Statistical Tasks Report

# Task 1: Comparison of Confidence Intervals for a Normally Distributed Population Variance

## 1. Theoretical Background

For a normally distributed population with variance σ², the theoretical (1 - α) × 100% confidence interval for the variance is given by:  
 ((n-1)s² / χ²\_(1−α/2), (n-1)s² / χ²\_(α/2))  
Where:  
- s² is the sample variance,  
- n is the sample size,  
- χ²\_(α/2) and χ²\_(1−α/2) are critical values from the Chi-square distribution with n−1 degrees of freedom.

## 2. Practical Implementation with Bootstrap

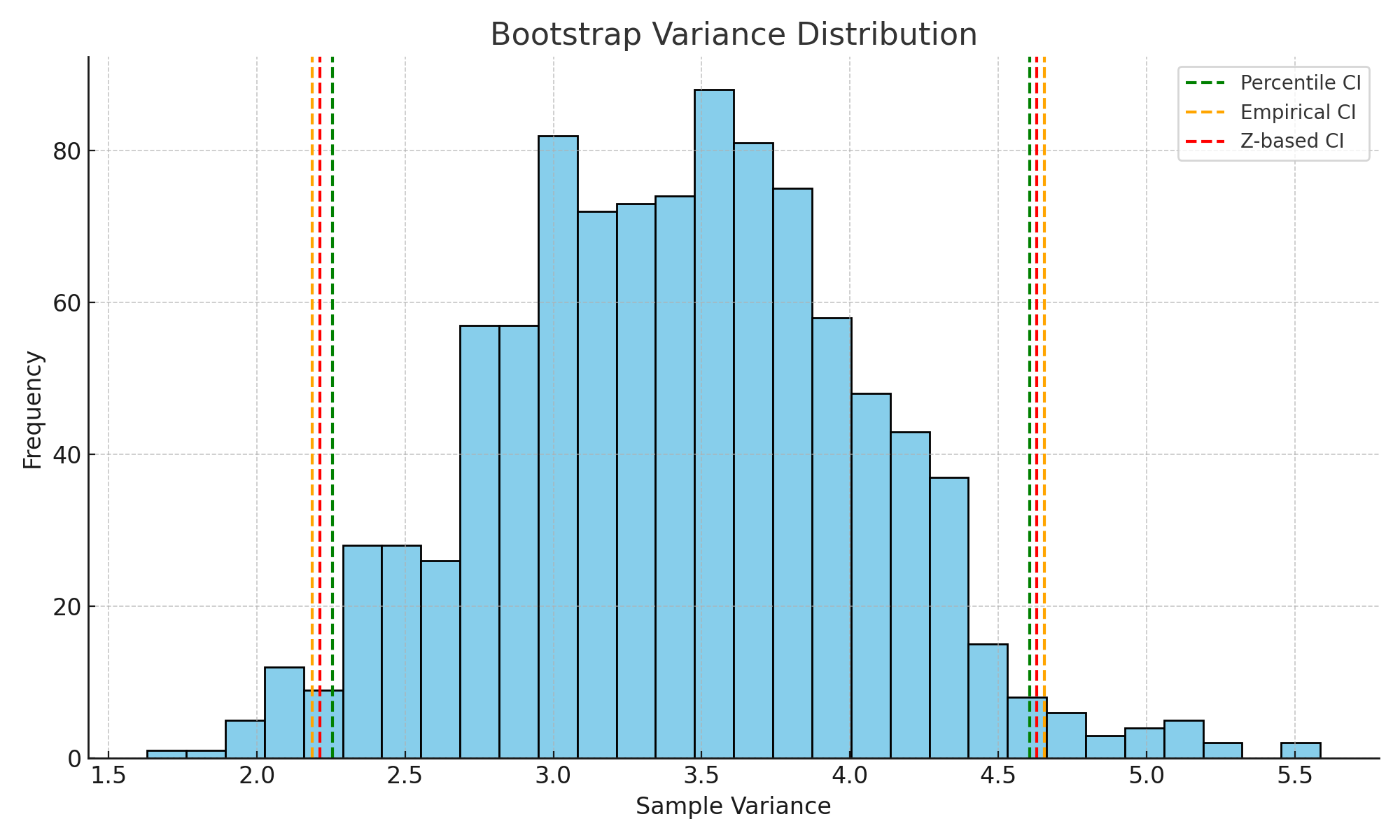
Bootstrap is a non-parametric resampling method that estimates the distribution of a statistic by repeated resampling from the data.  
We calculate three confidence intervals for the variance:  
1. Percentile Method: Uses the 2.5th and 97.5th percentiles of the bootstrapped variance estimates.  
2. Empirical Method: Uses mean ± 2 × SE assuming symmetry and approximate normality.  
3. Z-based Method: Uses mean ± z × SE, where z is from the standard normal distribution.

