Independent University, Bangladesh

Department of Physical Sciences

A Comparative Assessment of Methods for Constructing Confidence Intervals for the Population Coefficient of Variation

MAT 499 : Senior Project

This senior project has been conducted in partial fulfillment of the requirements for the degree of Bachelor of Science with a major in Mathematics

Ishrat Sumaiya Khan

ID : 2130078

Under the supervision of Prof. Dr. Shipra Banik

Declaration

This senior project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Mathematics (Hons). I declare that this project is my original work, except where stated otherwise. This project has never been submitted for any degree or examination to any other University.

Ishrat Sumaiya Khan

This project was conducted under the supervision of Prof. Dr. Shipra Banik.

Certification of Senior Project

The project titled "A Comparative Assessment of Methods for Constructing Confidence Intervals for the Population Coefficient of Variation" submitted by Ishrat Sumaiya Khan, ID: 2130078, has been approved in partial fulfillment of the requirement for the degree of Bachelor of Science in Mathematics (Hons) on August 20, 2026.

Board of Examiners

1. .....
   Prof. Dr. Shipra Banik (Supervisor)
   Department of Physical Sciences, IUB.

2. .....
   Rifat Ara Rouf, PhD (Head of the Department)
   Associate Professor and Head
   Department of Physical Sciences, IUB.

3. .....
   Mohammad Manir Hossain Mollah, Ph.D (Member)
   Associate Professor
   Department of Physical Sciences, IUB.

4. .....
   Md. Alim Hossen, PhD (Member)
   Assistant Professor
   Department of Physical Sciences, IUB.

Dedication

To my parents, for their love and sacrifices.

To my siblings, for their support and encouragement.

To my husband, for his patience, understanding and constant support.

Acknowledgement

All praise and gratitude belong to my Lord, Allah, the Most Merciful and the Most Compassionate. I am deeply thankful for His endless blessings, guidance, and strength, without which this journey would not have been possible.

I would like to express my sincere and heartfelt gratitude to my supervisor, Dr. Shipra Banik, for her invaluable guidance, constant support, and patience throughout the course of this research. Her insightful suggestions, encouragement, and belief in my abilities played a crucial role in shaping this thesis and motivating me to give my best.

I am profoundly grateful to all the teachers who have taught me throughout my academic journey.Each of them has contributed to my knowledge, character, and growth in unique ways and I will always carry their lessons with me.

My deepest gratitude goes to my parents, my first teachers and greatest supporters and the foundation of all my achievements, and I can never thank them enough.

I would like to thank my seniors and classmates for their guidance, friendship, and constant support throughout my studies. Their cooperation and shared moments made this journey easier and more memorable.

I am also grateful to my sister, brothers, cousins, and best friends for their love, support, and understanding. They helped me stay strong, positive, and mentally well throughout this journey, and I feel truly blessed to have them.

Finally, my heartfelt thanks go to my husband, who stood by me during my times of need with patience, care, and love. .His constant support and belief in me gave me strength when I needed it most.

This thesis is not solely the result of my efforts but a collective reflection of the support, guidance, and prayers of everyone mentioned above.

Abstract

The coefficient of variation (CV) is an important measure of relative variability used across many fields. Constructing accurate confidence intervals (CIs) for the population CV remains challenging, particularly for small samples and skewed distributions. This study compares classical, modified, median-based, and bootstrap CI methods for the CV using simulation under symmetric and skewed distributions. Performance is assessed through coverage probability and average interval width, and real-life datasets from health sciences, business, and manufacturing are used for illustration. The results show that classical methods perform well for normal data, median-based methods are more robust under skewness, and bootstrap methods offer a good balance between accuracy and precision across diverse settings.

Keywords: Coefficient of Variation; Confidence Intervals; Bootstrap Methods; Median-based Methods; Coverage Probability; Skewed Distributions

Contents

1. Introduction

The coefficient of variation (CV) is a widely used statistical measure of relative variability that expresses the standard deviation as a proportion of the mean. Unlike absolute measures of dispersion such as the variance or standard deviation, the CV is dimensionless, making it particularly suitable for comparing variability across datasets measured on different scales or expressed in different units. For a population with mean $\mu>0$ and standard deviation σ,the coefficient of variation is defined as

$$CV = \frac{\sigma}{\mu} \times 100$$

and is commonly reported as a percentage. Owing to its scale-invariant nature, the CV provides a standardized measure of dispersion and is especially valuable in comparative statistical analyses. The sampling properties of the coefficient of variation have been studied extensively in the literature (McKay, 1932; Hendricks & Robey, 1936).

Practical Importance and Applications of the CV

The coefficient of variation plays an important role in a wide range of real-life applications across diverse disciplines. In finance and business, the CV is frequently used to assess the risk return trade-off by comparing the variability of investment returns relative to their expected values. For example, when evaluating two portfolios with different average returns, the CV allows investors to identify which portfolio carries lower relative risk. In economics and marketing, the CV is applied to compare relative variability in income levels, consumer spending, market demand, sales performance, or customer response rates across regions or time

periods. In health sciences and epidemiology, the CV is used to quantify variability in biological and medical measurements such as disease incidence rates, survival times, laboratory test results, and physiological indicators. A lower CV in such contexts often indicates greater consistency and reliability in measurements, which is critical for clinical decision-making and public health planning. In engineering, manufacturing, and quality control, the CV serves as a key indicator of process stability and precision. A low CV reflects consistent production quality, whereas a high CV may signal excessive process variation requiring corrective action. In civil and architectural engineering, the CV is commonly used to assess variability in material properties (e.g., concrete strength or load-bearing capacity), construction tolerances, and structural performance measures, contributing to safety evaluation and design optimization. These examples highlight the practical importance of the CV as a tool for informed decision-making under uncertainty. Methodologically, Mark G. Vangel (1996) developed confidence interval procedures for the CV under normality, showing improved finite-sample performance. The sampling properties of the CV have been extensively studied by Robert G. Miller (1991), who proposed asymptotic test statistics and inference procedures for the coefficient of variation. Later, Krishnamoorthy and Lee (2014) further contributed to this area by developing improved confidence interval procedures for the coefficient of variation, particularly under normal and non-normal settings, demonstrating enhanced accuracy in small samples. These studies demonstrate that skewness and departures from normality can significantly affect the accuracy of inference based on the CV, motivating the development of alternative confidence interval procedures.

Statistical Challenges in Inference for the CV

Despite its widespread use, the statistical properties of the coefficient of variation are known to be complex. This complexity arises primarily because the CV is a ratio of two random variables, making its sampling distribution highly sensitive to distributional shape and sample size. In many real-world applications such as income data, reliability and failure-time data, survival times, and epidemiological counts the underlying distributions are often skewed rather than symmetric. Under such conditions, the sampling distribution of the CV can be highly non-normal, which may adversely affect statistical inference. As a result, point estimation alone is often insufficient, and the construction of confidence intervals (CIs) for the CV becomes essential for quantifying uncertainty and ensuring reliable inference. Recent developments include likelihood-based and modified estimators proposed by Curto and Pinto (2009) and Panichkitkosolkul (2009) which sought to produce shorter intervals with improved coverage properties. Simulation studies have played a central role in assessing the performance of these methods. Banik and Kibria (2011) and Gulhar et al. (2012) demonstrated that many classical confidence intervals perform well under normality but may become conservative or exhibit poor coverage when sample sizes are small or when the underlying distribution is skewed. To address robustness concerns, bootstrap-based confidence intervals were examined by Banik et al. (2012), showing improved performance under non-normal conditions. Further extensions by Abu-Shawiesh et al (2019) explored large-sample and augmented intervals, highlighting trade-offs between coverage probability and interval width.

Research Gap and Motivation

Although considerable work has been done on confidence interval estimation for the coefficient of variation (CV), many classical methods still rely on normality assumptions, which are often unrealistic in practical applications involving skewed or heavy-tailed data. Under such conditions, the sampling distribution of the CV becomes highly non-normal, leading to inaccurate coverage probability and unstable interval widths, especially for small to moderate sample sizes (Vangel, 1996; Krishnamoorthy & Lee, 2014). To address these limitations, several alternative approaches have been proposed, including bootstrap-based confidence intervals and modified robust procedures. Bootstrap methods are particularly useful because they do not require strict distributional assumptions and can better approximate the sampling distribution in skewed settings (Efron & Tibshirani, 1993; Davison & Hinkley, 1997). In addition, simulation studies have shown that different CV confidence interval methods can perform quite differently depending on the underlying distribution and sample size, but existing comparisons are often limited in scope and do not provide a unified assessment across multiple methods and distributional scenarios (Krishnamoorthy & Lee, 2014). Despite these developments, there is still a lack of comprehensive evaluation comparing classical, modified, and bootstrap confidence interval methods for the CV under a wide range of skewed distributions. Motivated by this gap, the present study conducts a simulation-based comparison of these methods under both symmetric and skewed distributions across varying sample sizes, using coverage probability (CP) and average interval width (AW) as performance criteria. Real-life datasets from applied fields are also used to demonstrate practical relevance.

Objectives of the Study

Motivated by these gaps, the present study conducts a comprehensive simulation-based comparison of several classical, modified, median-based, and bootstrap confidence interval methods for the CV. The specific objectives are to:

1. Evaluate the performance of different CI methods under both symmetric and skewed distributions.
2. Assess interval accuracy using coverage probability and average interval width as primary performance measures.
3. Illustrate the practical implications of the findings using real-life datasets from health sciences, business and manufacturing.

The remainder of this project is organized as follows. Section 2 describes the statistical methodology and the confidence interval estimators considered in the study. Section 3 outlines the simulation design and performance evaluation criteria. Section 4 presents and discusses the simulation results under different distributional settings. Section 5 provides real-life applications to demonstrate the practical usefulness of the methods. Finally, Section 6 concludes the paper with key findings and recommendations for practitioners.

2. Statistical Methodology

Suppose X_i, i = 1, 2, …, n represent a random sample of size n from a normal population with mean μ and standard deviation σ. The sample CV is defined as CV = (sample mean/sample standard deviation)100. The following is a quick description of the various approaches that have been put out for determining confidence intervals for the CV:

2.1 Classical CI (CCL)

Hendricks and Robey (1936) proposed CI for CV first and later reviewed by Koopman et al (1964),and Iglewicz(1967) and suggested the following 100(1-α)% CI, defined as follows:

$$\text{LCL} = \widehat{CV} - t_{(n-1),\frac{\alpha}{2}} S_{\widehat{CV}}$$

$$\text{UCL} = \widehat{CV} + t_{(n-1),\frac{\alpha}{2}} S_{\widehat{CV}}$$

where CV is the sample CV and $S_{\widehat{CV}} = \widehat{CV}/\sqrt{2n}$ and $t_{(n-1),\frac{\alpha}{2}}$ is the $(\alpha/2)^{\text{th}}$ quintile of the t-distribution with (n-1) df.

2.2 McKay's CI (MK)

McKay (1932) proposed a CI, based on a normal population using the McKay method. A 100(1-α)% CI is as follows:

$$LCL = \widehat{CV}\sqrt{\left(\frac{\chi^2_{(n-1),(1-\frac{\alpha}{2})}}{n} - 1\right)\widehat{CV}^2 + \frac{\chi^2_{(n-1),\frac{1-\alpha}{2}}}{n}}$$

$$UCL = \widehat{CV}\sqrt{\left(\frac{\chi^2_{(n-1),(\frac{\alpha}{2})}}{n} - 1\right)\widehat{CV}^2 + \frac{\chi^2_{(n-1),\frac{\alpha}{2}}}{n}}$$

where $\chi^2_{(n-1),\left(1-\frac{\alpha}{2}\right)}$ and $\chi^2_{(n-1),\left(\frac{\alpha}{2}\right)}$ are respectively the $100(1-\alpha/2)^{th}$ and $100(\alpha/2)^{th}$ percentiles of the chi-square distribution with (n−1) degrees of freedom.

2.3 Miller's CI (ML)

Miller (1991), using the normal distribution as a basis, proposed the following CI:

$$LCL = \widehat{CV} - z_{\frac{\alpha}{2}}\sqrt{\frac{CV^4+0.5CV^2}{n-1}}$$

$$UCL = \widehat{CV} + z_{\frac{\alpha}{2}}\sqrt{\frac{CV^4+0.5CV^2}{n-1}}$$

Where $\widehat{CV}$ is the sample CV, CV is the population CV and $z_{\alpha_{/2}}$ is the α/2 quantile of the standard normal distribution.

2.4 Sharma and Krishna CI (S&K)

The following are the suggested CIs by Sharma and Krishna (1994):

$$LCL = (\widehat{CV} + \varphi^{-1}(\alpha/2)/\sqrt{n})^{-1}$$

$$UCL = (\widehat{CV} - \varphi^{-1}(\alpha/2)/\sqrt{n})^{-1}$$

Where φ(.) is the cumulative standard normal distribution, $\widehat{CV}$ is the sample CV and α is the level of errors.

2.5 Abu-Shawiesh, Akyuz and Kibria Large sample CI (AA&K-LS)

Abu-Shawiesh et al (2019) hereafter AA&K-LS constructed the following confidence limits:

$$LCL = \widehat{CV}\sqrt{exp(-Z_{1-\frac{\alpha}{2}}\sqrt{A}}$$

$$UCL = \widehat{CV}\sqrt{exp(Z_{1-\frac{\alpha}{2}}\sqrt{A}}$$

Where A = $\frac{G_2+\frac{2n}{(n-1)}}{n}$, $G_2 = \frac{n-1}{(n-2)(n-3)}[(n-1)g_2 + 6]$, $g_2 = \frac{m_4}{m_2^2} - 3$, $m_4 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^4$ and $m_2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2$ and $Z_{1-\frac{\alpha}{2}}$ is the α/2 quantile of the standard normal distribution.

2.6 Abu-Shawiesh, Akyuz and Kibria Augmented Large sample CI (AA&K-ALS)

Abu-Shawiesh et al (2019) hereafter AA&K_ALS proposed the following confidence intervals

$$LCL = \widehat{CV}\sqrt{\exp exp\left(-Z_{1-\frac{\alpha}{2}}\right)\sqrt{B} + C})$$

$$UCL = \widehat{CV}\sqrt{\exp exp\left(-Z_{1-\frac{\alpha}{2}}\right)\sqrt{B} + C})$$

Where $B = \widehat{var}(\log log\left(S^2\right))$, $C = \frac{\widehat{K}_{e,5}+\frac{2n}{(n-1)}}{2n}$, $\widehat{K}_{e,5} = (\frac{n+1}{n-1})G_2(1 + \frac{5G_2}{n})$ and $G_2$is defined above.

2.7 Panichkitkosolkuls CI (PK)

Panichkitkosolkul (2009) introduces an enhanced approach for constructing CI for the CV in a normal distribution. This method substitutes the conventional sample CV with the maximum likelihood estimator, resulting in improved performance regarding coverage probability and expected interval length.

$$LCL = \tilde{k}\sqrt{\left(\frac{\chi^2_{(n-1),(1-\frac{\alpha}{2})}+2}{n} - 1\right)\tilde{k}^2 + \frac{\chi^2_{(n-1),\frac{1-\alpha}{2}}}{n}}$$

$$UCL = \tilde{k}\sqrt{\left(\frac{\chi^2_{(n-1),(\frac{\alpha}{2})}+2}{n} - 1\right)\tilde{k}^2 + \frac{\chi^2_{(n-1),\frac{\alpha}{2}}}{n}}$$

Where $\tilde{k} = \frac{\sqrt{\sum_{i=1}^{n}(X_i - \bar{x})^2}}{\sqrt{n\bar{x}}}$

2.8 Bootstrap t-approach CI (BT)

The bootstrap approach is a powerful, flexible, and computationally feasible way of estimating the uncertainty of a statistic without making strict assumptions about the underlying data distribution (Efron, 1979; Efron & Tibshirani, 1993). It is especially useful for small datasets or when traditional methods fail or are difficult to apply. More clearly this approach refers to a statistical method that involves resampling with replacement from a given dataset to estimate the sampling distribution of a statistic. It is commonly used in situations where it is difficult or impossible to make strong assumptions about the underlying population or to derive an analytical formula for the distribution of a statistic. Here is an overview of how it works: We take repeated random samples (called "bootstrap samples") from our data, with replacement. This means some data points may appear multiple times in the same sample, while others may not appear at all. By calculating the statistic of interest on each bootstrap sample, we generate an empirical sampling

distribution for that statistic. The bootstrap method is especially useful for estimating CI or conducting hypothesis tests without relying on parametric assumptions (like normality). Unlike traditional methods (such as t-tests, ANOVA, or linear regression), which often assume normality or specific distributions, the bootstrap method is non-parametric. It relies on the data itself to create distributions. Bootstrap resampling provides a flexible, data-driven approach to confidence interval construction when the sampling distribution of an estimator is unknown or analytically intractable (Efron, 1987; Shao & Tu, 1995). Recent methodological developments in confidence interval construction under challenging distributional settings (Krishnamoorthy & Lee, 2014) further support the exploration of alternative CI procedures beyond classical asymptotic methods. Following standard practice in resampling-based inference (Shao & Tu, 1995), the performance of the proposed confidence interval methods is evaluated using Monte Carlo simulation in terms of coverage probability and average interval width (Banik & Kibria, 2011; Abu-Shawiesh et al., 2019).

