Bayesian inference on shifted Lindley distribution based on different loss functions

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF SCIENCE
IN
STATISTICS

Submitted by

Tanmay Gayen

ROLL NO: M.Sc(Sem-IV)Stat-16

REGISTRATION NO:VB-0311 of 2022-2023

DEPARTMENT OF STATISTICS
SIKSHA BHAVANA
VISVA-BHARATI
(A Central University and an Institution of National Importance)
Santiniketan, 731235
JUNE, 2024

DECLARATION

I, **Tanmay Gayen (Roll No: M.Sc(Sem-IV)Stat-16)**, hereby declare that this report entitled **"Bayesian inference on shifted Lindley distribution based on different loss functions"** submitted to Visva Bharati, Bolpur, Santiniketan towards the partial requirement of **M.Sc** in **Statistics**, is an original work carried out by me under the supervision of **Dr. Arindom Chakraborty** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold academic ethics and honesty. Whenever a piece of external information or statement or result is used then, that has been duly acknowledged and cited.

Bolpur, Santiniketan - 731235
June,2024

Tanmay Gayen

ACKNOWLEDGEMENT

I thank everyone who helped me see this project through to completion. I would like to first express my profound gratitude and deep regard to Dr. Arindom Chakraborty, Visva-Bharati, and sincerely wish to acknowledge his vision, guidance, valuable feedback, and constant support throughout the duration of this project. I am also thankful to all of my teachers and my friends for their steadfast encouragement and time.

I am lastly grateful to the Visva-Bharati, Santiniketan for providing the necessary resources and facilities to complete this project to the best of my ability.

June,2024
Bolpur, Santiniketan - 731235

Tanmay Gayen
Department of Statistics
Visva-Bharati, Bolpur, Santiniketan
731235, India

ABSTRACT

This work focuses on Bayesian estimation of the parameters and reliability function of the shifted Lindley distribution by using symmetric and asymmetric loss functions. Squared error, linear exponential (linex) and general entropy loss functions are used to find the Bayes estimates of the parameters and the Reliability function. Instead of generating samples using MCMC algorithm, an approximate method based on Lindley's work has been used where the posterior distributions were approximated. A simulation study was performed to observe the performance of the proposed method under two set up with different sample sizes. Finally the proposed method has been used for a data set on strength of the sintered silicon nitride.

Contents

1 Introduction

Bayesian inference has emerged as a powerful tool in statistical modeling, offering a flexible framework for incorporating prior knowledge into the analysis of data. One area where Bayesian methods have found widespread application is in the modeling of probability distributions, enabling researchers to make robust statistical inferences about underlying processes. In recent years, there has been growing interest in extending Bayesian inference techniques to non-standard distributions, which better capture the complexities of real-world data.

The shifted Lindley distribution, a member of the Lindley family of distributions, has gained prominence in various fields due to its versatility in modeling count and survival data. Unlike traditional distributions, the shifted Lindley distribution incorporates an additional parameter to account for a shift in location, making it particularly suitable for capturing skewed and positively skewed data. However, despite its advantages, the statistical properties and inference methods for the shifted Lindley distribution remain relatively under explored.

This project focuses on Bayesian inference techniques applied to the shifted Lindley distribution, exploring the impact of different loss functions on parameter estimation. The shifted Lindley distribution, known for its versatility in modeling skewed data, requires robust statistical methods for accurate parameter estimation. By employing various loss functions, we aim to assess the effectiveness and robustness of Bayesian inference in this context. Through this study, we seek to enhance our understanding of Bayesian techniques for non-standard distributions and provide valuable insights for practical applications in diverse fields.