### Python Code Snippet

import numpy as np  
from scipy.stats import chi2, norm  
import matplotlib.pyplot as plt  
  
np.random.seed(42)  
data = np.random.normal(0, 2, size=50)  
n = len(data)  
sample\_variance = np.var(data, ddof=1)  
  
# Theoretical CI  
alpha = 0.05  
theo\_lower = (n - 1) \* sample\_variance / chi2.ppf(1 - alpha / 2, n - 1)  
theo\_upper = (n - 1) \* sample\_variance / chi2.ppf(alpha / 2, n - 1)  
  
# Bootstrap  
B = 1000  
bootstrap\_vars = [np.var(np.random.choice(data, n, replace=True), ddof=1) for \_ in range(B)]  
bootstrap\_mean = np.mean(bootstrap\_vars)  
bootstrap\_se = np.std(bootstrap\_vars, ddof=1)  
  
# CI Methods  
perc\_ci = np.percentile(bootstrap\_vars, [2.5, 97.5])  
emp\_ci = [bootstrap\_mean - 2 \* bootstrap\_se, bootstrap\_mean + 2 \* bootstrap\_se]  
z\_crit = norm.ppf(1 - alpha / 2)  
z\_ci = [bootstrap\_mean - z\_crit \* bootstrap\_se, bootstrap\_mean + z\_crit \* bootstrap\_se]

## 3. Findings from Visualization

The histogram displays the distribution of bootstrapped sample variances.  
- The Percentile CI (green dashed lines) accurately reflects the spread without assuming normality.  
- The Empirical CI (orange dashed lines) provides a symmetric estimate centered at the mean.  
- The Z-based CI (red dashed lines) gives the narrowest interval due to its parametric nature.  
Overall, all methods are closely aligned, showing that bootstrapping offers valid estimates even when theoretical assumptions are uncertain.



# Task 2: Estimating Parameters (a, b) for a Uniform Distribution Using Method of Moments

## 1. Theoretical Background

For a continuous uniform distribution U(a, b):  
- Mean: μ = (a + b) / 2  
- Variance: σ² = (b - a)² / 12  
  
Using method of moments:  
- From x̄ = (a + b) / 2 and s² = (b - a)² / 12, solve for a and b:  
 a = x̄ - √(3s²), b = x̄ + √(3s²)

### Python Code Snippet

data\_uniform = np.random.uniform(2, 10, 100)  
x\_bar = np.mean(data\_uniform)  
s2 = np.var(data\_uniform, ddof=1)  
  
# Method of moments estimators  
a\_est = x\_bar - np.sqrt(3 \* s2)  
b\_est = x\_bar + np.sqrt(3 \* s2)

## 2. Findings from Visualization

The histogram shows sample data from U(2, 10).  
- The red and green lines show the estimated a and b using the method of moments.  
- The bounds align well with the actual data range, demonstrating the effectiveness of the estimators.  
This task illustrates how the method of moments uses basic sample statistics to derive intuitive and practical parameter estimates.