The steps in the bootstrap method are as follows: Start with an original sample of size n, say X_i, i =1, 2, …, n. Generate a large number B of new samples by randomly drawing, with replacement, from the original sample. Each bootstrap sample will have the same size n as the original dataset, but may contain repeated data points. For each bootstrap sample, calculate the statistic of interest (e.g., the sample mean, median, variance, regression coefficient, etc.). After performing the above steps, we now have an empirical distribution of the statistic. This distribution approximates the sampling distribution of the statistic. We can construct confidence intervals for the statistic using percentiles from the bootstrap distribution. For example, the 2.5^{th} and 97.5^{th} percentiles of the bootstrap distribution form a 95% confidence interval. The mean or median of the bootstrap distribution can be used as an estimate of the statistic in the population.

Following Banik and Kibria (2010), Gulhar et al (2012) proposed the following CI for CV:

$$\text{LCL} = \widehat{CV} - T^{*}_{(n-1),\alpha/2} S_{\widehat{CV}} \text{ and UCL} = \widehat{CV} + T^{*}_{(n-1),1-\alpha/2} S_{\widehat{CV}}$$

Where $\widehat{CV}$ is the sample CV, $T^{*}_{(n-1),\alpha/2}$ and $T^{*}_{(n-1),1-\alpha/2}$ are the $\alpha/2^{th}$ and $(1-\alpha/2)^{th}$ sample quintiles of $T^{*}_{i} = \frac{CV^{*}_{i} - \bar{CV}}{\hat{\sigma}_{CV}}$, $\hat{\sigma}_{CV} = \sqrt{\frac{1}{B}\sum_{i=1}^{n} (CV^{*}_{i} - \bar{CV})^2}$ and $\tilde{CV} = \frac{1}{n}\sum_{i=1}^{n} CV^{*}_{i}$ is a bootstrap CV.

$$CV^{*}_{(1)} \leq CV^{*}_{(2)} \leq \ldots \leq CV^{*}_{(B)}$$

It is well-known the median is considered the best measure of central tendency in the case of a skewed distribution because it is less affected by extreme values or outliers. Unlike the mean, which can be heavily influenced by unusually large or small data points, the median represents the middle value of a dataset when ordered; ensuring that it accurately reflects the center of the distribution without being distorted by skewed data. In a skewed distribution, where one tail is longer or more spread out than the other, the median provides a more reliable measure of central location, making it a better choice for datasets that are not symmetrically distributed. Based on the assertion that the median depicts the center of the distribution more accurately than the mean for a skewed distribution (see for example, Kibria (2006), Shi and Kibria (2007)), the following intervals are suggested by Gulhar et al (2012):

2.9 Median modification of McKay's CI (MK_Me)

$$LCL = \tilde{CV}\sqrt{\left(\frac{\chi^{2}_{(n-1),(\frac{1-\alpha}{2})}}{n} - 1\right)\tilde{CV}^{2} + \frac{\chi^{2}_{(n-1),\frac{1-\alpha}{2}}}{n}}$$

$$UCL = \widetilde{CV}\sqrt{\left(\frac{\chi^2_{(n-1),(\frac{\alpha}{2})}}{n} - 1\right)\widetilde{CV}^2 + \frac{\chi^2_{(n-1),\frac{\alpha}{2}}}{n}}$$

Where $\widetilde{S} = \sqrt{\frac{1}{n-}\sum_{i=1}^{n}(X_i - \widetilde{X})^2}$ and $\widetilde{X}$ is the sample median.

2.10 Median modification of Miller's CI (ML_Me)

$$LCL = \widetilde{CV} - z_{\alpha_{/2}}\sqrt{\frac{\widetilde{CV}^4 + 0.5\widetilde{CV}^2}{n-1}}$$

$$UCL = \widetilde{CV} + z_{\alpha_{/2}}\sqrt{\frac{\widetilde{CV}^4 + 0.5\widetilde{CV}^2}{n-1}}$$

2.11 Median modification of Curto and Pinto CI (C&P_Me)

$$LCL = \widetilde{CV} - Z_{\frac{\alpha}{2}}\sqrt{\sqrt{\widetilde{CV}^4} + 0.5(\widetilde{CV}^2)}$$

$$UCL = \widetilde{CV} + Z_{\frac{\alpha}{2}}\sqrt{\sqrt{\widetilde{CV}^4} + 0.5(\widetilde{CV}^2)}$$

We suggest some interval estimation methods for the population CV based on the median, following a similar approach to that of Gulhar et al. (2012), to obtain a more precise CI.

2.12 Median modification of classical approach (CCL_Me)

The following 100(1-α)% CI is defined as follows:

$$\text{LCL} = \widetilde{CV} - t_{(n-1),\frac{\alpha}{2}}S_{\widetilde{CV}}$$

$$\text{UCL} = \widetilde{CV} + t_{(n-1),\frac{\alpha}{2}}S_{\widetilde{CV}}$$

where $\widetilde{CV}$ is the sample CV based on median and $S_{\widetilde{CV}} = \widetilde{CV}/\sqrt{2n}$ and $t_{(n-1),\frac{\alpha}{2}}$ is the (α/2)th quintile of the t-distribution with (n-1) df.

2.13 Median modification of Sharma and Krishna approach (S&K_Me)

The following 100(1-α)% CI is defined as follows:

$$LCL = (\widehat{CV} + \varphi^{-1}(\alpha/2)/\sqrt{n})^{-1}$$

$$UCL = (\widehat{CV} - \varphi^{-1}(\alpha/2)/\sqrt{n})^{-1}$$

where φ(.) is the cumulative standard normal distribution, $\widehat{CV}$ is the sample CV based on median and α is the level of errors.

2.14 Median modification of bootstrap approach (BT_Me)

The proposed the following CI for CV:

$$\text{LCL} = \widetilde{CV} - T^{*}_{(n-1),\alpha/2} S_{\widehat{CV}}$$

$$\text{UCL} = \text{UCL} = \widetilde{CV} + T^{*}_{(n-1),1-\alpha/2} S_{\widetilde{CV}}$$

Where $\widetilde{CV}$ is the sample CV, $T^{*}_{(n-1),\alpha/2}$ and $T^{*}_{(n-1),1-\alpha/2}$ are the α/2th and (1- α/2)th sample quintiles of $T^{*}_{i} = \frac{CV^{*}_{i} - \widetilde{\widetilde{CV}}}{\hat{\sigma}_{\widetilde{CV}}}$, $\hat{\sigma}_{\widetilde{CV}} = \sqrt{\frac{1}{B}\sum_{i=1}^{n}(CV^{*}_{i} - \widetilde{\widetilde{CV}})^{2}}$ and $\widetilde{\widetilde{CV}} = \frac{1}{n}\sum_{i=1}^{n} CV^{*}_{i}$ is a bootstrap CV based on median.

3. Simulation design

To compare the performance of our various considered test statistics, a simulation study has been conducted. This study employs a Monte Carlo simulation framework to evaluate confidence interval (CI) estimation for the coefficient of variation (CV). The simulation procedure is structured as follows:

Sample sizes: n=20, 50, 70 and 100.

Data generation process (DGP): For each n, generate 5000 random samples from each of the following distributions:

(i) Normal: N(2,1) and N(4,1)

(ii) Right-skewed: Exponential(2)

(iii) Left-skewed: Beta(3,1)

Statistic computation: For each sample, compute the CV as the ratio of the sample standard deviation to the sample mean (or median, where applicable).

Repetition: Repeat the above steps to obtain the empirical distribution of CV estimates.

CI construction: Construct confidence intervals using the percentile method, based on the $\alpha/2$ and $1-\alpha/2$ quintiles of the simulated CV distribution.

Performance measures:

(iv) CP: Proportion of intervals containing the true CV,(95%)

(v) AW: Mean length of the constructed intervals, reflecting precision.

Following Banik and Kibria (2011), coverage probability and average interval width are used as the primary criteria for evaluating confidence interval performance.

Significance level: A nominal level of $\alpha=0.05$ is used throughout.

Implementation: All simulations and computations are carried out using MATLAB programming codes (please refer in the appendix section).

The simulation results, summarized in terms of CP and AW, are reported in Tables 4.1 to 4.4.

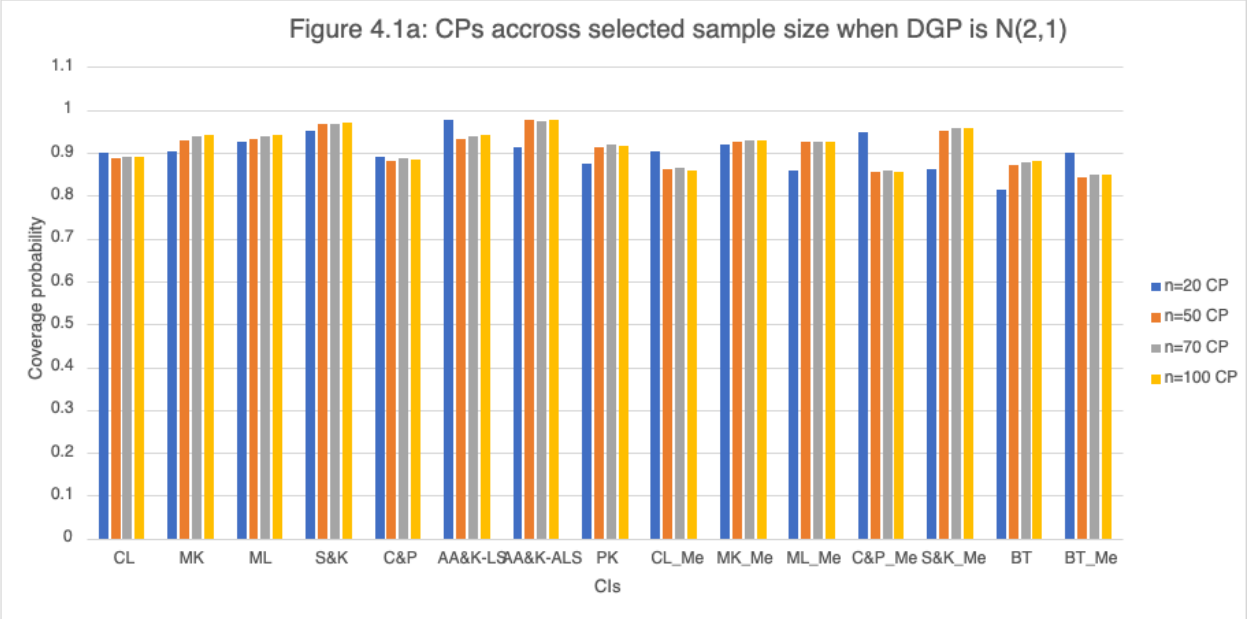
4. Simulation results and discussion

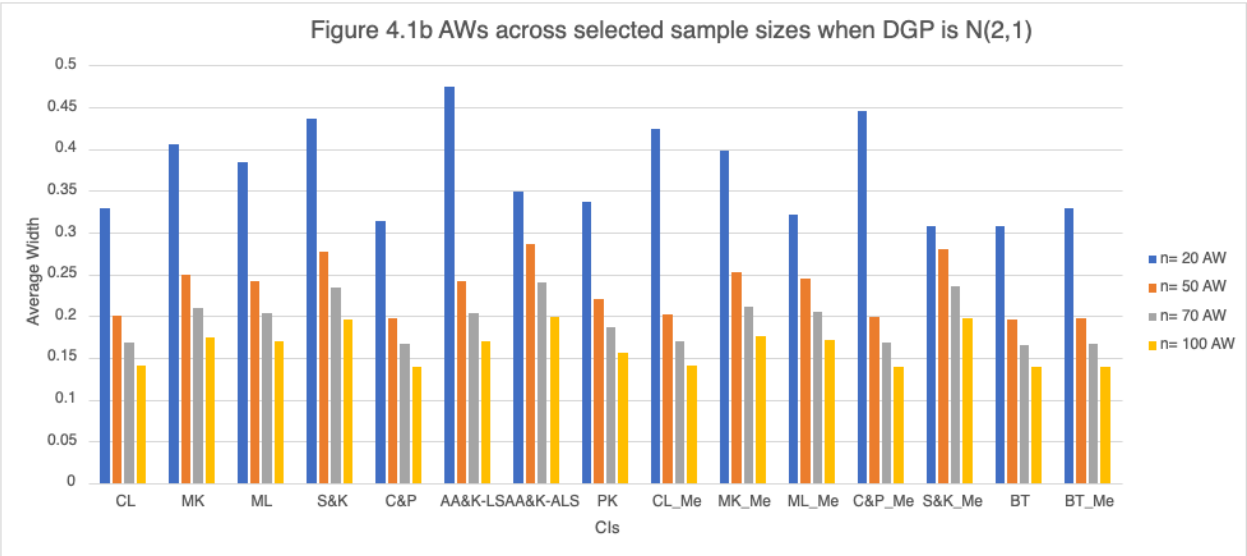
In this section, we present and analyze the simulation results for the N(2,1), N(4,1), Exponential(2), and Beta(3,1) distributions. Tables 4.1 to 4.4 report the CP and AW of the confidence intervals for the CV across different n. Corresponding graphical representations are provided in Figures 4.1(a,b) to 4.4(a,b) to facilitate visual comparison. The key findings from these results are summarized below.

Table 4.1: CP and AW of selected CIs when DGP N(2,1) with skewness 0

CIs	n=20		n=50		n=70		n=100	
	CP	AW	CP	AW	CP	AW	CP	AW
CCL	0.9002	0.3297	0.8874	0.2008	0.8894	0.1687	0.8904	0.1404
MK	0.9040	0.4051	0.9276	0.2488	0.9388	0.2096	0.9418	0.1746
ML	0.9246	0.3842	0.9326	0.2415	0.9396	0.2040	0.9418	0.1704
S&K	0.9520	0.4366	0.9668	0.2770	0.9688	0.2344	0.9708	0.1961
C&P	0.8896	0.3144	0.8820	0.1973	0.8864	0.1666	0.8856	0.1392
AA&K-LS	0.9756	0.4742	0.9326	0.2415	0.9396	0.2040	0.9418	0.1704
AA&K-ALS	0.9120	0.3487	0.9782	0.2863	0.9750	0.2400	0.9770	0.1993
PK	0.8744	0.3368	0.9138	0.2199	0.9184	0.1859	0.9164	0.1553
CCL_Me	0.9020	0.4247	0.8624	0.2026	0.8636	0.1696	0.8602	0.1408
MK_Me	0.9208	0.3976	0.9270	0.2526	0.9300	0.2115	0.9288	0.1756
ML_Me	0.8584	0.3212	0.9246	0.2448	0.9252	0.2057	0.9260	0.1713
C&P_Me	0.9478	0.4460	0.8570	0.1991	0.8582	0.1675	0.8564	0.1396
S&K_Me	0.8626	0.3075	0.9526	0.2795	0.9564	0.2357	0.9588	0.1967
BT	0.8142	0.3077	0.8706	0.1956	0.8794	0.1654	0.8824	0.1385
BT_Me	0.9002	0.3297	0.8418	0.1967	0.8484	0.1663	0.8498	0.1388

Table 4.1 summarizes the 95% CP and AW of 15 different confidence interval methods for the CV under the DGP N(2,1), which has zero skewness. These measures are evaluated across varying n to assess the accuracy (CP) and precision (AW) of each method.





