method)

powerlaw\_percentile = percentile(I, method)

geometric\_percentile = percentile(H, method)

plt.boxplot([gaussian\_percentile, powerlaw\_percentile, geometric\_percentile],

labels=["Gaussian", "Powerlaw", "Geometric"],

patch\_artist=True,

boxprops=dict(facecolor="lightgreen"))

plt.title(f"Box-plot of Percentile Normalized Variables ({method} method)")

plt.show()

plot\_histogram([B, gaussian\_percentile],["Original-Gaussian", f"Normalized-Gaussian ({method})"],f"Histogram - Gaussian variable using {method}")

plot\_histogram([I, powerlaw\_percentile],["Original-Powerlaw", f"Normalized-Powerlaw ({method})"],f"Histogram - Powerlaw variable using {method}")

plot\_histogram([H, geometric\_percentile],["Original-Geometric", f"Normalized-Geometric ({method})"],f"Histogram - Geometric variable using {method})

geo\_percentile = percentile(H, "dense")

plt.hist(geo\_percentile, bins=100, label="fv", alpha=0.7)

plt.xlabel("Value")

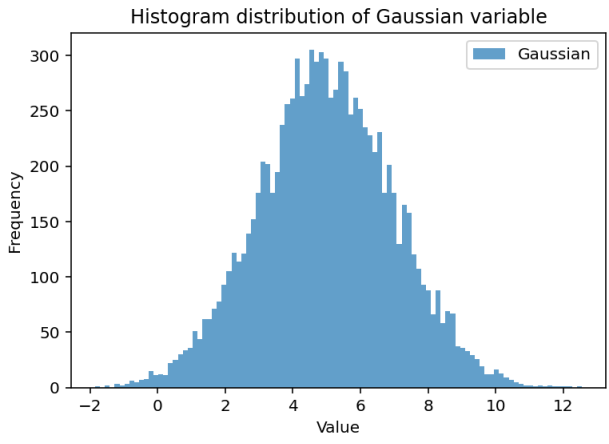
plt.ylabel("Frequency")

plt.title("dense hist")

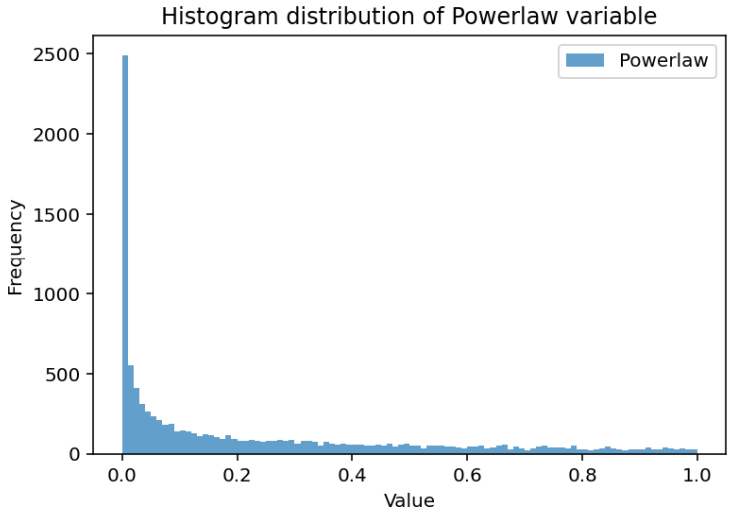
plt.legend(loc='upper right')

plt.show()