The probability density function (pdf) of the Shifted Lindley distribution from Maiti et al. (2021) with parameters (θ, μ) is given by:

$$f(x;\theta,\mu) = \frac{\theta^2}{1+\theta^(1+\mu)}(1+x)e^{-\theta(x-\mu)}, x > \mu > 0 \tag{1.1}$$

The CDF of a Shifted Lindley distribution with parameters (θ, μ) is given by:

$$F(x;\theta,\mu) = 1 - \frac{1+\theta(1+x)}{1+\theta(1+\mu)}e^{-\theta(x-\mu)}, x > \mu > 0 \tag{1.2}$$

Note that if we put μ=0 in equations(1.1)and in (1.2), these equations become the PDF and CDF, respectively, of a Lindley distribution with a single parameter θ.

The shape of the shifted Lindley distribution depends on its parameters μ and θ. Figure 1 illustrates the pdf and cdf for various values of these parameters.

- Smaller θ: The pdf is right-skewed, indicating a longer tail on the right side.
- Larger θ: The pdf resembles an inverted J shape, tapering off to look like a standard exponential curve, indicating a rapid decline in probability for larger values.

This demonstrates the distribution's flexibility and the impact of θ on its shape.

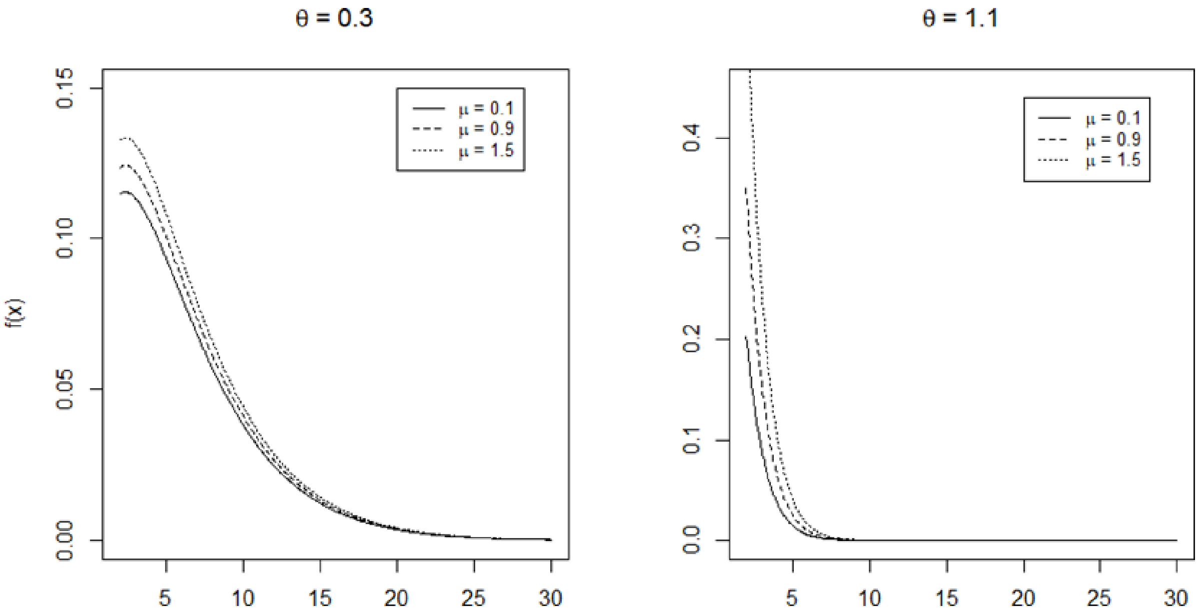

Figure 1: The PDF's of various Shifted Lindley distribution for different values of parameters. Reproduced from original version as presented by Maiti et al. (2021).

The survivor function $R(t)$ is defined as:

$$R(t) = 1 - F(t) = P(T > t) \quad \text{for } t > 0.$$

The survivor function simply indicates the probability that the event of interest has not yet occurred by time t; thus, if T denotes time until death, $R(t)$ denotes the probability of surviving beyond time t.

$$R(t, \theta, \mu) = p(T > t) = 1 - F(t) = 1 - (1 - \frac{1 + \theta(1 + t)}{1 + \theta(1 + \mu)} e^{-\theta(t-\mu)}) = \frac{1 + \theta(1 + t)}{1 + \theta(1 + \mu)} e^{-\theta(t-\mu)}. \tag{1.3}$$