From Table 4.1 and the corresponding figures, several trends emerge for the N(2,1) distribution. Across all methods, an increase in sample size leads to a decrease in average width, indicating improved precision of the confidence intervals.In terms of coverage probability, most methods show a gradual stabilization toward values close to or slightly above 0.90 as sample size increases. This indicates improved reliability with larger samples, although some methods exhibit over or under coverage depending on their construction.Among all the methods, S&K clearly gives the highest coverage probabilities, for example, 0.9520 at n = 20 and increasing to 0.9708 at n = 100. This suggests that the method is quite reliable in capturing the true parameter. However, its interval widths are also the largest (e.g., 0.4366 at n = 20), which means this reliability comes at the cost of less precision. In contrast, MK and ML provide a better balance. Their coverage probabilities are close to the desired level for larger samples (both around 0.94 at n = 100), while their widths remain moderate. This indicates that they manage to maintain accuracy without making the intervals unnecessarily wide. The CCL method produces narrower intervals (e.g., 0.1404 at n = 100), but its coverage stays slightly below the target (around 0.89), meaning it may miss the true value more often. Similarly, C&P shows consistently low coverage (around 0.88-0.89), which suggests under coverage despite having relatively narrow intervals. AA&K-ALS stands out with the highest coverage overall (e.g., 0.9782 at n = 50 and 0.9770 at n = 100). This indicates very strong reliability, but again the intervals are wider (AW = 0.20-0.28), so the method is less efficient. The PK method shows a more balanced behavior, improving from 0.8744 at n = 20 to 0.9164 at n = 100, which suggests it becomes more reliable as sample size increases while keeping widths at a reasonable level. The median based methods show mixed results. For instance, S&K_Me performs quite well for larger samples (0.9588 at n = 100), but others like CCL_Me and C&P_Me have lower coverage (around 0.85-0.86 at n = 100). This may

mean that median-based adjustments do not always improve performance, especially for smaller samples, but can become more stable as n increases. The bootstrap methods (BT and BT_Me) tend to produce narrower intervals (e.g., BT has AW = 0.1385 at n = 100), which is good in terms of precision. However, their coverage is relatively low, especially for small samples (BT = 0.8142 at n = 20). Although the coverage improves with larger samples (up to 0.8824 at n = 100), it still does not reach the nominal level. This suggests that bootstrap methods may underestimate variability in this case. The BT_Me version performs slightly better for small samples but still remains below the desired coverage level overall.

In summary, the results indicate a clear trade off between coverage probability and interval width across all methods:

Best coverage performance: AA&K-ALS and S&K methods

Most balanced performance: MK, ML, and PK methods

Most efficient (narrowest intervals): CCL, C&P, and bootstrap methods

Most stable under increasing n: S&K, ML, and MK variants

The findings also confirm that increasing sample size significantly improves the reliability and precision of all confidence interval estimators.

Table 4.2: CP and AW of selected CIs when DGP N(4,1) with skewness 0

CIs	n=20		n=50		n=70		n=100	
	CP	AW	CP	AW	CP	AW	CP	AW
CCL	0.9298	0.1634	0.9262	0.1000	0.9334	0.0841	0.9304	0.0700
MK	0.9090	0.1627	0.9280	0.1036	0.9392	0.0878	0.9374	0.0735
ML	0.9252	0.1628	0.9348	0.1035	0.9434	0.0877	0.9422	0.0734
S&K	0.9688	0.2164	0.9820	0.1379	0.9842	0.1168	0.9860	0.0978
C&P	0.9222	0.1558	0.9256	0.0982	0.9318	0.0831	0.9282	0.0694
AA&K-LS	0.9252	0.1628	0.9348	0.1035	0.9434	0.0877	0.9422	0.0734
AA&K-ALS	0.9218	0.1580	0.9906	0.1425	0.9892	0.1196	0.9898	0.0995
PK	0.9280	0.1658	0.9268	0.1006	0.9374	0.0852	0.9344	0.0713
CCL_Me	0.9110	0.1655	0.9246	0.1006	0.9236	0.0844	0.9256	0.0702
MK_Me	0.9252	0.1656	0.9272	0.1044	0.9336	0.0882	0.9338	0.0737
ML_Me	0.9170	0.1582	0.9322	0.1043	0.9342	0.0881	0.9384	0.0737
C&P_Me	0.9692	0.2196	0.9202	0.0988	0.9202	0.0834	0.9246	0.0696
S&K_Me	0.9068	0.1521	0.9824	0.1388	0.9832	0.1173	0.9848	0.0981
BT	0.9022	0.1551	0.9204	0.0973	0.9268	0.0825	0.9258	0.0691
BT_Me	0.9298	0.1634	0.9140	0.0981	0.9162	0.0830	0.9212	0.0694

The results in Table 4.2 summarise the 95% CP and AW of 15 different CIs for the CV under the DGP N(4,1) with skewness 0.

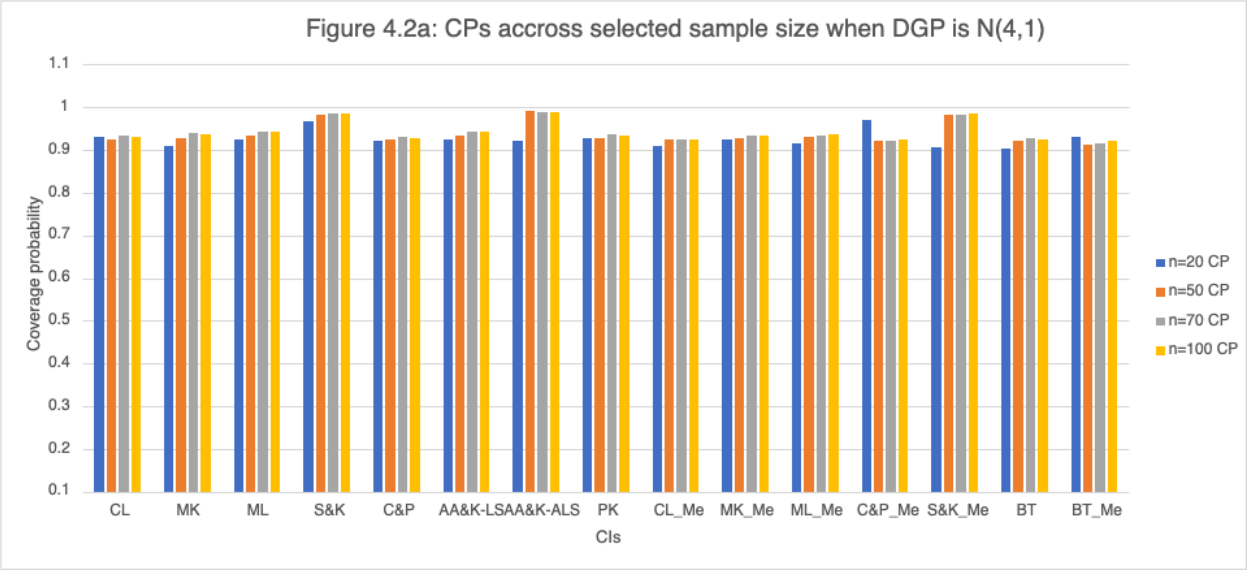


Figure 4.2a: CPs accross selected sample size when DGP is N(4,1)
Coverage probability
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
CL
MK
ML
S&K
C&P
AA&K-LS
AA&K-ALS
PK
CL_Me
MK_Me
ML_Me
C&P_Me
S&K_Me
BT
BT_Me
CIs
n=20 CP
n=50 CP
n=70 CP
n=100 CP

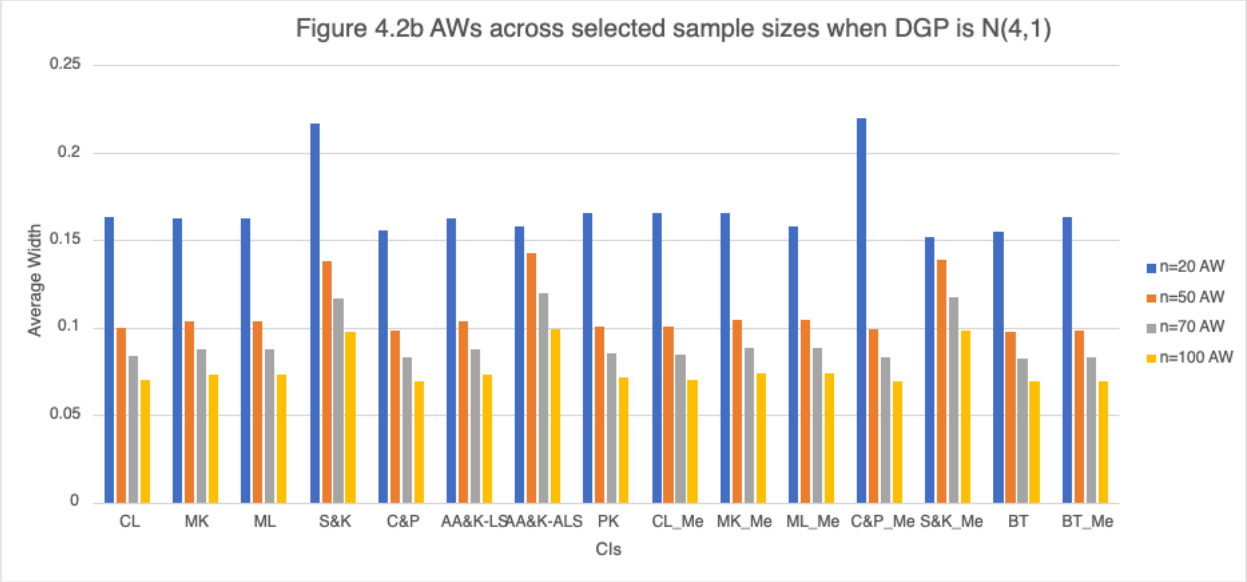


Figure 4.2b AWs across selected sample sizes when DGP is N(4,1)
Average Width
0.25
0.2
0.15
0.1
0.05
0
CL
MK
ML
S&K
C&P
AA&K-LS
AA&K-ALS
PK
CL_Me
MK_Me
ML_Me
C&P_Me
S&K_Me
BT
BT_Me
CIs
n=20 AW
n=50 AW
n=70 AW
n=100 AW

From Table 4.2 and the corresponding graphical representations, it is evident that as the sample size increases, most confidence interval methods for the N(4,1) distribution show improved performance. Across all methods, a clear pattern can be observed: as the sample size increases from n = 20 to n = 100, the average width (AW) decreases, indicating that the confidence intervals become more precise. For example, under the CCL method, AW decreases from 0.1634 to 0.0700, and similar reductions are seen for all other methods. At the same time, the coverage probability (CP) becomes more stable and generally moves closer to the desired level (around 0.95), showing improved reliability with larger samples. Among all the methods, S&K again produces the highest coverage probabilities, increasing from 0.9688 at n = 20 to 0.9860 at n = 100. This suggests that it is very reliable in capturing the true parameter. However, its interval widths are also the largest (e.g., 0.2164 at n = 20 and 0.0978 at n = 100), meaning that this high coverage comes at the cost of wider, less precise intervals. In comparison, MK and ML provide a more balanced performance. Their coverage probabilities improve with sample size (e.g., MK: 0.9090 to 0.9374, ML: 0.9252 to 0.9422), and their interval widths remain moderate (around 0.07-0.10 for larger samples). This indicates that these methods achieve a good trade-off between reliability and precision. The CCL method also performs reasonably well, with relatively narrow intervals (AW = 0.0700 at n = 100), but its coverage remains slightly below the target level (~0.93), suggesting mild under-coverage. Similarly, C&P shows slightly lower CP values (around 0.92-0.93), indicating that it may also miss the true value more often despite having narrow intervals. Among the modified methods, AA&K-ALS stands out with extremely high coverage probabilities, such as 0.9906 at n = 50 and 0.9898 at n = 100.This shows very strong reliability, but again the intervals are wider (AW = 0.10–0.14), meaning reduced efficiency.The PK method shows stable and balanced behavior, with CP values around 0.93–0.94 for larger

samples and relatively small widths (e.g., 0.0713 at n = 100), making it a competitive alternative. The median-based methods show mixed results. For example, S&K_Me performs very well at larger sample sizes (CP = 0.9848 at n = 100), similar to its original version, while methods like CCL_Me and ML_Me show slightly lower coverage (around 0.925–0.938). Interestingly, C&P_Me has very high CP at small sample size (0.9692 at n = 20) but drops for larger samples (around 0.92), suggesting some inconsistency. Overall, median-based adjustments do not always guarantee improvement but tend to stabilize as sample size increases. For the bootstrap methods, BT produces relatively narrow intervals (e.g., AW = 0.0691 at n = 100), which is good in terms of precision. However, its coverage is slightly low, especially at smaller sample sizes (0.9022 at n = 20), and only improves to about 0.9258 at n = 100. The BT_Me method shows a similar pattern, starting at 0.9298 for n = 20 but then slightly decreasing and stabilizing around 0.92. This suggests that bootstrap methods, while efficient, may still underestimate variability and lead to slight under-coverage.

Overall, the results highlight a clear trade-off between coverage probability and interval width:

Best coverage: AA&K-ALS and S&K (but with wider intervals)

Most balanced: MK, ML, and PK

Most efficient (narrow intervals): CCL, C&P, and bootstrap methods

Most stable with increasing sample size: S&K, ML, and MK

In conclusion, increasing sample size improves both the precision and reliability of all methods. However, no single method is best in all situations, and the choice depends on whether priority is given to higher coverage or narrower intervals.

Table 4.3: CP and AW of selected CIs when DGP EXP(2) with skewness 2.0618

CIs	n=20		n=50		n=70		n=100	
	CP	AW	CP	AW	CP	AW	CP	AW
CCL	0.8750	0.6348	0.8604	0.3950	0.8552	0.3329	0.8532	0.2779
MK	0.9334	1.3648	0.9668	0.8395	0.9730	0.6884	0.9750	0.5671
ML	0.9476	1.0309	0.9700	0.6685	0.9740	0.5670	0.9730	0.4759
S&K	0.9484	0.8407	0.9564	0.5448	0.9548	0.4625	0.9504	0.3882
C&P	0.8674	0.6055	0.8586	0.3881	0.8544	0.3288	0.8520	0.2755
AA&K-LS	0.9476	1.0309	0.9700	0.6685	0.9740	0.5670	0.9730	0.4759
AA&K-ALS	0.9784	0.9131	0.9670	0.5630	0.9634	0.4735	0.9560	0.3946
PK	0.9230	0.8428	0.9444	0.5460	0.9506	0.4631	0.9486	0.3886
CCL_Me	0.5336	1.0147	0.2762	0.6112	0.1728	0.5128	0.1018	0.4239
MK_Me	0.9642	3.0415	0.9974	2.2819	0.9996	2.0373	0.9988	0.7074
ML_Me	0.9700	2.6885	0.9974	1.5123	0.9030	1.2520	0.4914	1.0215
C&P_Me	0.4980	0.9678	0.2610	0.6006	0.1654	0.5065	0.0972	0.4203
S&K_Me	0.6988	1.3438	0.4212	0.8430	0.2872	0.7125	0.1802	0.5921
BT	0.3836	0.8935	0.8556	0.3814	0.8558	0.3243	0.8514	0.2723
BT_Me	0.8750	0.6348	0.1994	0.5829	0.1288	0.4963	0.0724	0.4139

The performance of 15 confidence interval methods for estimating the CV under the EXP(2) distribution (skewness = 2.0618) is presented in Table 4.3, with CP and AW reported for selected n.