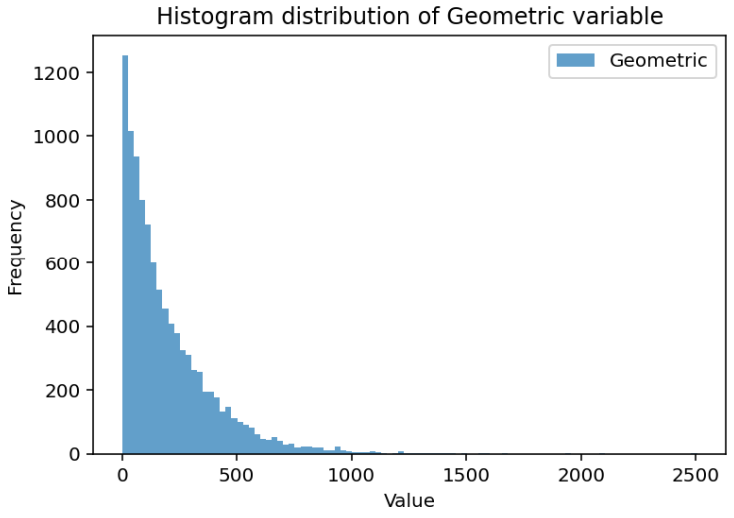
* 1. Generate 3 variables, 10000 samples each
     1. B: Gaussian mean 5 sd 2



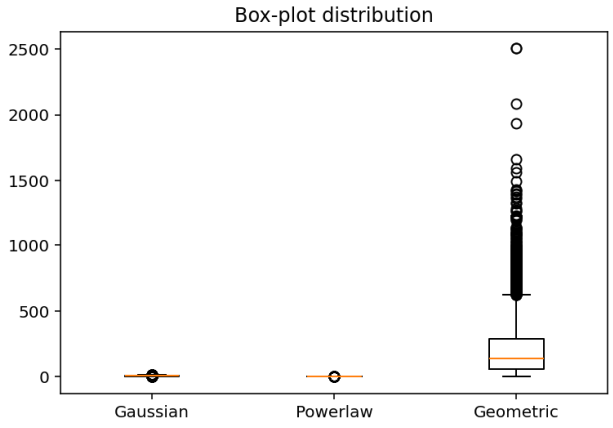
* + 1. I : Power law , use scipy.stats.powerlaw.rvs with a= 0.3,



* + 1. H : Geometric p= 0.005



1. Compare above variables in single box plot



1) In Gaussian distribution, the median will approximately lie close to 5 since mean = 5 and as SD = 2, with 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.

3) In geometric distribution, p = 0.005 which is very small so, the no. of trials required for first success is large. Therefore, median is high, also most of the outliers lie at the upper whisker

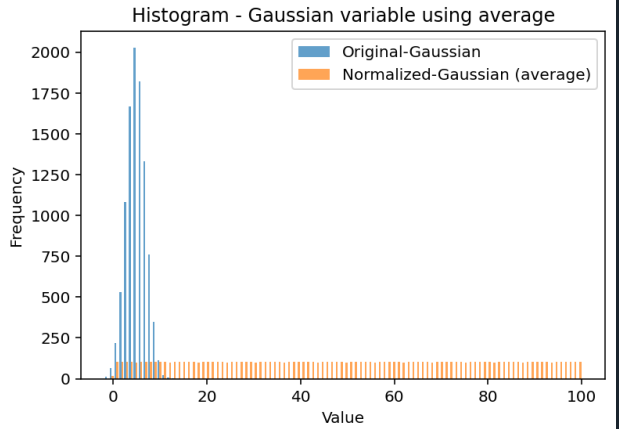
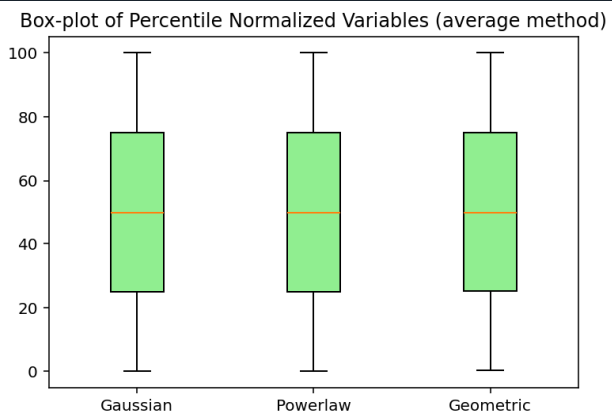
For odd PRNs