This particular distribution with the specified survivor function as above was introduced by Maiti et al. (2021) as a new distribution useful to analyze lifetime data. They propose a shifted version of widely-used lindley distribution. Some statistical properties such as stochastic ordering, moment generating function, reliability characteristic etc. are studied for this new distribution. For estimating unknown parameters, two types of estimation method viz. method of moments and maximum likelihood method are explored. A simulation study for several choices of parameters is executed. Finally, a real data application illustrates the performance of our proposed distribution. From now on, the shifted lindley distribution with parameters θ and μ will be denoted as SL(θ, μ).

2 Moments of the Shifted Lindley Distribution

The mean of a distribution can be obtained by calculating the first moment about the origin, which represents the center or average value of the distribution. This involves summing or integrating the product of the variable's values and their corresponding probabilities (for discrete distributions) or probability density function (for continuous distributions). The mean provides a measure of the central tendency of the distribution.

2.1 Mean

$$\mu_1^{'} = E(X) = \frac{\theta^2}{1+\theta(1+\mu)} \int_{\mu}^{\infty} x(1+x)e^{-\theta(x-\mu)}\, dx = \mu + \frac{2}{\theta} - \frac{1+\mu}{1+\theta(1+\mu)}$$

2.2 Theorem

For $k \geq 0$, the recurrence relation for the higher order moments is:

$$\mu_{k+1}^{'} = \mu_1^{'}\mu_k^{'} - \frac{d}{d\theta}\mu_1^{'}.$$

Using this recurrence relation we will find all higher order moment.

2.3 Variance, Skewness, Kurtosis

The second moment (μ_2) can be calculated as:

$$\mu_2 = \mu_2^{'} - \mu_1^{'\,2} = \frac{2}{\theta^2} - \frac{(1+\mu)^2}{(1+\theta(1+\mu))^2}.$$

Putting $\mu = 0$ implies:

$$\mu_2 = \frac{2}{\theta^2} - \frac{1}{(1+\theta)^2}.$$

which is the variance of a Lindley distribution with parameter θ.
Similarly, for a SL(μ, θ) distribution:

$$\mu_3 = \frac{4}{\theta^3} - \frac{2(1+\mu)^3}{(1+\theta(1+\mu))^3}.$$

and

$$\mu_4 = \frac{24}{\theta^4} - \frac{3(1+\mu)^4}{(1+\theta(1+\mu))^4} - \frac{12(1+\mu)^2}{\theta^2(1+\theta(1+\mu))^2}.$$

Then the formula of skewness is -

$$\beta_1 = \frac{\mu_3^2}{\mu_2^3},$$

and the formula of Kurtosis is -

$$\beta_2 = \frac{\mu_4}{\mu_2^2}.$$

3 Maximum Likelihood (ML) Estimation of Parameters:

The likelihood function for a random sample $X_1, X_2, \ldots, X_n$ which is taken from SL(μ, θ) distribution is given by:

$$L(X, \mu, \theta) = \frac{\theta^{2n}}{(1+\theta(1+\mu))^n} \prod_{i=1}^{n} (1+x_i) e^{-\theta \sum_{i=1}^{n}(x_i-\mu)} \tag{3.1}$$

Then the log-likelihood function is given by

$$log(L(X, \mu, \theta)) = (2n \log\theta) - n\log(1+\theta(1+\mu)) + \sum_{i=1}^{n} log(1+X_i) - \theta\sum_{i=1}^{n}(X_i - \mu). \tag{3.2}$$

It is to be noted that the maximum likelihood estimator (MLE) of μ is:

$$\hat{\mu}_{\text{MLE}} = \min_i X_i = X_{(1)}. \tag{3.3}$$

Differentiating the log-likelihood with respect to θ, we get the following equation:

$$\frac{2n}{\theta} - \frac{n(1+\hat{\mu}_{\text{MLE}})}{1+\theta(1+\hat{\mu}_{\text{MLE}})} - \sum_{i=1}^{n}(X_i - \hat{\mu}_{\text{MLE}}) = 0. \tag{3.4}$$

To find the maximum likelihood (ML) estimate of the parameter θ, we typically maximize the log-likelihood function which needs to be solved using some iterative procedure. We will use R programming for iterative procedure.