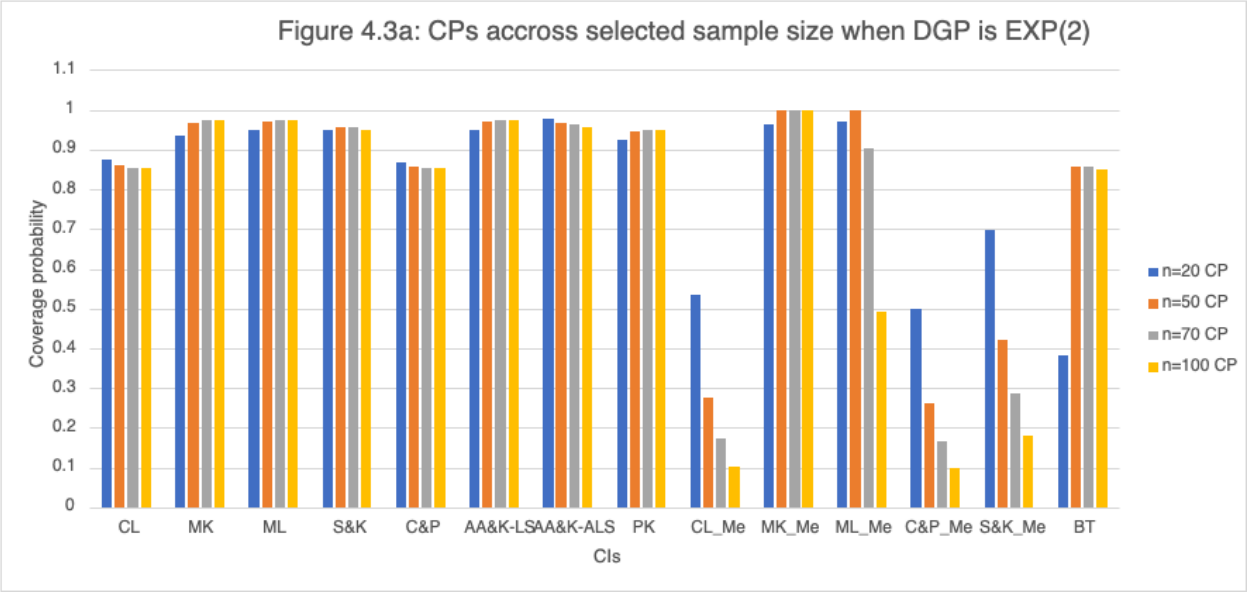


Figure 4.3a: CPs accross selected sample size when DGP is EXP(2)
Coverage probability
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
CL
MK
ML
S&K
C&P
AA&K-LS
AA&K-ALS
PK
CL_Me
MK_Me
ML_Me
C&P_Me
S&K_Me
BT
CIs
n=20 CP
n=50 CP
n=70 CP
n=100 CP

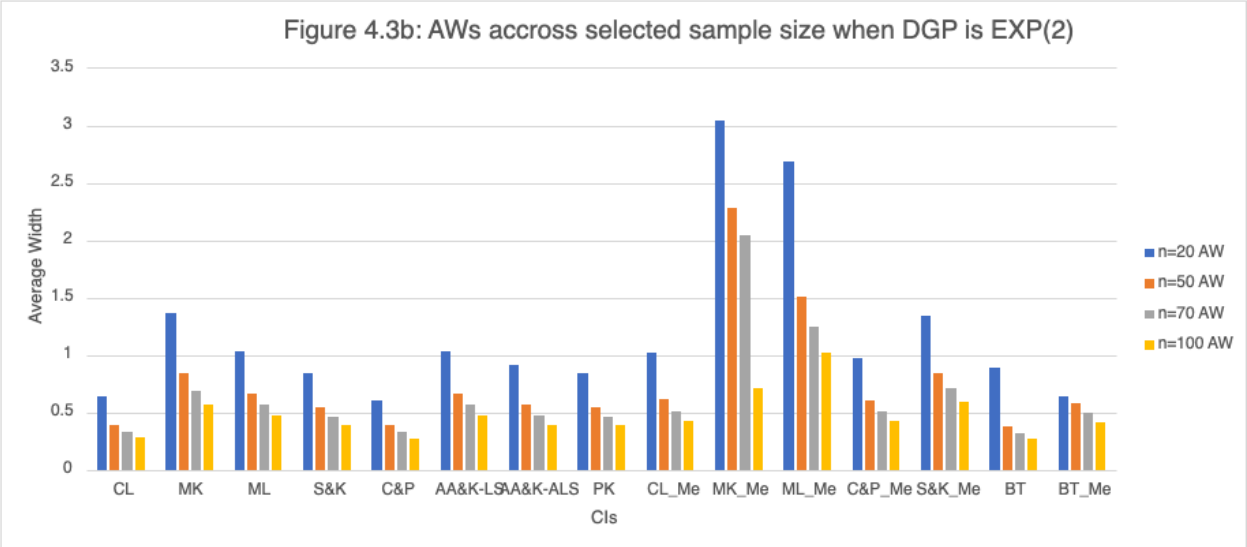


Figure 4.3b: AWs accross selected sample size when DGP is EXP(2)
Average Width
3.5
3
2.5
2
1.5
1
0.5
0
CL
MK
ML
S&K
C&P
AA&K-LS
AA&K-ALS
PK
CL_Me
MK_Me
ML_Me
C&P_Me
S&K_Me
BT
BT_Me
CIs
n=20 AW
n=50 AW
n=70 AW
n=100 AW

From Table 4.3 and the figures 4.3, a similar overall pattern can still be observed, as the sample size increases, the average width (AW) decreases for all methods, indicating improved precision. For example, under the CCL method, AW decreases from 0.6348 at n = 20 to 0.2779 at n = 100. However, unlike the previous cases, the coverage probabilities (CP) do not consistently move toward the nominal level. In fact, several methods show clear under-coverage or instability, even for larger samples. Among the classical methods, ML, MK, and S&K perform relatively well in terms of coverage. For instance, ML maintains CP values around 0.97 across all sample sizes (0.9476 to 0.9730), while MK improves from 0.9334 to 0.9750 as n increases. S&K also provides stable coverage close to 0.95 (e.g., 0.9484 at n = 20 and 0.9504 at n = 100), with moderate interval widths compared to MK and ML. On the other hand, CCL and C&P clearly underperform, with CP values consistently around 0.85, indicating under-coverage despite having relatively narrower intervals (e.g., CCL AW = 0.2779 at n = 100). This suggests that these methods are too compacted and fail to capture the true parameter often enough. Among the other methods, AA&K-ALS again shows very high coverage, especially at smaller sample sizes (0.9784 at n = 20), and remains above 0.95 even at n = 100 (0.9560). However, its interval widths are moderately large (e.g., 0.9131 at n = 20), meaning that this high reliability comes with reduced precision. The PK method provides a good balance, with CP improving from 0.9230 to 0.9486 and interval widths that are smaller than MK and ML but still reasonable. The median-based methods behave very inconsistently in this case. Some methods such as MK_Me and ML_Me show extremely high coverage (even close to 1, e.g., MK_Me = 0.9996 at n = 70), but this comes at the cost of extremely wide intervals (AW = 2.0373), making them impractical. On the other hand, methods like CCL_Me, C&P_Me, and S&K_Me perform very poorly, with coverage dropping drastically as sample size increases (e.g., CCL_Me falls to 0.1018 at n = 100).

This indicates that these methods fail to produce reliable intervals in this scenario. For the bootstrap methods, BT shows poor performance at small sample sizes (CP = 0.3836 at n = 20), although it improves for larger samples (around 0.85). However, it still remains below the desired level, indicating under-coverage. The BT_Me method is highly unstable, starting reasonably at n = 20 (0.8750) but dropping sharply to 0.0724 at n = 100. This suggests that the bootstrap median-based approach is not reliable in this setting. Overall, this table shows a stronger trade-off and more instability compared to the previous results. While methods like ML, MK, S&K, and AA&K-ALS maintain good coverage, others, especially several median-based and bootstrap variants perform poorly.

In summary:

Best coverage performance: ML, MK, S&K, and AA&K-ALS

Most balanced methods: ML and PK

Under-performing methods: CCL, C&P, and most median-based variants

Highly unstable methods: CCL_Me, C&P_Me, S&K_Me, and BT_Me

Effect of sample size: Improves precision (narrower intervals) but does not guarantee good coverage for all methods

These results suggest that method selection becomes even more important in this scenario, as some approaches fail to provide reliable confidence intervals even with larger sample sizes.

Table 4.4: CP and AW of selected CIs when DGP Beta(3, 1) with skewness -0.85224

CIs	n=20		n=50		n=70		n=100	
	CP	AW	CP	AW	CP	AW	CP	AW
CCL	0.8910	0.1819	0.9004	0.1107	0.9004	0.0925	0.8976	0.0771
MK	0.8564	0.1704	0.8874	0.1081	0.8882	0.0910	0.8894	0.0763
ML	0.8764	0.1703	0.8934	0.1079	0.8942	0.0909	0.8934	0.0762
S&K	0.9362	0.2248	0.9570	0.1430	0.9594	0.1206	0.9624	0.1011
C&P	0.8640	0.1619	0.8722	0.1019	0.8728	0.0857	0.8748	0.0717
AA&K-LS	0.8764	0.1703	0.8934	0.1079	0.8942	0.0909	0.8934	0.0762
AA&K-ALS	0.9648	0.2442	0.9724	0.1478	0.9688	0.1234	0.9690	0.1028
PK	0.8672	0.1648	0.8814	0.1046	0.8840	0.0881	0.8834	0.0739
CCL_Me	0.8882	0.1776	0.8898	0.1074	0.8768	0.0896	0.8718	0.0746
MK_Me	0.8490	0.1663	0.8690	0.1048	0.8638	0.0882	0.8590	0.0738
ML_Me	0.8716	0.1663	0.8820	0.1047	0.8718	0.0881	0.8670	0.0737
C&P_Me	0.8702	0.1586	0.8702	0.0992	0.8618	0.0833	0.8538	0.0697
S&K_Me	0.9332	0.2203	0.9512	0.1392	0.9486	0.1172	0.9470	0.0982
BT	0.8002	0.2417	0.8814	0.1360	0.8940	0.1128	0.9098	0.0936
BT_Me	0.8080	0.2272	0.8484	0.1258	0.8550	0.1040	0.8598	0.0859

Table 4.4 analyses 95% CP and AW of 15 CI methods for the CV parameter when DGP follows a Beta(3,1) distribution with skewness -0.85224. The performance of 15 different CIs is examined across selected n.

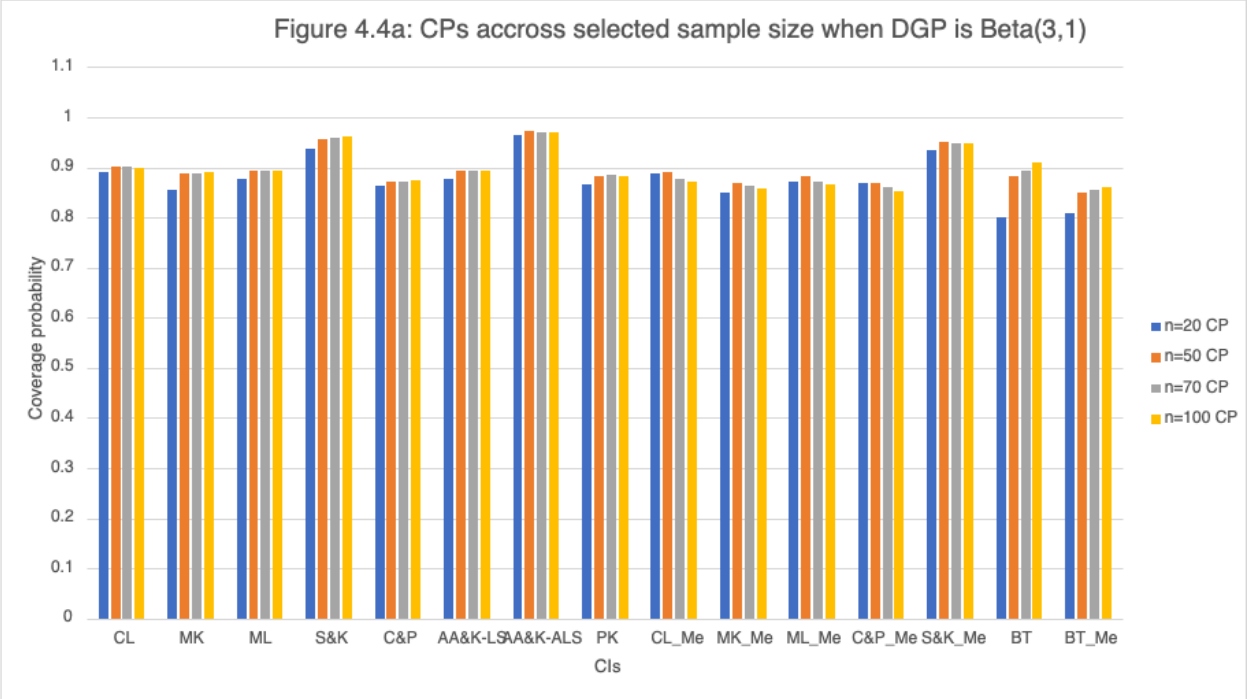


Figure 4.4a: CPs accross selected sample size when DGP is Beta(3,1)
Coverage probability
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
CL
MK
ML
S&K
C&P
AA&K-LS
AA&K-ALS
PK
CL_Me
MK_Me
ML_Me
C&P_Me
S&K_Me
BT
BT_Me
CIs
n=20 CP
n=50 CP
n=70 CP
n=100 CP


Figure 4.4b: AWs accross selected sample size when DGP is Beta(3,1)
Average Width
0.3
0.25
0.2
0.15
0.1
0.05
0
CL
MK
ML
S&K
C&P
AA&K-LS
AA&K-ALS
PK
CL_Me
MK_Me
ML_Me
C&P_Me
S&K_Me
BT
BT_Me
CIs
n=20 AW
n=50 AW
n=70 AW
n=100 AW

From Table 4.4 and the corresponding figures, a consistent pattern can again be observed, as the sample size increases, the average width (AW) decreases for all methods, indicating improved precision of the confidence intervals. For example, under the CCL method, AW decreases from 0.1819 at n = 20 to 0.0771 at n = 100. Similar reductions are seen across all methods. However, in terms of coverage probability (CP), most methods remain slightly below the nominal level, suggesting mild under-coverage even for larger samples. Among the classical methods, S&K clearly performs the best in terms of coverage, with CP increasing from 0.9362 at n = 20 to 0.9624 at n = 100. This indicates that it is more reliable in capturing the true parameter. However, its interval widths are larger compared to others (e.g., 0.2248 at n = 20 and 0.1011 at n = 100), meaning this reliability comes at the cost of reduced precision. In contrast, MK and ML show similar and more moderate behavior, with CP values improving slightly with sample size (around 0.89 for larger samples) and relatively smaller widths (around 0.076 at n = 100). This suggests they provide a better balance, although their coverage is somewhat below the desired level. The CCL method also behaves similarly, with CP close to 0.90 but slightly lower for larger samples (0.8976 at n = 100), while maintaining relatively narrow intervals. The C&P method consistently shows lower coverage (around 0.86–0.87), indicating under-coverage despite having one of the smallest interval widths. Among the modified methods, AA&K-ALS again stands out with the highest coverage probabilities, for example, 0.9724 at n = 50 and 0.9690 at n = 100. This shows strong reliability, but its interval widths are relatively larger (e.g., 0.2442 at n = 20), indicating that higher coverage is achieved at the cost of wider intervals. The PK method shows moderate performance, with CP values around 0.88–0.88 for larger samples and relatively narrow widths, but it does not reach the desired coverage level. The median-based methods

mostly show lower coverage compared to their original versions. For example, MK_Me and ML_Me remain below 0.87–0.88 even at larger sample sizes, while C&P_Me drops further to 0.8538 at n = 100. However, S&K_Me performs relatively well, maintaining CP close to 0.95 (0.9470 at n = 100), though with wider intervals similar to the original S&K method. This suggests that median-based adjustments do not always improve performance and may even reduce coverage in some cases. For the bootstrap methods, BT shows low coverage at small sample sizes (0.8002 at n = 20) but improves steadily as n increases, reaching 0.9098 at n = 100. However, it still remains slightly below the nominal level. The BT_Me method performs worse, with CP values remaining below 0.86 even for larger samples. Both methods produce relatively moderate interval widths, but their lower coverage suggests they may underestimate variability.

Overall, this table again highlights the trade-off between coverage probability and interval width:

Best coverage: AA&K-ALS and S&K

Most balanced: CCL, MK, and ML (but slightly under-covering)

Under-performing methods: C&P, PK, and most median-based variants

Bootstrap methods: improve with sample size but still underperform in coverage

In conclusion, increasing sample size improves precision (narrower intervals), but many methods still fail to achieve the desired coverage level.

Summary and Recommendations from the Simulation Results

This simulation study provides a comparison of fifteen confidence interval methods across four distributions: N(2,1), N(4,1), Exponential(2), and Beta(3,1). These include both symmetric and skewed cases. Overall, a consistent pattern is observed across all tables, as the sample size increases, the average width (AW) decreases and the coverage probability (CP) becomes more stable. This means the intervals become more precise and reliable with larger samples, although the extent of improvement varies depending on the method and distribution.

For the normal distributions such as N(2,1) and N(4,1), most classical methods including CCL, S&K, MK, ML, PK, AA&K-LS and AA&K-ALS perform well. Methods like S&K and AA&K-LS achieve very high coverage, however their intervals are wider. Methods such as CCL and C&P give narrower intervals, but their coverage is slightly below the target level. MK and ML provide a balanced performance, with moderate width and reasonably good coverage, making them more stable across different sample sizes.

For skewed distributions, the performance of several methods becomes less stable. Some methods, particularly MK and AA&K-ALS give very wide intervals, while others fail to achieve good coverage. On the other hand, methods such as CCL and C&P often exhibit under-coverage despite having narrow intervals, suggesting that they may underestimate variability under skewed conditions.In some cases, ML also shows slight instability in maintaining the nominal coverage level when the distribution is highly asymmetric.

Median-based methods generally show improved robustness under skewed distributions. Methods such as S&K_Me and MK_Me often provide improved or comparable coverage relative

to their classical versions, particularly for moderate and large sample sizes. However, some variants, such as CCL_Me and C&P_Me, may become unstable depending on the distribution, sometimes resulting in either excessive interval widths or significant under-coverage.

Bootstrap methods (BT and BT_Me) generally give narrower intervals and perform reasonably well across all scenarios. However, they may have lower coverage for small sample sizes, their performance improves as sample size increases, making them more reliable for moderate to large samples. Overall, bootstrap approaches offer a good balance between precision and efficiency and are particularly useful when distributional assumptions are difficult to justify.