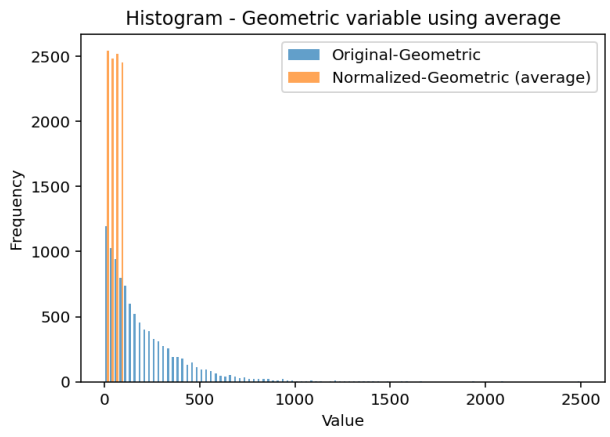
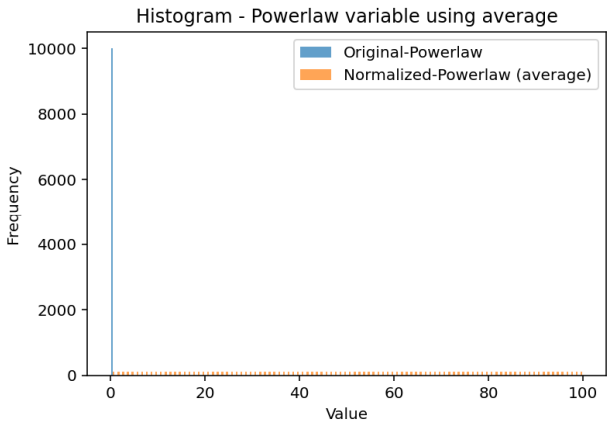
Try percentile normalization using all the tie breaking mechanism

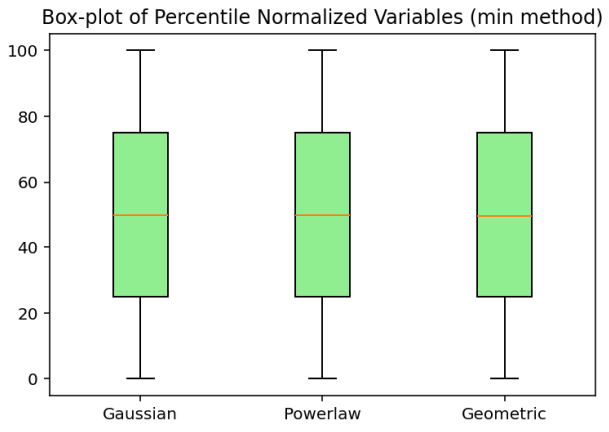
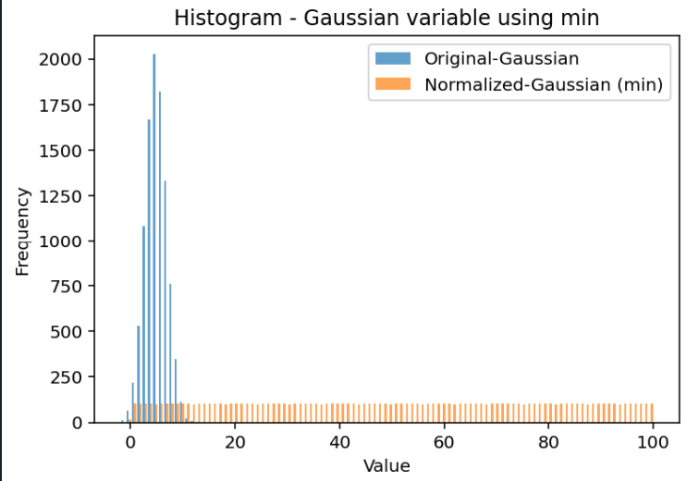
Compare the impact of each tie breaking method on range and shape of the output distributions