4 Loss Functions

In Bayesian estimation, a loss function measures the penalty associated with inaccuracies in estimating parameters. It quantifies the "cost" or undesirability of estimation errors. By specifying a loss function, Bayesian estimators aim to minimize the expected loss, considering the uncertainty in parameter estimation. The choice of loss function depends on the specific problem domain and the consequences of estimation errors. Different loss functions can lead to different Bayesian estimators.

4.1 Squared Error (SE) Loss Function

The squared error (SE) loss function is a way to quantify the discrepancy between the true value of a parameter, denoted as $g(\theta)$, and its estimate, denoted as $\hat{g}(\theta)$. It is often used to evaluate the performance of different estimation methods and to derive Bayesian estimators.

The squared error loss function is defined as:

$$\Delta_{SE}(g(\theta), \hat{g}(\theta) = (g(\theta) - \hat{g}(\theta))^2. \tag{4.1}$$

Here, $g(\theta)$ represents the true parameter value, and $\hat{g}(\theta)$ represents its estimate. The squared difference $(g(\theta) - \hat{g}(\theta))^2$ is used to measure the discrepancy between the true value and its estimate. This discrepancy is squared to ensure that both positive and negative errors contribute to the overall loss, and larger discrepancies are penalized more severely.

The Bayes estimate of $g(\theta)$ against SE loss function is the posterior mean is given by

$$\hat{g}_{\mathrm{SE}}(\theta) = E_\theta[g(\theta) \mid \mathrm{data}]. \tag{4.2}$$

The Bayesian method,Pak et al. (2019), apply Squared error loss function in the data analysis problem.

4.2 General Entropy (GE) Loss Function

The General Entropy (GE) loss function is defined as:

$$\Delta_{\mathrm{GE}}(g(\theta), \hat{g}(\theta)) \propto \left(\frac{\hat{g}(\theta)}{g(\theta)}\right)^{\omega} - \omega \log\left(\frac{\hat{g}(\theta)}{g(\theta)}\right) - 1, \quad \omega \neq 0, \tag{4.3}$$

where $g(\theta)$ is the true parameter, $\hat{g}(\theta)$ is the estimated parameter, and ω is a parameter controlling the asymmetry of the loss function. Alduais (2021) provide a comprehensive review on applications of this particular loss function in classical and Bayesian statistical computations.

Bayesian Estimate Using GE Loss Function: The Bayesian estimate of $g(\theta)$ based on the GE loss function is given by:

$$\hat{g}_{\text{GE}}(\theta) = \left(E_\theta \left[(g(\theta))^{-\omega} \mid \text{data}\right]\right)^{-1/\omega}. \tag{4.4}$$

4.3 Linex Loss Function

The Linex loss function is defined as:

$$\Delta_{\text{LE}}(g(\theta), \hat{g}(\theta)) = \exp(\nu(g(\theta) - \hat{g}(\theta))) - \nu(g(\theta) - \hat{g}(\theta)) - 1, \quad \nu \neq 0, \tag{4.5}$$

where $g(\theta)$ is the true parameter, $\hat{g}(\theta)$ is the estimated parameter, and ν is a parameter controlling the asymmetry of the loss function.