Based on the overall simulation results, the following recommendations can be made:

Best-performing methods (balanced coverage and interval width): MK, ML, PK, and S&K_Me

Highly reliable but conservative methods (high coverage, wider intervals): S&K and AA&K-ALS

Efficient but slightly under-covering methods: CCL and C&P

Robust methods under skewness: MK_Me, ML_Me, and bootstrap methods (BT, BT_Me)

Methods requiring caution due to instability or inconsistency: CCL_Me, C&P_Me, and some median-based variants under extreme skewness

In conclusion, no single method is best in all situations. Classical methods work well for normal data, while median-based and bootstrap methods are more suitable for skewed data.Among all methods, MK, ML, PK, and bootstrap-based approaches provide the most balanced and practically reliable performance across different distributions and sample sizes.

5. Some Real-Life Examples

Incorporating real-life examples alongside simulation studies is valuable for multiple reasons. Such examples help to contextualize theoretical concepts, making them more understandable and relatable to practical situations. They also serve as a means to validate the simulation models and illustrate the applicability of the findings in real-world decision-making. Additionally, these examples can reveal the limitations of simulation studies, highlighting scenarios where models may not fully capture the complexities of practical applications. In this study, we consider three real-life examples, demonstrating how combining simulations with practical cases provides a more comprehensive understanding and facilitates the effective application of results. The datasets used in this exercise are simulated (hypothetical) datasets constructed to demonstrate positively skewed, negatively skewed, and normally distributed data. The examples are based on real-world phenomena commonly reported in business and health science literature.

5.1 Health Science Data

The daily number of confirmed COVID-19 cases observed in a city during a surge period is reported by the Health Department

10, 15, 25, 60, 150, 320, 450, 610, 405, 280, 190, 135, 100, 75, 50.

This dataset is positively skewed and reflects the rapid escalation and gradual decline typically associated with epidemic waves.

A descriptive statistical analysis is conducted to summarize the key characteristics of the data. The sample consists of 15 daily observations. The mean number of confirmed cases is 191.67, while the standard deviation is 182.81, indicating dispersion in daily case counts. The estimated skewness is 1.067, confirming a high positive skewness in the distribution. This suggests the presence of several unusually high daily counts that extend the right tail and increase the sample mean. The coefficient of variation (CV) is computed as 95.38%, indicating that the variation in the data is nearly as large as the mean itself. Such a high CV is common in epidemic data during surge periods, where case numbers can increase rapidly within short time intervals.

The combination of non-negative values, strong right-skewness, and high relative variability indicates that the daily COVID-19 case counts do follow a normal distribution. Instead, a positively skewed exponential distribution provides a better probabilistic model for this dataset. Notably, the observed sample CV is close to the theoretical CV of unity associated with the exponential distribution, lending further support to this modeling choice. Table 5.1 presents the estimated confidence intervals for the coefficient of variation along with their corresponding confidence widths, providing insight into the degree of uncertainty associated with the estimates. Figure 5.1 illustrates the distributional shape of the data and offers a visual assessment of the adequacy of the model fit.

Table 5.1: CIs for selected CVs and corresponding class width

Method	Confidence Interval	Class width
CCL	(0.6125, 1.2951)	0.6826
MK	(0.6983, 1.5042)	0.8059
ML	(0.4711, 1.4365)	0.9654
S&K	(0.5750, 1.5821)	1.0071
C&P	(0.6333, 1.9310)	1.2977
AA&K-LS	(0.3807, 1.5269)	1.1462
AA&K-ALS	(0.2868, 1.6208)	1.3341
PK	(0.4740, 1.9195)	1.4455
CCL_Me	(0.5232, 1.1064)	0.5831
MK_Me	(0.5965, 1.2850)	0.6885
ML_Me	(0.4025, 1.2272)	0.8247
C&P_Me	(0.5410, 1.6496)	1.1086
S&K_Me	(0.4912, 1.3516)	0.8603
BT	(0.6400, 1.2676)	0.6277
BT_Me	(0.4884, 1.1412)	0.6528

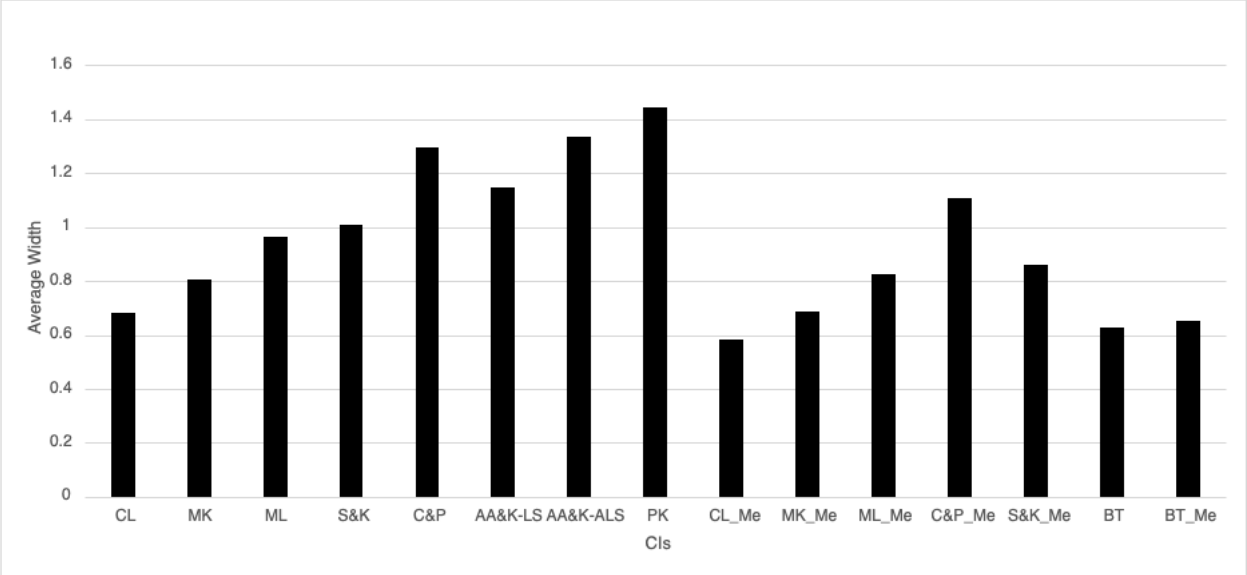


Figure 5.1: Class width for selected CIs

Table 5.1 compares confidence intervals for the coefficient of variation (CV) using different methods, along with their corresponding class widths. The results show noticeable differences in interval widths, which reflect the precision of each method. Methods such as CCL, CCL_Me, MK_Me, BT, and BT_Me produce relatively narrow intervals, with class widths ranging from about 0.58 to 0.80. This indicates higher precision and lower uncertainty. In contrast, C&P, PK, AA&K-LS, and AA&K-ALS produce wider intervals,with widths exceeding 1.10 in several cases. The PK method gives the widest interval (1.4455)**,** suggesting greater uncertainty in estimation. ML and S&K show moderate interval widths with class widths of 0.9654 and 1.0071 respectively, representing a balance between narrow and wide intervals. The bootstrap methods show relatively good performance. Both BT and BT_Me produce comparatively narrow intervals, suggesting that bootstrap approaches can achieve good precision while maintaining reasonable stability.

The key findings are:

Narrowest intervals (highest precision): CCL_Me, BT, MK_Me, CCL, BT_Me

Widest intervals (lowest precision): AA&K-ALS, PK, C&P, C&P_Me

Most unstable method: PK (widest interval)

In general, the results highlight a trade-off between precision and uncertainty, where narrower intervals give more precise estimates but wider intervals reflect more cautious estimation.

5.2 Business Data

The scores obtained in an easy, company-wide compliance test are reported in Company Records (2025) as

98, 100, 95, 99, 100, 97, 100, 42, 96, 100.

These data reflect performance on a mandatory assessment designed to be straightforward, with most employees expected to achieve near-perfect scores.

A descriptive statistical analysis is performed to summarize the main features of the dataset. The deviation is 17.91, indicating noticeable dispersion in scores despite the overall high performance. The estimated skewness is −3.10, revealing strong negative skewness in the distribution. This pronounced left-skewness arises from a small number of unusually low scores,

most notably a single extreme observation, while the majority of employees scored at or near the maximum. Consequently, the mean score (92.70) is pulled downward relative to the median score (98.50), and the mode is located at the upper boundary of the scale (100), reflecting a concentration of high achievers.

The coefficient of variation (CV) is 19.32%, sample consists of 10 observations, with an average (mean) test score of 92.70. The standard indicates moderate relative variability in test performance. This variability is not representative of widespread inconsistency among employees but is primarily driven by the presence of a low outlying score in an otherwise tightly clustered set of high values. The strong negative skewness and bounded nature of the data (with an upper limit of 100) clearly violate the assumptions of normality. Therefore, a left-skewed distribution with bounded support, such as a reflected Beta distribution, provides a more appropriate modeling framework than the normal distribution.

Table 5.2 reports the estimated confidence intervals for the coefficient of variation along with their corresponding confidence widths, while Figure 5.2 shows a graph that helps us understand the shape of the data and how much it varies.

Table 5.2: CIs for selected CVs and corresponding class width

Method	Confidence Interval	Class width
CCL	(0.1085, 0.2778)	0.1693
MK	(0.1329, 0.3527)	0.2198
ML	(0.0734, 0.3129)	0.2395
S&K	(0.1039, 0.3590)	0.2551
C&P	(0.1193, 0.5081)	0.3888
AAK-LS	(0.1054, 0.2809)	0.1755
AAK-ALS	(0.0712, 0.3151)	0.2439
PK	(0.1028, 0.3632)	0.2604
CCL_Me	(0.0086, 0.0219)	0.0133
MK_Me	(0.0105, 0.0278)	0.0173
ML_Me	(0.0058, 0.0247)	0.0189
C&P_Me	(0.0094, 0.0401)	0.0307
S&K_Me	(0.0082, 0.0283)	0.0201
BT	(0.0000, 0.4104)	0.4104
BT_Me	(0.0000, 0.0563)	0.0563

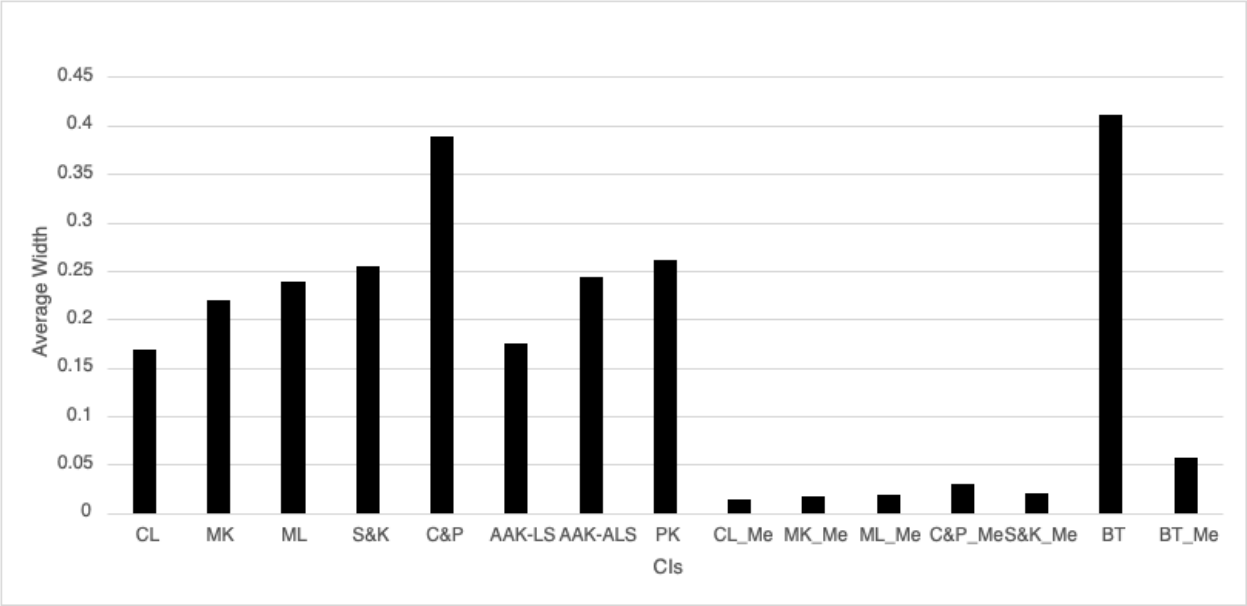


Figure 5.2: Class width for selected CIs

Table 5.2 presents the confidence intervals for the coefficient of variation (CV) along with their class widths for different methods. The results show clear differences in the precision of the methods. Among the classical methods, CCL and AAK-LS produce relatively narrow intervals (0.1693 and 0.1755), indicating higher precision. In contrast, C&P, MK, ML, S&K, and PK show wider intervals ranging from 0.21 to 0.38, with C&P having one of the widest ranges (0.3888), suggesting lower precision and higher uncertainty. The modified or median-based methods generally produce much smaller interval values, especially CCL_Me, MK_Me, ML_Me, and S&K_Me. These methods show very narrow intervals ranging from 0.01 to 0.02, indicating high precision in estimation. However, C&P_Me is relatively wider compared to other median-based methods, showing slightly higher uncertainty. For the bootstrap methods, the results vary significantly. BT produces a very wide interval (0.4104), indicating high variability and low

precision. On the other hand, BT_Me gives a much narrower interval (0.0563) compared to BT, showing improved performance under the median-based adjustment.

The key observations are:

Narrowest intervals (highest precision): CCL_Me, MK_Me, S&K_Me, ML_Me

Wider intervals (lower precision): C&P, PK, S&K, ML

Most unstable method: BT (very wide interval)

Overall, the results show a clear difference in precision across methods. Median-based methods generally produce the narrowest confidence intervals, while some classical and bootstrap methods show wider intervals, indicating greater uncertainty in estimating the coefficient of variation.

5.3 Manufacturing Data

The height of products from a manufacturing line, measured in milli meters, is represented by the following data:

99.8, 100.1, 100.2, 99.9, 100.0, 100.3, 99.7, 100.0, 99.8, 100.1.

We analyse the heights of products produced on a manufacturing line to summarise the key statistical characteristics of the data. The sample size is 10. The average product height is approximately 99.99 mm, which is very close to both the median(100 mm) and the mode(99.8 mm), indicating symmetry in the data. The standard deviation is 0.1912, reflecting minimal variability around the target height. The skewness(0.0572) is close to zero, suggesting that the distribution of heights is approximately normal with no noticeable skewness. The coefficient of variation (CV) is 0.1912% which is very low, indicating high consistency and precision in the manufacturing process.

The product height measurements exhibit negligible skewness (0.0572) and very low variability (CV = 0.1912%), indicating that the data closely follow a normal distribution. The symmetry of the data, together with the near equality of the mean, median, and mode, supports the assumption of normality for this manufacturing process. The small coefficient of variation reflects high process precision and stability.

Table 5.3 presents the confidence intervals for the coefficient of variation along with their corresponding class widths for this dataset, and Figure 5.3 provides a graphical illustration for clearer comparison.