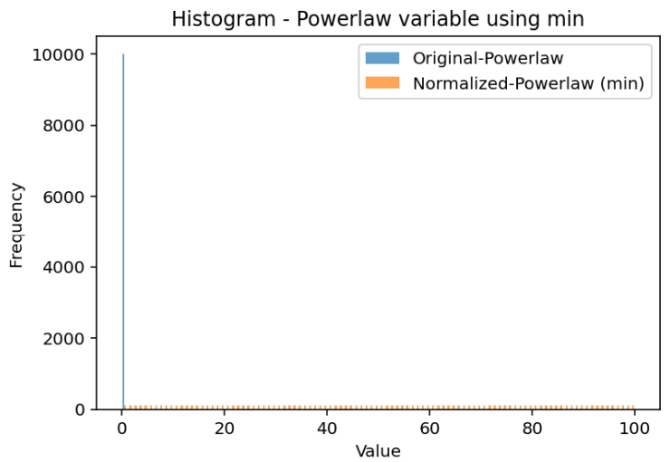
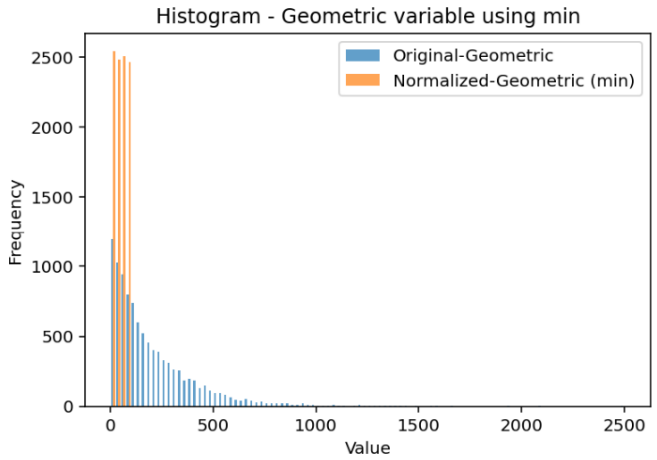
Tie-breaking methods:

1. Average

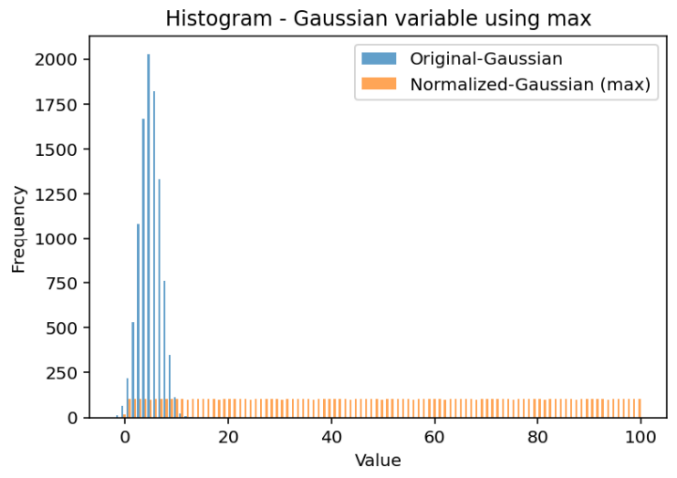


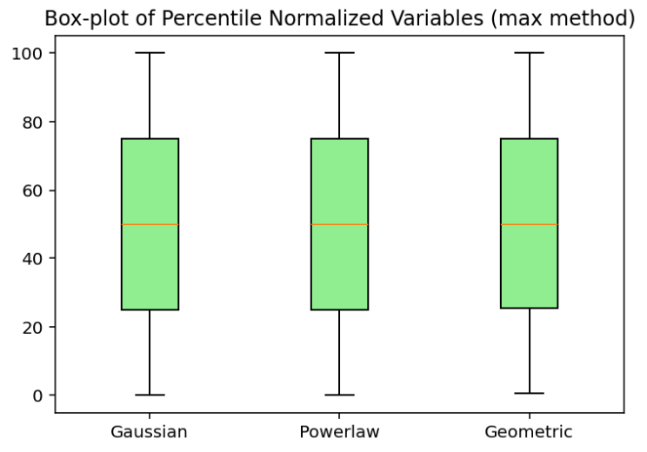


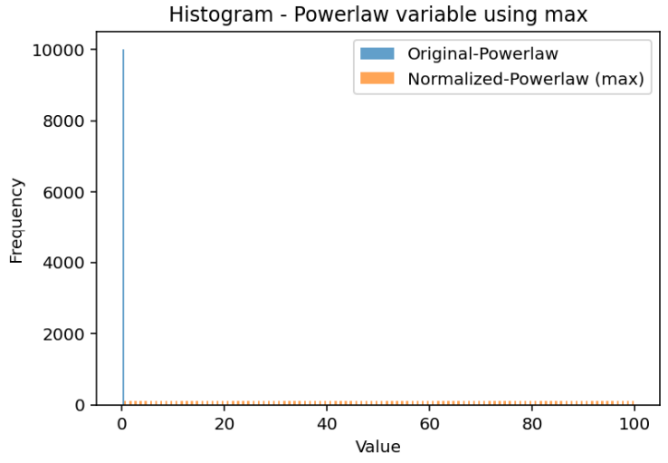
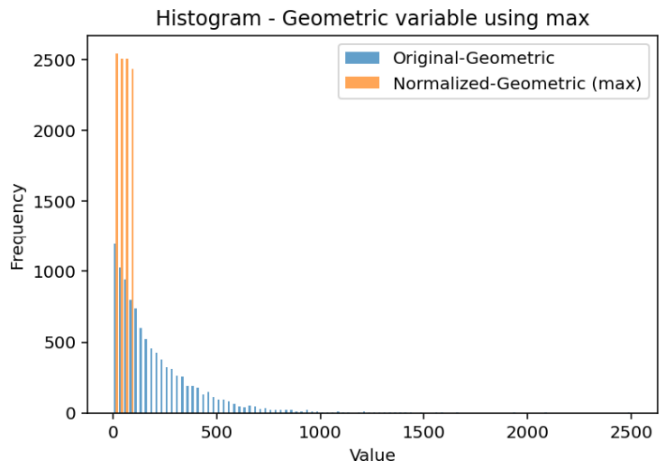
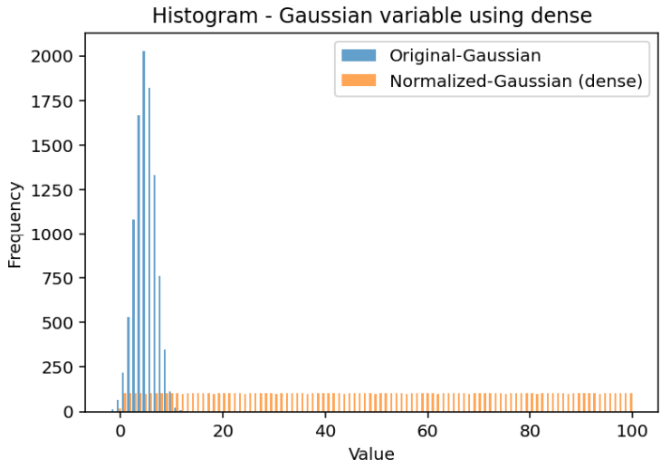
1. Min method