Bayesian Estimate Using Linex Loss Function: The Bayesian estimate of $g(\theta)$ based on the Linex loss function is given by:

$$\hat{g}_{\text{LE}}(\theta) = -\frac{1}{\nu} \log E_\theta \left[\exp(-\nu g(\theta)) \mid \text{data}\right]. \tag{4.6}$$

The Bayesian method,Pak et al. (2019), apply Squared error loss function in the data analysis problem

5 Bayesian Analysis

Let $X_1, \ldots, X_n$ be a random sample of size n from the SL(θ, μ) distribution. For given data from this sample, $x = (x_1, \ldots, x_n)$, the likelihood function is given by:

$$l(X, \mu, \theta) = \frac{\theta^{2n}}{(1 + \theta(1 + \mu))^n} \prod_{i=1}^{n} (1 + x_i) e^{-\theta \sum_{i=1}^{n} (x_i - \mu)}. \tag{5.1}$$

Some prior distributions on the parameters are required to implement a Bayesian analysis.

$$\pi_1(\theta; a_1, b_1) \propto \theta^{a_1 - 1} e^{-b_1 \theta}, \quad \theta > 0; \tag{5.2}$$
$$\pi_2(\mu; a_2, b_2) \propto \mu^{a_2 - 1} e^{-b_2 \mu}, \quad \mu > 0, \tag{5.3}$$

where $\{a_1, a_2, b_1, b_2\}$ are positive hyperparameters.

The posterior joint probability density function of θ and μ given the data can be written as:

$$\text{Posterior}(\theta, \mu | \text{data}) \propto \text{likelihood}(\text{data} | \theta, \mu) \times \text{prior}(\theta) \times \text{prior}(\mu). \tag{5.4}$$

Here, one would substitute the specific forms of the likelihood function and the prior densities into the equation above.The posterior joint probability density of θ and μ is

$$\pi^*(\theta,\mu|x) = \frac{\theta^{2n+a_1-1}}{C(1+\theta(1+\mu))^n}e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))}e^{-b_2\mu}\mu^{a_2-1}[\prod_{i=1}^n(1+x_i)]. \tag{5.5}$$

In the above equation, the constant is determined as:

$$C = \int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1-1}}{(1+\theta(1+\mu))^n}e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))}e^{-b_2\mu}\mu^{a_2-1}[\prod_{i=1}^n(1+x_i)]d\theta d\mu. \tag{5.6}$$

We derive the Bayes estimates of θ, μ, and the reliability parameter $R(t)$. First, assuming squared error (SE) loss function, the estimates of θ and μ become:

$$\hat{\theta}_{SE} = E[\theta|X] = \frac{1}{c}\int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1}}{(1+\theta(1+\mu))^n}e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))}e^{-b_2\mu}\mu^{a_2-1}[\prod_{i=1}^n(1+x_i)]d\theta d\mu, \tag{5.7}$$

$$\hat{\mu}_{SE} = E[\mu|X] = \frac{1}{c}\int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1-1}}{(1+\theta(1+\mu))^n}e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))}e^{-b_2\mu}\mu^{a_2}[\prod_{i=1}^n(1+x_i)]d\theta d\mu. \tag{5.8}$$

The Bayes estimate of R(t) against the squared error loss function can be derived

$$\hat{R}_{SE}(t) = E[\frac{1+\theta(1+t)}{1+\theta(1+\mu)}e^{-\theta(t-\mu)}|\,X] \tag{5.9}$$

$$= \frac{1}{c}\int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1-1}}{(1+\theta(1+\mu))^n}e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))}e^{-b_2\mu}\mu^{a_2-1}[\prod_{i=1}^n(1+x_i)] \frac{1+\theta(1+t)}{1+\theta(1+\mu)}e^{-\theta(t-\mu)}d\theta d\mu \tag{5.10}$$

By using the loss function Δ_{LE}, the Bayes estimates of θ, μ, and the reliability parameter become, respectively:

$$\hat{\theta}_{LE} = -\frac{1}{\nu}\log\left(E\left[\exp(-\nu\theta)\,|\,X\right]\right) \tag{5.11}$$

$$= -\frac{1}{\nu} log[\frac{1}{c}\int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1-1}}{(1+\theta(1+\mu))^n} e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu)+\nu)} e^{-b_2\mu}\mu^{a_2-1}[\prod_{i=1}^n(1+x_i)]d\theta d\mu] \tag{5.12}$$

$$\hat{\mu}_{LE} = -\frac{1}{\nu}\log\left(E\left[\exp(-\nu\mu)\mid X\right]\right) \tag{5.13}$$

$$= -\frac{1}{\nu} log[\frac{1}{c}\int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1-1}}{(1+\theta(1+\mu))^n} e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))} e^{-\mu(b_2+\nu)}\mu^{a_2-1}[\prod_{i=1}^n(1+x_i)]d\theta d\mu] \tag{5.14}$$