Table 5.3: CIs for selected CVs and corresponding class width

Method	Confidence Interval	Class width
CCL	(0.0011, 0.0028)	0.0017
MK	(0.0013, 0.0035)	0.0022
ML	(0.0007, 0.0031)	0.0024
S&K	(0.0010, 0.0036)	0.0025
C&P	(0.0012, 0.0050)	0.0038
AAK-LS	(0.0011, 0.0028)	0.0017
AAK-ALS	(0.0007, 0.0031)	0.0024
PK	(0.0010, 0.0036)	0.0025
CCL_Me	(0.0008, 0.0022)	0.0013
MK_Me	(0.0010, 0.0027)	0.0017
ML_Me	(0.0006, 0.0024)	0.0019
C&P_Me	(0.0009, 0.0039)	0.0030
S&K_Me	(0.0008, 0.0028)	0.0020
BT	(0.0013, 0.0026)	0.0013
BT_Me	(0.0006, 0.0024)	0.0019

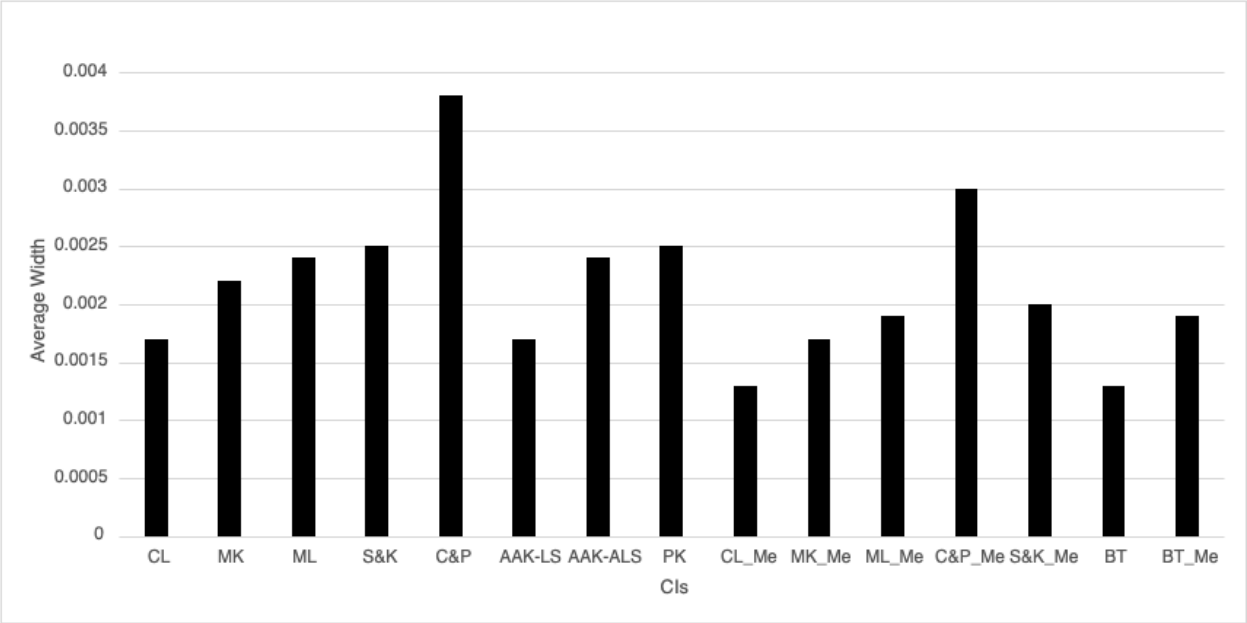


Figure 5.3: Class width for selected CIs

Table 5.3 presents the confidence intervals for the coefficient of variation (CV) along with their class widths for different methods. The results show clear differences in the precision of the estimators. Among the classical methods, CCL and AAK-LS produce relatively narrow intervals (both 0.0017) with small class widths, indicating higher precision. In contrast, C&P shows the widest interval among classical methods (0.0038), suggesting greater uncertainty compared to other approaches. MK, ML, S&K, PK, and AAK-ALS fall in between, showing moderate interval widths ranging from 0.0022 to 0.0025 and balanced performance. For the median-based methods, CCL_Me, MK_Me, ML_Me, and S&K_Me generally produce narrower intervals ranging from 0.0013 to 0.002, indicating improved precision compared to their classical versions. However, C&P_Me remains relatively wide (0.0030), showing less improvement and higher variability. The bootstrap methods show mixed results. BT produces a narrow interval(0.0013), similar to CCL_Me, indicating good precision. BT_Me also performs

reasonably well with a moderate interval width(0.0019), showing improvement compared to some classical methods.

The key observations are:

Narrowest intervals (highest precision): CCL_Me, BT, CCL, AAK-LS

Wider intervals (lower precision): C&P and C&P_Me

Most unstable method: C&P (widest interval)

In conclusion, the results show that most methods produce very small confidence intervals, indicating high precision in estimating the coefficient of variation. Median-based and bootstrap methods generally maintain or improve precision, while C&P-based methods show relatively wider intervals, indicating higher uncertainty compared to other approaches.

6. Conclusion

This study demonstrates that the performance of confidence interval estimators for the coefficient of variation (CV) is strongly influenced by distributional characteristics, including shape, bounded support, skewness, outliers and sample size. Through an extensive Monte Carlo simulation framework and real-life data applications, the performance of fifteen classical, modified, median-based, and bootstrap CI methods was systematically assessed using coverage probability (CP) and average interval width (AW) as the primary criteria. Under normal distributions, most classical methods perform satisfactorily. Methods such as MK, ML, and PK provide a good balance between coverage probability (CP) and average width (AW), while S&K and AA&K-ALS achieve higher coverage at the cost of wider intervals. However, under skewed distributions, several methods exhibit under-coverage or instability. In such cases, ML, MK, S&K, and AA&K-ALS remain relatively reliable, whereas methods like CCL and C&P perform poorly. Median-based methods show mixed results, offering some robustness but lacking consistency, particularly under extreme skewness. Bootstrap methods provide flexible, distribution-free alternatives with relatively narrow intervals, but may suffer from under-coverage in small samples. Bootstrap-based approaches provide a flexible alternative, particularly in the absence of strict distributional assumptions. These methods generally yield narrower intervals and show improved performance as sample size increases. Nevertheless, their tendency toward under-coverage in small samples suggests that caution is required when applying them in limited data settings. The real-life examples further validate the simulation findings, highlighting that method performance varies with data characteristics. In skewed datasets, such as health and business data, method performance varies substantially, reinforcing the importance of selecting appropriate CI procedures based on data characteristics. In contrast,

under near-normal conditions, such as manufacturing data, most methods perform consistently well, reflecting the robustness of classical approaches in such settings. Overall, no single method is universally optimal. Instead, the choice of CI estimator should be guided by the distributional properties of the data and the trade-off between accuracy (coverage probability) and precision (average/interval width).

Limitations and Future Work

Despite providing a comprehensive comparative analysis, this study is subject to several limitations that offer opportunities for further research. First, the analysis is restricted to a limited set of distributions, namely normal, exponential, and beta distributions. While these cover both symmetric and skewed cases, they may not capture all real-world data complexities. Future research should consider a wider range of distributions, including heavy-tailed, multimodal and mixed models. Future studies could extend this analysis to a broader class of distributions, including log-normal, gamma, Weibull, and contaminated distributions, to further evaluate robustness. Second, only selected sample sizes were examined (n = 20, 50, 70, and 100). Further work could explore very small and large sample behavior which would enhance the generalizability of the findings. Third, the evaluation relies on CP and AW; additional measures such as interval symmetry, bias, mean squared error of interval bounds, and computational efficiency could provide a more comprehensive assessment. Fourth, only the bootstrap-t method was considered. Other advanced bootstrap techniques, such as bias-corrected and accelerated (BCa) intervals or percentile-based refinements, were not considered. Exploring these alternatives may yield improved performance, particularly under skewed or small-sample conditions. Fifth, median-based methods showed instability in some cases, indicating the need

for further methodological refinement to develop more stable and theoretically grounded robust CI estimators for the CV, particularly under extreme skewness and the presence of outliers. Finally, the real-life examples, while illustrative, are based on relatively small datasets. Future research could include larger and more diverse datasets from different fields to validate the empirical findings and strengthen practical recommendations.

In future research, it would also be valuable to develop adaptive or data-driven methods that can automatically select the most appropriate CI procedure based on sample characteristics such as skewness, kurtosis, and sample size. Such approaches could significantly enhance the practical applicability of CI estimation for the coefficient of variation in complex real-world settings.

References

[1] A. T. McKay, “Distribution of the coefficient of variation and the extended t distribution,” *Journal of the Royal Statistical Society*, vol. 95, no. 4, pp. 695–698, 1932.

[2] W. A. Hendricks and K. W. Robey, “The sampling distribution of the coefficient of variation,” *Annals of Mathematical Statistics*, vol. 7, no. 3, pp. 129–132, 1936.

[3] M. G. Vangel, “Confidence intervals for a normal coefficient of variation,” *The American Statistician*, vol. 50, no. 1, pp. 21–26, 1996.

[4] E. G. Miller, “Asymptotic test statistics for coefficient of variation,” *Communication in Statistics - Theory and Methods*, vol. 20, no. 10, pp. 3351–3363, 1991.

[5]K. Krishnamoorthy and M. Lee, “Improved tests for the coefficient of variation,” *Computational Statistics*, vol. 29, no. 1–2, pp. 215–232, 2014.

[6] J. D. Curto and J. C. Pinto,“The coefficient of variation asymptotic distribution in the case of non-iid random variables,” *Journal of Applied Statistics,* vol. 36, no. 1, pp. 21–32, 2009.

[7] K. Panichkitkosolkul,“Confidence intervals for coefficient of variation in a normal distribution with a known population mean,” *Journal of Applied Statistics*, vol. 36, no. 4, pp. 459–468, 2009.

[8] S. Banik and B. M. G. Kibria, “A simulation study on confidence intervals for the coefficient of variation,” *Communications in Statistics – Simulation and Computation*, vol. 40, no. 8, pp. 1233–1252, 2011.

[9]S. Banik, B. M. G. Kibria, and R. Gulhar, “Bootstrap confidence intervals for the coefficient of variation,” *Journal of Applied Statistics*, vol. 39, no. 5, pp. 1033–1045, 2012.

[10] M. Abu-Shawiesh, F. Akyüz, and B. M. G. Kibria, “Large-sample and modified confidence intervals for the coefficient of variation,” *Journal of Statistical Computation and Simulation*, vol. 89, no. 10, pp. 1900–1917, 2019.

[11]B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall/CRC, 1993.

[12]K. Krishnamoorthy and M. Lee, “Improved confidence intervals for the coefficient of variation,” *Computational Statistics*, vol. 29, no. 1–2, pp. 335–351, 2014.

[13] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and Their Application*. Cambridge, U.K.: Cambridge University Press, 1997.

[14] J. Q. Shi and B. M. G. Kibria, “Median-based confidence intervals for skewed distributions,” *Communications in Statistics – Theory and Methods*, vol. 36, no. 5, pp. 825–841, 2007.

[15] B. M. G. Kibria, “On robust measures of variation under skewed distributions,” *Statistical Methodology*, vol. 3, no. 2, pp. 121–135, 2006.

[16] R. Gulhar, S. Banik, and B. M. G. Kibria, “Robust median-based intervals for the coefficient of variation,” *Journal of Applied Statistics*, vol. 39, no. 10, pp. 2231–2247, 2012.

[17] S. Sharma and H. Krishna, “Asymptotic confidence intervals for the coefficient of variation,” *Journal of Statistical Planning and Inference*, vol. 41, no. 3, pp. 345–357, 1994.

[18] K. K. Sharma and H. Krishna, “On the coefficient of variation and its applications,” *Communications in Statistics – Theory and Methods*, vol. 23, no. 3, pp. 787–795, 1994.

[19] B. M. G. Kibria, “Some new confidence intervals for the coefficient of variation,” *Statistical Methodology*, vol. 3, no. 1, pp. 42–51, 2006.

[20] J. M. Koopmans, D. B. Owen, and J. M. Rhoades, “Confidence intervals for the coefficient of variation,” *Biometrika*, vol. 51, no. 1–2, pp. 25–32, 1964.

[21] I. Iglewicz, “On estimating the coefficient of variation,” *Technometrics*, vol. 9, no. 3, pp. 515–518, 1967.

[22] K. Krishnamoorthy and H. Lee, “Confidence interval for the coefficient of variation in normal and non-normal distributions,” *Communications in Statistics – Simulation and Computation,* vol. 43, no. 2, pp. 423–438, 2014.

[23] J. Shao and D. Tu, *The Jackknife and Bootstrap*. New York, NY, USA: Springer, 1995.

[24] B. Efron, “Bootstrap methods: another look at the jackknife,” *Annals of Statistics,* vol. 7, no. 1, pp. 1–26, 1979.

APPENDIX A

Simulation Algorithms and MATLAB Implementation

This appendix presents the Monte Carlo simulation algorithms and MATLAB codes used to evaluate the performance of various confidence interval (CI) methods for the coefficient of variation (CV) and the implementation used for analyzing real-life datasets. The performance measures considered are coverage probability (CP) and average width (AW). All simulations were conducted using MATLAB.

Appendix A.1

Simulation Framework

The following simulation framework was used for all distributions. The sample sizes considered were

$$n=20,\ 50,\ 70,\ 100$$

with 5000 Monte Carlo replications and a 95% confidence level.

Simulation Code for Normal Distribution:

Case 1: N(2,1) and N(4,1)

```
clc; clear;
rng(1); % For reproducibility
```

```
% Parameters
n_values = [20, 50, 70, 100];
%for N(2,1)
mu = 2;
sigma = 1;
%for N(4,1)
mu=4;
sigma=1;
true_CV = sigma / mu;

alpha = 0.05;
replications = 5000;
numBootstraps = 1000;

z = norminv(1 - alpha/2);

% Loop over sample sizes
for ni = 1:length(n_values)

    n = n_values(ni);
    tcri = tinv(1 - alpha/2, n-1);

    chi2_lower = chi2inv(alpha/2, n-1);
    chi2_upper = chi2inv(1 - alpha/2, n-1);

    % Storage
    methods = 15;
    CP = zeros(methods,1);
    AW = zeros(methods,1);

    coverage = zeros(replications,methods);
    width = zeros(replications,methods);

    for i = 1:replications

        % Generate data
        sample = normrnd(mu, sigma, [n,1]);

        xbar = mean(sample);
        s = std(sample);
        CV = s / xbar;

        % Median version
        med = median(sample);
        s_med = sqrt(sum((sample - med).^2)/(n-1));
        CV_me = s_med / med;
```

```
%% ========== 1. Classical ==========
SCV = CV / sqrt(2*n);
L = CV - tcri * SCV;
U = CV + tcri * SCV;
coverage(i,1) = (true_CV >= L && true_CV <= U);
width(i,1) = U - L;

%% ========== 2. McKay ==========
L = CV * sqrt(((chi2_lower/n - 1)*CV^2) + (chi2_lower/n));
U = CV * sqrt(((chi2_upper/n - 1)*CV^2) + (chi2_upper/n));
coverage(i,2) = (true_CV >= L && true_CV <= U);
width(i,2) = U - L;

%% ========== 3. Miller ==========
L = CV - z * sqrt((CV^4 + 0.5*CV^2)/n);
U = CV + z * sqrt((CV^4 + 0.5*CV^2)/n);
coverage(i,3) = (true_CV >= L && true_CV <= U);
width(i,3) = U - L;

%% ========== 4. Sharma & Krishna ==========
L = CV - z * (CV / sqrt(n));
U = CV + z * (CV / sqrt(n));
coverage(i,4) = (true_CV >= L && true_CV <= U);
width(i,4) = U - L;

%% ========== 5. Curto & Pinto ==========
A = 1 + (1/(4*n));
B = sqrt((1/(2*n)) * (1 + 3/(4*n)));
L = CV * (A - z * B);
U = CV * (A + z * B);
coverage(i,5) = (true_CV >= L && true_CV <= U);
width(i,5) = U - L;


%% ========== 6. AA&K-LS ==========
var_CV = (CV^2 * (1 + 2*CV^2)) / (2*n);
L = CV - z * sqrt(var_CV);
U = CV + z * sqrt(var_CV);
coverage(i,6) = (true_CV >= L && true_CV <= U);
width(i,6) = U - L;

%% ========== 7. AA&K-ALS ==========
error_margin = z * sqrt((1/n) + (2/(n^2)));
L = CV * exp(-error_margin);
U = CV * exp(error_margin);
coverage(i,7) = (true_CV >= L && true_CV <= U);
width(i,7) = U - L;
```

```
%% ========== 8. Panichkitkosolkul ==========
L = CV * (1 - z * sqrt((1 + CV^2)/(2*n)));
U = CV * (1 + z * sqrt((1 + CV^2)/(2*n)));
coverage(i,8) = (true_CV >= L && true_CV <= U);
width(i,8) = U - L;

%% ========== 9. Classical (Median) ==========
SCV_me = CV_me / sqrt(2*n);
L = CV_me - tcri * SCV_me;
U = CV_me + tcri * SCV_me;
coverage(i,9) = (true_CV >= L && true_CV <= U);
width(i,9) = U - L;

%% ========== 10. McKay (Median) ==========
L = CV_me * sqrt(((chi2_lower/n - 1)*CV_me^2) + (chi2_lower/n));
U = CV_me * sqrt(((chi2_upper/n - 1)*CV_me^2) + (chi2_upper/n));
coverage(i,10) = (true_CV >= L && true_CV <= U);
width(i,10) = U - L;

%% ========== 11. Miller (Median) ==========
L = CV_me - z * sqrt((CV_me^4 + 0.5*CV_me^2)/n);
U = CV_me + z * sqrt((CV_me^4 + 0.5*CV_me^2)/n);
coverage(i,11) = (true_CV >= L && true_CV <= U);
width(i,11) = U - L;

%% ========== 12. Curto & Pinto (Median) ==========
L = CV_me * (A - z * B);
U = CV_me * (A + z * B);
coverage(i,12) = (true_CV >= L && true_CV <= U);
width(i,12) = U - L;

%% ========== 13. Sharma & Krishna (Median) ==========
L = CV_me - z * (CV_me / sqrt(n));
U = CV_me + z * (CV_me / sqrt(n));
coverage(i,13) = (true_CV >= L && true_CV <= U);
width(i,13) = U - L;

%% ========== 14. Bootstrap-t ==========
bootCV = zeros(numBootstraps,1);
for b = 1:numBootstraps
   bs = datasample(sample, n);
   bootCV(b) = std(bs)/mean(bs);
end

tboot = (bootCV - mean(bootCV)) / std(bootCV);
t_sorted = sort(tboot);

SCV = CV / sqrt(2*n);
```

```
        L = CV + t_sorted(round(alpha/2*numBootstraps)) * SCV;
        U = CV + t_sorted(round((1-alpha/2)*numBootstraps)) * SCV;

        coverage(i,14) = (true_CV >= L && true_CV <= U);
        width(i,14) = U - L;

        %% ========== 15. Bootstrap-t (Median) ==========
        bootCV = zeros(numBootstraps,1);
        for b = 1:numBootstraps
            bs = datasample(sample, n);
            med_b = median(bs);
            s_b = sqrt(sum((bs - med_b).^2)/(n-1));
            bootCV(b) = s_b / med_b;
        end

        tboot = (bootCV - mean(bootCV)) / std(bootCV);
        t_sorted = sort(tboot);

        SCV_me = CV_me / sqrt(2*n);
        L = CV_me + t_sorted(round(alpha/2*numBootstraps)) * SCV_me;
        U = CV_me + t_sorted(round((1-alpha/2)*numBootstraps)) * SCV_me;

        coverage(i,15) = (true_CV >= L && true_CV <= U);
        width(i,15) = U - L;

    end

    % Final results
    CP = mean(coverage);
    AW = mean(width);

    fprintf('\n===== n = %d =====\n', n);
    for k = 1:15
        fprintf('Method %d -> CP: %.4f  AW: %.4f\n', k, CP(k), AW(k));
    end

end
```