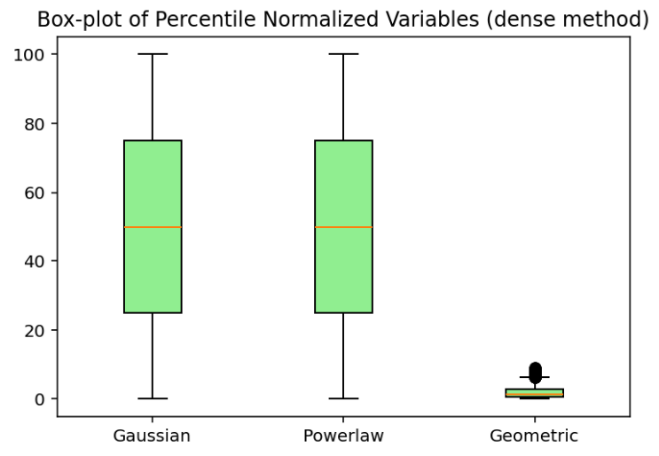
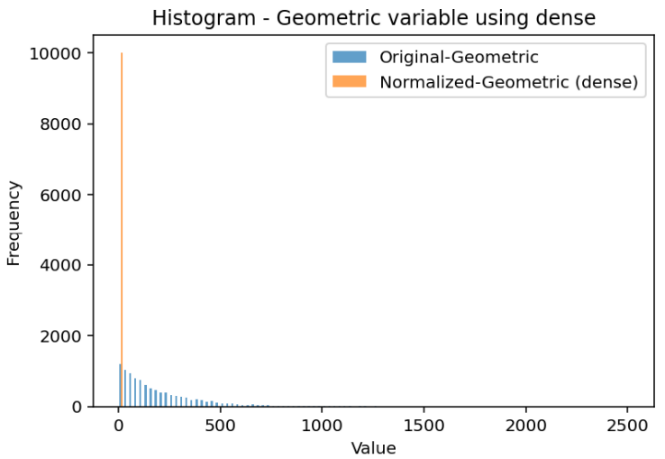