Also,the Bayes estimate of R(t) against the loss function Δ_{LE} can be derived as

$$\hat{R}(t)_{LE} = -\frac{1}{\nu}\log\mathbb{E}\left[\exp\left(-\nu R(t)\right)\mid x\right] \tag{5.15}$$

$$= -\frac{1}{\nu} log\frac{1}{c}\int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1-1}}{(1+\theta(1+\mu))^n} e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))} e^{-b_2\mu}\mu^{a_2-1}[\prod_{i=1}^n(1+x_i)]$$
$$\exp[-\nu\frac{1+\theta(1+t)}{1+\theta(1+\mu)}e^{-\theta(t-\mu)}]d\theta d\mu \tag{5.16}$$

Then, considering the entropy loss function Δ_{GE}, We obtain the Bayes estimates of the parameters as

$$\hat{\theta}_{\mathrm{GE}} = \mathbb{E}[\theta^{-w}\mid X]^{-\frac{1}{w}} \tag{5.17}$$
$$\hat{\mu}_{\mathrm{GE}} = \mathbb{E}[\mu^{-w}\mid X]^{-\frac{1}{w}} \tag{5.18}$$

and,

$$\hat{R}_{GE}(t) = E[(\frac{1+\theta(1+t)}{1+\theta(1+\mu)})^{-w}e^{\theta w(t-\mu)}| X]^{-\frac{1}{w}} \tag{5.19}$$

The conditional expectation can be calculated as

$$\mathbb{E}[\theta^{-w}\mid X] = \frac{1}{c}\int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1-1-w}}{(1+\theta(1+\mu))^n} e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))} e^{-b_2\mu}\mu^{a_2-1}[\prod_{i=1}^n(1+x_i)]d\theta d\mu \tag{5.20}$$

$$\mathbb{E}[\mu^{-w}\mid X] = \frac{1}{c}\int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1-1}}{(1+\theta(1+\mu))^n} e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))} e^{-b_2\mu}\mu^{a_2-1-w}[\prod_{i=1}^n(1+x_i)]d\theta d\mu \tag{5.21}$$

$$E[(\frac{1+\theta(1+t)}{1+\theta(1+\mu)})^{-w}e^{\theta w(t-\mu)}|\,X] \tag{5.22}$$

$$= \frac{1}{c}\int_0^\infty\int_0^\infty \frac{\theta^{2n+a_1-1}}{(1+\theta(1+\mu))^n}e^{-\theta(b_1+\sum_{i=1}^n(x_i-\mu))}e^{-b_2\mu}\mu^{a_2-1}[\prod_{i=1}^n(1+x_i)]$$
$$(\frac{1+\theta(1+t)}{1+\theta(1+\mu)})^{-w}e^{w\theta(t-\mu)}]d\theta d\mu \tag{5.23}$$

The preceding Bayes estimators all rely on double integrals that defy simple closed-form solutions. Consequently, we turn to two popular approximation techniques in the following sections to derive approximate Bayes estimates for the parameters.