Simulation Code for Exponential Distribution:

Case 2 : EXP(2)

```
clc; clear;
rng(1); % For reproducibility
% Parameters
n_values = [20, 50, 70, 100];
lambda = 2;
true_CV = 1;
alpha = 0.05;
replications = 5000;
numBootstraps = 1000;
z = norminv(1 - alpha/2);
% Loop over sample sizes
for ni = 1:length(n_values)

    n = n_values(ni);
    tcri = tinv(1 - alpha/2, n-1);

    chi2_lower = chi2inv(alpha/2, n-1);
    chi2_upper = chi2inv(1 - alpha/2, n-1);

    % Storage

    coverage = zeros(replications, 15);
    width = zeros(replications, 15);

    for i = 1:replications

        % Generate data
        sample = exprnd(1/lambda, [n,1]);

        xbar = mean(sample);
        s = std(sample);
        CV = s / xbar;

        % Median version
        med = median(sample);
        s_med = sqrt(sum((sample - med).^2)/(n-1));
        CV_me = s_med / med;

        %% ========== 1. Classical ==========
        SCV = CV / sqrt(2*n);
        L = CV - tcri * SCV;
        U = CV + tcri * SCV;
        coverage(i,1) = (true_CV >= L && true_CV <= U);
        width(i,1) = U - L;

        %% ========== 2. McKay ==========
        L = CV * sqrt(((chi2_lower/n - 1)*CV^2) + (chi2_lower/n));
        U = CV * sqrt(((chi2_upper/n - 1)*CV^2) + (chi2_upper/n));
```

```
coverage(i,2) = (true_CV >= L && true_CV <= U);
width(i,2) = U - L;

%% ========== 3. Miller ==========
L = CV - z * sqrt((CV^4 + 0.5*CV^2)/n);
U = CV + z * sqrt((CV^4 + 0.5*CV^2)/n);
coverage(i,3) = (true_CV >= L && true_CV <= U);
width(i,3) = U - L;

%% ========== 4. Sharma & Krishna ==========
L = CV - z * (CV / sqrt(n));
U = CV + z * (CV / sqrt(n));
coverage(i,4) = (true_CV >= L && true_CV <= U);
width(i,4) = U - L;

%% ========== 5. Curto & Pinto ==========
A = 1 + (1/(4*n));
B = sqrt((1/(2*n)) * (1 + 3/(4*n)));
L = CV * (A - z * B);
U = CV * (A + z * B);
coverage(i,5) = (true_CV >= L && true_CV <= U);
width(i,5) = U - L;


%% ========== 6. AA&K-LS ==========
var_CV = (CV^2 * (1 + 2*CV^2)) / (2*n);
L = CV - z * sqrt(var_CV);
U = CV + z * sqrt(var_CV);
coverage(i,6) = (true_CV >= L && true_CV <= U);
width(i,6) = U - L;

%% ========== 7. AA&K-ALS ==========
error_margin = z * sqrt((1/n) + (2/(n^2)));
L = CV * exp(-error_margin);
U = CV * exp(error_margin);
coverage(i,7) = (true_CV >= L && true_CV <= U);
width(i,7) = U - L;

%% ========== 8. Panichkitkosolkul ==========
L = CV * (1 - z * sqrt((1 + CV^2)/(2*n)));
U = CV * (1 + z * sqrt((1 + CV^2)/(2*n)));
coverage(i,8) = (true_CV >= L && true_CV <= U);
width(i,8) = U - L;

%% ========== 9. Classical (Median) ==========
SCV_me = CV_me / sqrt(2*n);
L = CV_me - tcri * SCV_me;
U = CV_me + tcri * SCV_me;
```

```
coverage(i,9) = (true_CV >= L && true_CV <= U);
width(i,9) = U - L;

%% ========== 10. McKay (Median) ==========
L = CV_me * sqrt(((chi2_lower/n - 1)*CV_me^2) + (chi2_lower/n));
U = CV_me * sqrt(((chi2_upper/n - 1)*CV_me^2) + (chi2_upper/n));
coverage(i,10) = (true_CV >= L && true_CV <= U);
width(i,10) = U - L;

%% ========== 11. Miller (Median) ==========
L = CV_me - z * sqrt((CV_me^4 + 0.5*CV_me^2)/n);
U = CV_me + z * sqrt((CV_me^4 + 0.5*CV_me^2)/n);
coverage(i,11) = (true_CV >= L && true_CV <= U);
width(i,11) = U - L;

%% ========== 12. Curto & Pinto (Median) ==========
L = CV_me * (A - z * B);
U = CV_me * (A + z * B);
coverage(i,12) = (true_CV >= L && true_CV <= U);
width(i,12) = U - L;

%% ========== 13. Sharma & Krishna (Median) ==========
L = CV_me - z * (CV_me / sqrt(n));
U = CV_me + z * (CV_me / sqrt(n));
coverage(i,13) = (true_CV >= L && true_CV <= U);
width(i,13) = U - L;

%% ========== 14. Bootstrap-t ==========
bootCV = zeros(numBootstraps,1);
for b = 1:numBootstraps
   bs = datasample(sample, n);
   bootCV(b) = std(bs)/mean(bs);
end

tboot = (bootCV - mean(bootCV)) / std(bootCV);
t_sorted = sort(tboot);

SCV = CV / sqrt(2*n);
L = CV + t_sorted(round(alpha/2*numBootstraps)) * SCV;
U = CV + t_sorted(round((1-alpha/2)*numBootstraps)) * SCV;

coverage(i,14) = (true_CV >= L && true_CV <= U);
width(i,14) = U - L;

%% ========== 15. Bootstrap-t (Median) ==========
bootCV = zeros(numBootstraps,1);
for b = 1:numBootstraps
   bs = datasample(sample, n);
```

```
        med_b = median(bs);
        s_b = sqrt(sum((bs - med_b).^2)/(n-1));
        bootCV(b) = s_b / med_b;
    end

    tboot = (bootCV - mean(bootCV)) / std(bootCV);
    t_sorted = sort(tboot);

    SCV_me = CV_me / sqrt(2*n);
    L = CV_me + t_sorted(round(alpha/2*numBootstraps)) * SCV_me;
    U = CV_me + t_sorted(round((1-alpha/2)*numBootstraps)) * SCV_me;

    coverage(i,15) = (true_CV >= L && true_CV <= U);
    width(i,15) = U - L;

  end

  % Final results
  CP = mean(coverage);
  AW = mean(width);

  fprintf('\n===== n = %d =====\n', n);
  for k = 1:15
    fprintf('Method %d -> CP: %.4f  AW: %.4f\n', k, CP(k), AW(k));
  end

end
```

Simulation Code for Beta Distribution:

Case 3 : Beta(3,1)

```
clc; clear;
rng(1); % Reproducibility
%% PARAMETERS
n_values = [20, 50, 70, 100];
a = 3; b = 1;
mu = a / (a + b);
varX = (a*b)/((a+b)^2*(a+b+1));
sigma = sqrt(varX);
true_CV = sigma / mu;
alpha = 0.05;
replications = 5000;
numBootstraps = 1000;
```

```
z = norminv(1 - alpha/2);
%% LOOP OVER SAMPLE SIZES
for ni = 1:length(n_values)

    n = n_values(ni);
    tcri = tinv(1 - alpha/2, n-1);

    chi2_lower = chi2inv(alpha/2, n-1);
    chi2_upper = chi2inv(1 - alpha/2, n-1);

    coverage = zeros(replications, 15);
    width = zeros(replications, 15);

    for i = 1:replications

        %% SAMPLE
        sample = betarnd(a, b, n, 1);

        xbar = mean(sample);
        s = std(sample);
        CV = s / xbar;

        %% MEDIAN VERSION
        med = median(sample);
        s_med = sqrt(sum((sample - med).^2)/(n-1));
        CV_me = s_med / med;

        %% ========================
        %% 1. Classical
        SCV = CV * sqrt((1 + 2*CV^2)/(2*n));
        L = CV - tcri * SCV;
        U = CV + tcri * SCV;
        coverage(i,1) = (true_CV >= L && true_CV <= U);
        width(i,1) = U - L;

        %% 2. McKay
        L = CV * sqrt(((chi2_lower/n - 1)*CV^2) + (chi2_lower/n));
        U = CV * sqrt(((chi2_upper/n - 1)*CV^2) + (chi2_upper/n));
        coverage(i,2) = (true_CV >= L && true_CV <= U);
        width(i,2) = U - L;

        %% 3. Miller
        L = CV - z * sqrt((CV^4 + 0.5*CV^2)/n);
        U = CV + z * sqrt((CV^4 + 0.5*CV^2)/n);
        coverage(i,3) = (true_CV >= L && true_CV <= U);
        width(i,3) = U - L;

        %% 4. Sharma & Krishna
```

```
L = CV - z * (CV / sqrt(n));
U = CV + z * (CV / sqrt(n));
coverage(i,4) = (true_CV >= L && true_CV <= U);
width(i,4) = U - L;

%% 5. Curto & Pinto
A = 1 + (1/(4*n));
B = sqrt((1/(2*n)) * (1 + 3/(4*n)));
L = CV * (A - z * B);
U = CV * (A + z * B);
coverage(i,5) = (true_CV >= L && true_CV <= U);
width(i,5) = U - L;

%% 6. AA&K-LS
var_CV = (CV^2 * (1 + 2*CV^2)) / (2*n);
L = CV - z * sqrt(var_CV);
U = CV + z * sqrt(var_CV);
coverage(i,6) = (true_CV >= L && true_CV <= U);
width(i,6) = U - L;

%% 7. AA&K-ALS
error_margin = z * sqrt((1/n) + (2/(n^2)));
L = CV * exp(-error_margin);
U = CV * exp(error_margin);
coverage(i,7) = (true_CV >= L && true_CV <= U);
width(i,7) = U - L;

%% 8. Panichkitkosolkul
L = CV * (1 - z * sqrt((1 + CV^2)/(2*n)));
U = CV * (1 + z * sqrt((1 + CV^2)/(2*n)));
coverage(i,8) = (true_CV >= L && true_CV <= U);
width(i,8) = U - L;

%% ========================
%% MEDIAN METHODS

%% 9. Classical (Median)
SCV_me = CV_me * sqrt((1 + 2*CV_me^2)/(2*n));
L = CV_me - tcri * SCV_me;
U = CV_me + tcri * SCV_me;
coverage(i,9) = (true_CV >= L && true_CV <= U);
width(i,9) = U - L;

%% 10. McKay (Median)
L = CV_me * sqrt(((chi2_lower/n - 1)*CV_me^2) + (chi2_lower/n));
U = CV_me * sqrt(((chi2_upper/n - 1)*CV_me^2) + (chi2_upper/n));
coverage(i,10) = (true_CV >= L && true_CV <= U);
width(i,10) = U - L;
```

```
%% 11. Miller (Median)
L = CV_me - z * sqrt((CV_me^4 + 0.5*CV_me^2)/n);
U = CV_me + z * sqrt((CV_me^4 + 0.5*CV_me^2)/n);
coverage(i,11) = (true_CV >= L && true_CV <= U);
width(i,11) = U - L;

%% 12. Curto & Pinto (Median)
L = CV_me * (A - z * B);
U = CV_me * (A + z * B);
coverage(i,12) = (true_CV >= L && true_CV <= U);
width(i,12) = U - L;

%% 13. Sharma & Krishna (Median)
L = CV_me - z * (CV_me / sqrt(n));
U = CV_me + z * (CV_me / sqrt(n));
coverage(i,13) = (true_CV >= L && true_CV <= U);
width(i,13) = U - L;

%% ==========================
%% 14. Bootstrap-t

bootCV = zeros(numBootstraps,1);
for b_iter = 1:numBootstraps
   bs = datasample(sample, n);
   bootCV(b_iter) = std(bs)/mean(bs);
end

tboot = (bootCV - CV) ./ (bootCV .* sqrt((1 + 2*bootCV.^2)/(2*n)));
t_sorted = sort(tboot);

lower_idx = max(1, floor((alpha/2)*numBootstraps));
upper_idx = min(numBootstraps, ceil((1 - alpha/2)*numBootstraps));

SCV = CV * sqrt((1 + 2*CV^2)/(2*n));
L = CV + t_sorted(lower_idx) * SCV;
U = CV + t_sorted(upper_idx) * SCV;

coverage(i,14) = (true_CV >= L && true_CV <= U);
width(i,14) = U - L;

%% ==========================
%% 15. Bootstrap-t (Median)

bootCV = zeros(numBootstraps,1);
for b_iter = 1:numBootstraps
   bs = datasample(sample, n);
   med_b = median(bs);
```

```
        s_b = sqrt(sum((bs - med_b).^2)/(n-1));
        bootCV(b_iter) = s_b / med_b;
    end

    tboot = (bootCV - CV_me) ./ (bootCV .* sqrt((1 + 2*bootCV.^2)/(2*n)));
    t_sorted = sort(tboot);

    SCV_me = CV_me * sqrt((1 + 2*CV_me^2)/(2*n));
    L = CV_me + t_sorted(lower_idx) * SCV_me;
    U = CV_me + t_sorted(upper_idx) * SCV_me;

    coverage(i,15) = (true_CV >= L && true_CV <= U);
    width(i,15) = U - L;

  end

  %% RESULTS
  CP = mean(coverage);
  AW = mean(width);

  fprintf('\n===== n = %d =====\n', n);
  for k = 1:15
    fprintf('Method %d -> CP: %.4f  AW: %.4f\n', k, CP(k), AW(k));
  end

end
```