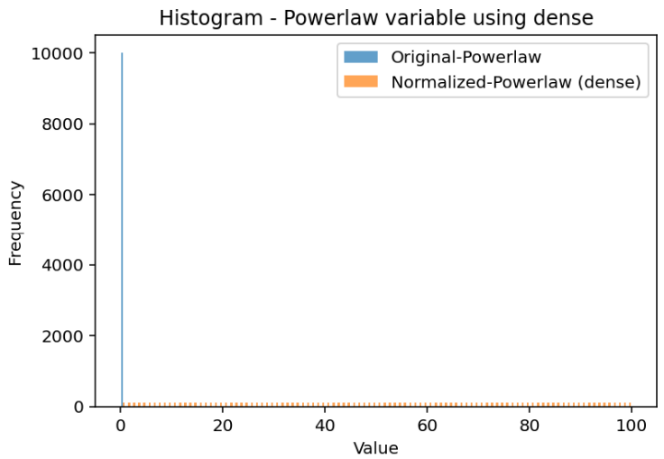
1. Max Method



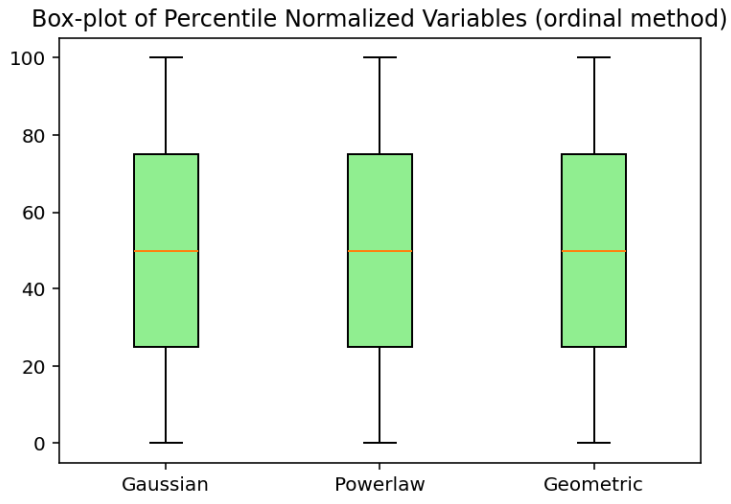


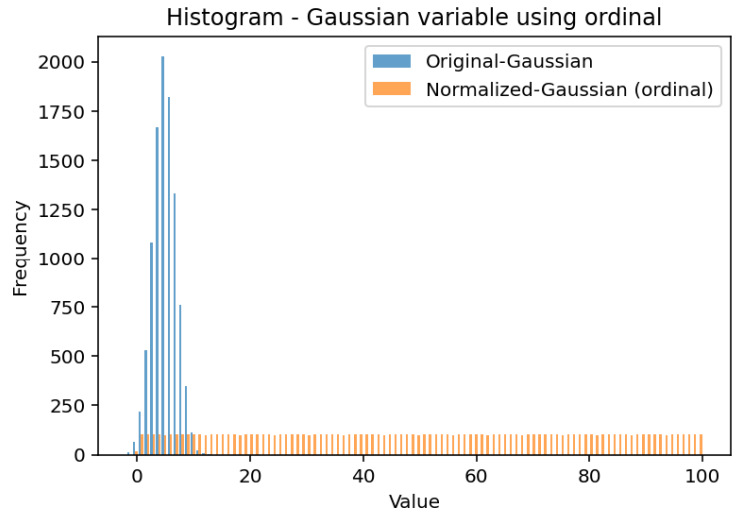
Dense Method

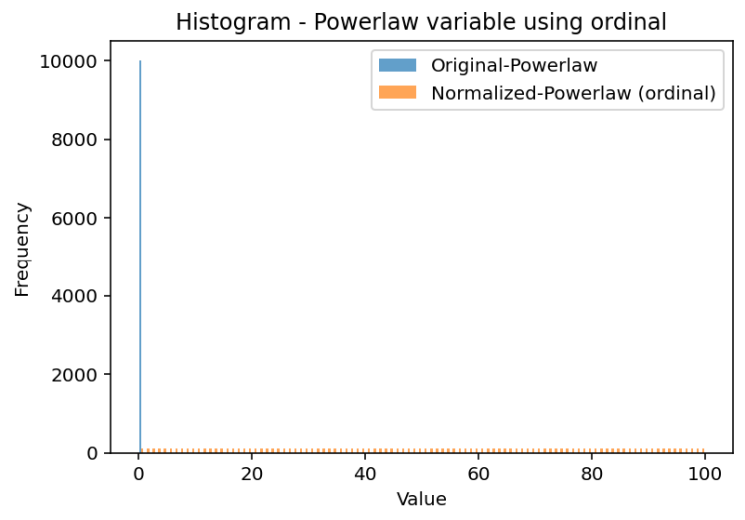


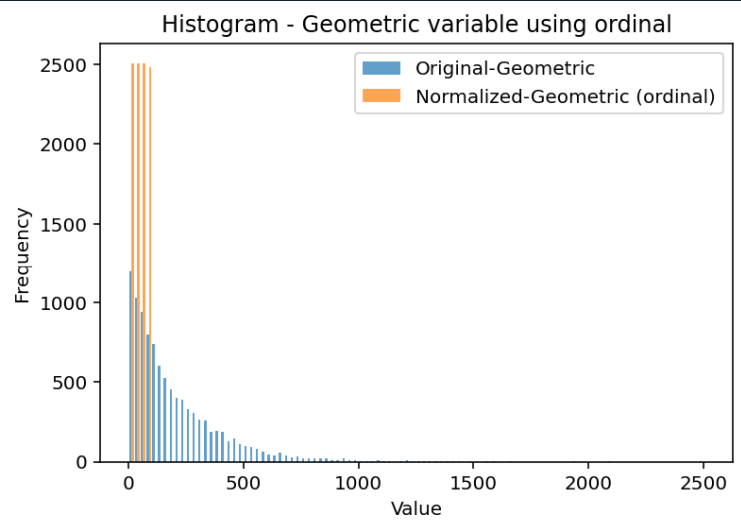


Ordinal Method:







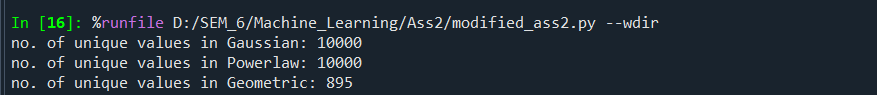


Observations:

Normalized percentiles – spread uniformly across the 0–100 range

In case on power law distribution, the compressed values are stretched across full range removing the skewness of data

For geometric distribution, in case of dense method (The **dense method** assigns **ranks without skipping numbers**, even if there are ties), the range is compressed because of lower number of unique percentile ranks and multiple ties.



Elsewhere for all the methods the range remains same 0-100 and the shape becomes uniform and skewness is removed after normalization from power law and geometric distribution.