Appendix A.2

Real-Life Example Codes

Case 1: Health Science Data (positively skewed)

```
clc; clear; close all;
```

```
%% DATA
x = [10, 15, 25, 60, 150, 320, 450, 610, 405, 280, 190, 135, 100, 75, 50];
x = x(:)'; % ensure row vector
n = length(x);
alpha = 0.05;

%% BASIC STATISTICS
xbar = mean(x);
s = std(x);
cv = s / xbar;

med = median(x);
mad_val = mad(x,1); % FIXED (renamed variable)
cv_med = mad_val / med;

z = norminv(1 - alpha/2);

%% CLASSICAL
CL_L = cv * (1 - z/sqrt(2*n));
CL_U = cv * (1 + z/sqrt(2*n));

%% MCKAY
chiL = chi2inv(alpha/2, n-1);
chiU = chi2inv(1-alpha/2, n-1);

MK_L = cv * sqrt((n-1)/chiU);
MK_U = cv * sqrt((n-1)/chiL);

%% MILLER
ML_L = cv * (1 - z/sqrt(n));
ML_U = cv * (1 + z/sqrt(n));

%% SHARMA & KRISHNA
SK_L = cv * exp(-z/sqrt(n));
SK_U = cv * exp(z/sqrt(n));

%% CURTO & PINTO
CP_L = cv / (1 + z/sqrt(n));
CP_U = cv / (1 - z/sqrt(n));

%% AA&K-LS
AAK_LS_L = cv * (1 - z*sqrt((1+2*cv^2)/(2*n)));
AAK_LS_U = cv * (1 + z*sqrt((1+2*cv^2)/(2*n)));

%% AA&K-ALS
AAK_ALS_L = cv * (1 - z*sqrt((1+cv^2)/(n)));
AAK_ALS_U = cv * (1 + z*sqrt((1+cv^2)/(n)));

%% PANICHKITKOSOLKUL
PK_L = cv * exp(-z*sqrt((1+cv^2)/(n)));
PK_U = cv * exp(z*sqrt((1+cv^2)/(n)));

%% MEDIAN-BASED METHODS

CL_Me_L = cv_med * (1 - z/sqrt(2*n));
```

```
CL_Me_U = cv_med * (1 + z/sqrt(2*n));

MK_Me_L = cv_med * sqrt((n-1)/chiU);
MK_Me_U = cv_med * sqrt((n-1)/chiL);

ML_Me_L = cv_med * (1 - z/sqrt(n));
ML_Me_U = cv_med * (1 + z/sqrt(n));

CP_Me_L = cv_med / (1 + z/sqrt(n));
CP_Me_U = cv_med / (1 - z/sqrt(n));

SK_Me_L = cv_med * exp(-z/sqrt(n));
SK_Me_U = cv_med * exp(z/sqrt(n));

%% BOOTSTRAP
B = 1000;
boot_cv = zeros(B,1);

for i = 1:B
   xb = x(randi(n,n,1)); % safer than datasample
   boot_cv(i) = std(xb)/mean(xb);
end

se_boot = std(boot_cv);
BT_L = cv - z*se_boot;
BT_U = cv + z*se_boot;

%% BOOTSTRAP MEDIAN
boot_cv_me = zeros(B,1);

for i = 1:B
   xb = x(randi(n,n,1));
   boot_cv_me(i) = mad(xb,1)/median(xb); % works now
end

se_boot_me = std(boot_cv_me);
BT_Me_L = cv_med - z*se_boot_me;
BT_Me_U = cv_med + z*se_boot_me;

%% WIDTH FUNCTION
width = @(L,U) U - L;

%% DISPLAY
fprintf('\nMethod\t\tLower\t\tUpper\t\tWidth\n');

fprintf('CCL\t\t%.4f\t%.4f\t%.4f\n',CL_L,CL_U,width(CL_L,CL_U));
fprintf('MK\t\t%.4f\t%.4f\t%.4f\n',MK_L,MK_U,width(MK_L,MK_U));
fprintf('ML\t\t%.4f\t%.4f\t%.4f\n',ML_L,ML_U,width(ML_L,ML_U));
fprintf('S&K\t\t%.4f\t%.4f\t%.4f\n',SK_L,SK_U,width(SK_L,SK_U));
fprintf('C&P\t\t%.4f\t%.4f\t%.4f\n',CP_L,CP_U,width(CP_L,CP_U));
fprintf('AAK-LS\t\t%.4f\t%.4f\t%.4f\n',AAK_LS_L,AAK_LS_U,width(AAK_LS_L,AAK_LS_U));
fprintf('AAK-ALS\t\t%.4f\t%.4f\t%.4f\n',AAK_ALS_L,AAK_ALS_U,width(AAK_ALS_L,AAK_ALS_U));
fprintf('PK\t\t%.4f\t%.4f\t%.4f\n',PK_L,PK_U,width(PK_L,PK_U));

fprintf('CCL_Me\t\t%.4f\t%.4f\t%.4f\n',CL_Me_L,CL_Me_U,width(CL_Me_L,CL_Me_U));
```

```
fprintf('MK_Me\t\t%.4f\t%.4f\t%.4f\n',MK_Me_L,MK_Me_U,width(MK_Me_L,MK_Me_U));
fprintf('ML_Me\t\t%.4f\t%.4f\t%.4f\n',ML_Me_L,ML_Me_U,width(ML_Me_L,ML_Me_U));
fprintf('C&P_Me\t\t%.4f\t%.4f\t%.4f\n',CP_Me_L,CP_Me_U,width(CP_Me_L,CP_Me_U));
fprintf('S&K_Me\t\t%.4f\t%.4f\t%.4f\n',SK_Me_L,SK_Me_U,width(SK_Me_L,SK_Me_U));

fprintf('BT\t\t%.4f\t%.4f\t%.4f\n',BT_L,BT_U,width(BT_L,BT_U));
fprintf('BT_Me\t\t%.4f\t%.4f\t%.4f\n',BT_Me_L,BT_Me_U,width(BT_Me_L,BT_Me_U));
```

Case 2: Business Data (negatively skewed)

```
clc; clear; close all;

%% DATA
x = [98, 100, 95, 99, 100, 97, 100, 42, 96, 100];
x = x(:)';
n = length(x);
alpha = 0.05;

%% BASIC STATISTICS
xbar = mean(x);
s = std(x);

if abs(xbar) < 1e-8
   error('Mean too close to zero. CV undefined.');
end

cv = s / xbar;

med = median(x);

if abs(med) < 1e-8
   warning('Median near zero → median CV unstable.');
end

mad_val = mad(x,1);   % you can switch to mad(x,0) if needed
cv_med = mad_val / med;

z = norminv(1 - alpha/2);

%% CLASSICAL
CL_L = max(0, cv * (1 - z/sqrt(2*n)));
CL_U = cv * (1 + z/sqrt(2*n));

%% MCKAY
chiL = chi2inv(alpha/2, n-1);
chiU = chi2inv(1-alpha/2, n-1);

MK_L = cv * sqrt((n-1)/chiU);
MK_U = cv * sqrt((n-1)/chiL);
```

```
%% MILLER
ML_L = max(0, cv * (1 - z/sqrt(n)));
ML_U = cv * (1 + z/sqrt(n));

%% SHARMA & KRISHNA
SK_L = cv * exp(-z/sqrt(n));
SK_U = cv * exp(z/sqrt(n));

%% CURTO & PINTO
if (1 - z/sqrt(n)) <= 0
   CP_L = NaN;
   CP_U = NaN;
else
   CP_L = cv / (1 + z/sqrt(n));
   CP_U = cv / (1 - z/sqrt(n));
end

%% AAK-LS
AAK_LS_L = max(0, cv * (1 - z*sqrt((1+2*cv^2)/(2*n))));
AAK_LS_U = cv * (1 + z*sqrt((1+2*cv^2)/(2*n)));

%% AAK-ALS
AAK_ALS_L = max(0, cv * (1 - z*sqrt((1+cv^2)/(n))));
AAK_ALS_U = cv * (1 + z*sqrt((1+cv^2)/(n)));

%% PANICHKITKOSOLKUL
PK_L = cv * exp(-z*sqrt((1+cv^2)/(n)));
PK_U = cv * exp(z*sqrt((1+cv^2)/(n)));

%% MEDIAN-BASED METHODS

CL_Me_L = max(0, cv_med * (1 - z/sqrt(2*n)));
CL_Me_U = cv_med * (1 + z/sqrt(2*n));

MK_Me_L = cv_med * sqrt((n-1)/chiU);
MK_Me_U = cv_med * sqrt((n-1)/chiL);

ML_Me_L = max(0, cv_med * (1 - z/sqrt(n)));
ML_Me_U = cv_med * (1 + z/sqrt(n));

if (1 - z/sqrt(n)) <= 0
   CP_Me_L = NaN;
   CP_Me_U = NaN;
else
   CP_Me_L = cv_med / (1 + z/sqrt(n));
   CP_Me_U = cv_med / (1 - z/sqrt(n));
end

SK_Me_L = cv_med * exp(-z/sqrt(n));
SK_Me_U = cv_med * exp(z/sqrt(n));

%% BOOTSTRAP (ROBUST)
B = 2000;
boot_cv = zeros(B,1);
```

```
for i = 1:B
   xb = x(randi(n,n,1));
   m = mean(xb);

   if abs(m) > 1e-8
      boot_cv(i) = std(xb)/m;
   else
      boot_cv(i) = NaN;
   end
end

boot_cv = boot_cv(~isnan(boot_cv));
se_boot = std(boot_cv);

BT_L = max(0, cv - z*se_boot);
BT_U = cv + z*se_boot;

%% BOOTSTRAP MEDIAN
boot_cv_me = zeros(B,1);

for i = 1:B
   xb = x(randi(n,n,1));
   m = median(xb);

   if abs(m) > 1e-8
      boot_cv_me(i) = mad(xb,1)/m;
   else
      boot_cv_me(i) = NaN;
   end
end

boot_cv_me = boot_cv_me(~isnan(boot_cv_me));
se_boot_me = std(boot_cv_me);

BT_Me_L = max(0, cv_med - z*se_boot_me);
BT_Me_U = cv_med + z*se_boot_me;

%% WIDTH FUNCTION
width = @(L,U) U - L;

%% DISPLAY
fprintf('\nMethod\t\tLower\t\tUpper\t\tWidth\n');

fprintf('CCL\t\t%.4f\t%.4f\t%.4f\n',CL_L,CL_U,width(CL_L,CL_U));
fprintf('MK\t\t%.4f\t%.4f\t%.4f\n',MK_L,MK_U,width(MK_L,MK_U));
fprintf('ML\t\t%.4f\t%.4f\t%.4f\n',ML_L,ML_U,width(ML_L,ML_U));
fprintf('S&K\t\t%.4f\t%.4f\t%.4f\n',SK_L,SK_U,width(SK_L,SK_U));
fprintf('C&P\t\t%.4f\t%.4f\t%.4f\n',CP_L,CP_U,width(CP_L,CP_U));
fprintf('AAK-LS\t\t%.4f\t%.4f\t%.4f\n',AAK_LS_L,AAK_LS_U,width(AAK_LS_L,AAK_LS_U));
fprintf('AAK-ALS\t\t%.4f\t%.4f\t%.4f\n',AAK_ALS_L,AAK_ALS_U,width(AAK_ALS_L,AAK_ALS_U));
fprintf('PK\t\t%.4f\t%.4f\t%.4f\n',PK_L,PK_U,width(PK_L,PK_U));

fprintf('CCL_Me\t\t%.4f\t%.4f\t%.4f\n',CL_Me_L,CL_Me_U,width(CL_Me_L,CL_Me_U));
fprintf('MK_Me\t\t%.4f\t%.4f\t%.4f\n',MK_Me_L,MK_Me_U,width(MK_Me_L,MK_Me_U));
```

```
fprintf('ML_Me\t\t%.4f\t%.4f\t%.4f\n',ML_Me_L,ML_Me_U,width(ML_Me_L,ML_Me_U));
fprintf('C&P_Me\t\t%.4f\t%.4f\t%.4f\n',CP_Me_L,CP_Me_U,width(CP_Me_L,CP_Me_U));
fprintf('S&K_Me\t\t%.4f\t%.4f\t%.4f\n',SK_Me_L,SK_Me_U,width(SK_Me_L,SK_Me_U));

fprintf('BT\t\t%.4f\t%.4f\t%.4f\n',BT_L,BT_U,width(BT_L,BT_U));
fprintf('BT_Me\t\t%.4f\t%.4f\t%.4f\n',BT_Me_L,BT_Me_U,width(BT_Me_L,BT_Me_U));
```

Case 3: Business Data (normally distributed)

```
clc; clear; close all;
%% DATA
x = [99.8, 100.1, 100.2, 99.9, 100.0, 100.3, 99.7, 100.0, 99.8, 100.1];
x = x(:)'; % ensure row vector
n = length(x);
alpha = 0.05;
%% BASIC STATISTICS
xbar = mean(x);
s = std(x);
cv = s / xbar;
med = median(x);
mad_val = mad(x,1); % FIXED (renamed variable)
cv_med = mad_val / med;
z = norminv(1 - alpha/2);
%% CLASSICAL
CL_L = cv * (1 - z/sqrt(2*n));
CL_U = cv * (1 + z/sqrt(2*n));
%% MCKAY
chiL = chi2inv(alpha/2, n-1);
chiU = chi2inv(1-alpha/2, n-1);
MK_L = cv * sqrt((n-1)/chiU);
MK_U = cv * sqrt((n-1)/chiL);
%% MILLER
ML_L = cv * (1 - z/sqrt(n));
ML_U = cv * (1 + z/sqrt(n));
%% SHARMA & KRISHNA
SK_L = cv * exp(-z/sqrt(n));
SK_U = cv * exp(z/sqrt(n));
%% CURTO & PINTO
CP_L = cv / (1 + z/sqrt(n));
CP_U = cv / (1 - z/sqrt(n));
%% AA&K-LS
AAK_LS_L = cv * (1 - z*sqrt((1+2*cv^2)/(2*n)));
AAK_LS_U = cv * (1 + z*sqrt((1+2*cv^2)/(2*n)));
%% AA&K-ALS
```

```
AAK_ALS_L = cv * (1 - z*sqrt((1+cv^2)/(n)));
AAK_ALS_U = cv * (1 + z*sqrt((1+cv^2)/(n)));
%% PANICHKITKOSOLKUL
PK_L = cv * exp(-z*sqrt((1+cv^2)/(n)));
PK_U = cv * exp(z*sqrt((1+cv^2)/(n)));
%% MEDIAN-BASED METHODS
CL_Me_L = cv_med * (1 - z/sqrt(2*n));
CL_Me_U = cv_med * (1 + z/sqrt(2*n));
MK_Me_L = cv_med * sqrt((n-1)/chiU);
MK_Me_U = cv_med * sqrt((n-1)/chiL);
ML_Me_L = cv_med * (1 - z/sqrt(n));
ML_Me_U = cv_med * (1 + z/sqrt(n));
CP_Me_L = cv_med / (1 + z/sqrt(n));
CP_Me_U = cv_med / (1 - z/sqrt(n));
SK_Me_L = cv_med * exp(-z/sqrt(n));
SK_Me_U = cv_med * exp(z/sqrt(n));
%% BOOTSTRAP
B = 1000;
boot_cv = zeros(B,1);
for i = 1:B
   xb = x(randi(n,n,1)); % safer than datasample
   boot_cv(i) = std(xb)/mean(xb);
end
se_boot = std(boot_cv);
BT_L = cv - z*se_boot;
BT_U = cv + z*se_boot;
%% BOOTSTRAP MEDIAN
boot_cv_me = zeros(B,1);
for i = 1:B
   xb = x(randi(n,n,1));
   boot_cv_me(i) = mad(xb,1)/median(xb); % works now
end
se_boot_me = std(boot_cv_me);
BT_Me_L = cv_med - z*se_boot_me;
BT_Me_U = cv_med + z*se_boot_me;
%% WIDTH FUNCTION
width = @(L,U) U - L;
%% DISPLAY
fprintf('\nMethod\t\tLower\t\tUpper\t\tWidth\n');
fprintf('CCL\t\t%.4f\t%.4f\t%.4f\n',CL_L,CL_U,width(CL_L,CL_U));
fprintf('MK\t\t%.4f\t%.4f\t%.4f\n',MK_L,MK_U,width(MK_L,MK_U));
fprintf('ML\t\t%.4f\t%.4f\t%.4f\n',ML_L,ML_U,width(ML_L,ML_U));
fprintf('S&K\t\t%.4f\t%.4f\t%.4f\n',SK_L,SK_U,width(SK_L,SK_U));
fprintf('C&P\t\t%.4f\t%.4f\t%.4f\n',CP_L,CP_U,width(CP_L,CP_U));
fprintf('AAK-LS\t\t%.4f\t%.4f\t%.4f\n',AAK_LS_L,AAK_LS_U,width(AAK_LS_L,AAK_LS_U));
fprintf('AAK-ALS\t\t%.4f\t%.4f\t%.4f\n',AAK_ALS_L,AAK_ALS_U,width(AAK_ALS_L,AAK_ALS_U));
fprintf('PK\t\t%.4f\t%.4f\t%.4f\n',PK_L,PK_U,width(PK_L,PK_U));
fprintf('CCL_Me\t\t%.4f\t%.4f\t%.4f\n',CL_Me_L,CL_Me_U,width(CL_Me_L,CL_Me_U));
```

```
fprintf('MK_Me\t\t%.4f\t%.4f\t%.4f\n',MK_Me_L,MK_Me_U,width(MK_Me_L,MK_Me_U));
fprintf('ML_Me\t\t%.4f\t%.4f\t%.4f\n',ML_Me_L,ML_Me_U,width(ML_Me_L,ML_Me_U));
fprintf('C&P_Me\t\t%.4f\t%.4f\t%.4f\n',CP_Me_L,CP_Me_U,width(CP_Me_L,CP_Me_U));
fprintf('S&K_Me\t\t%.4f\t%.4f\t%.4f\n',SK_Me_L,SK_Me_U,width(SK_Me_L,SK_Me_U));
fprintf('BT\t\t%.4f\t%.4f\t%.4f\n',BT_L,BT_U,width(BT_L,BT_U));
fprintf('BT_Me\t\t%.4f\t%.4f\t%.4f\n',BT_Me_L,BT_Me_U,width(BT_Me_L,BT_Me_U));
```