6 Lindley Approximation

This section ,we use Lindley technique for computing the Bayes estimates.Let h(θ,μ)be any function of the parameters.Then we have

$$\begin{aligned}\mathbb{E}[h(\theta,\mu)\,|\,x] &= \int_0^\infty\int_0^\infty h(\theta,\mu)\pi^*(\theta,\mu\,|\,x)d\theta d\mu \\ &= \frac{\int_0^\infty\int_0^\infty h(\theta,\mu)\exp(L(\theta,\mu;x)+\eta(\theta,\mu))d\theta d\mu}{\int_0^\infty\int_0^\infty \exp(L(\theta,\mu;x)+\eta(\theta,\mu))d\theta d\mu}\end{aligned} \tag{6.1}$$

The function $\eta(\theta,\mu) = \log\pi_1(\theta;a_1,b_1)+\log\pi_2(\mu;a_2,b_2)$, and $L(\theta,\mu;x)$ is the log-likelihood function of the parameters θ and μ given by,

$$L(X,\mu,\theta) = logl(X,\mu,\theta) = log(\frac{\theta^{2n}}{(1+\theta(1+\mu))^n}\prod_{i=1}^n(1+x_i)e^{-\theta\sum_{i=1}^n(x_i-\mu)}) \tag{6.2}$$

$$= 2nlog\theta - nlog(1+\theta(1+\mu)) + \sum_{i=1}^n\log(1+X_i) - \theta\sum_{i=1}^n(X_i-\mu) \tag{6.3}$$

The Bayes estimate of h(θ,μ) is approximated by

$$E[h(\theta,\mu)|x] \approx h(\theta,\mu) + \frac{1}{2}\sum_{i=1}^2\sum_{j=1}^2 h_{ij}\tau_{ij} + \sum_{i=1}^2\eta_iV_i + \frac{1}{2}\sum_{i=1}^2 L_{iii}\tau_{ii}V_i \tag{6.4}$$

$$+\frac{1}{2}\left(L_{112}(2\tau_{12}V_1+\tau_{11}V_2)+L_{122}(\tau_{22}V_1+2\tau_{12}V_2)\right), \tag{6.5}$$

where,

$$h_1 = \frac{\partial h}{\partial \theta}, h_2 = \frac{\partial h}{\partial \mu}, h_{11} = \frac{\partial^2 h}{\partial \theta^2}, h_{22} = \frac{\partial^2 h}{\partial \mu^2}, h_{12} = h_{21} = \frac{\partial^2 h}{\partial \theta \partial \mu}, \text{ and } V_r = \sum_j h_j \tau_{rj} \quad (6.6)$$

with τ_{ij} being (i,j)th elements of the inverse of the matrix $[-\frac{\partial^2 L}{\partial\theta\partial\mu}]$.Calculating all the expression in (6.5) at the MLEs of the parameters $\theta and \mu$, the approximate Bayes estimate of h(θ, μ) is obtained.

$$L_1 = \left.\frac{\partial L(\theta, \mu)}{\partial \theta}\right|_{(\hat{\theta},\hat{\mu})} = \frac{2n}{\theta} - \frac{n(1+\mu)}{1+\theta(1+\mu)} - \sum_{i=1}^{n}(x_i - \mu) \quad (6.7)$$

$$L_2 = \left.\frac{\partial L(\theta, \mu)}{\partial \mu}\right|_{(\hat{\theta},\hat{\mu})} = \frac{-n\theta}{1+\theta(1+\mu)} + n\theta \quad (6.8)$$

$$L_{11} = \left.\frac{\partial^2 L(\theta, \mu)}{\partial \theta^2}\right|_{(\hat{\theta},\hat{\mu})} = -\frac{2n}{\theta^2} + \frac{n(1+\mu)^2}{(1+\theta(1+\mu))^2} \quad (6.9)$$

$$L_{12} = L_{21} = \left.\frac{\partial^2 L(\theta, \mu)}{\partial \theta \partial \mu}\right|_{(\hat{\theta},\hat{\mu})} = \frac{-n}{(1+\theta(1+\mu))^2} + n \quad (6.10)$$

$$L_{22} = \left.\frac{\partial^2 L(\theta, \mu)}{\partial \mu^2}\right|_{\hat{\theta}=\hat{\theta},\mu=\hat{\mu}} = \frac{n\theta^2}{(1+\theta(1+\mu))^2} \quad (6.11)$$

$$L_{112} = \left.\frac{\partial^3 L(\theta, \mu)}{\partial \theta^2 \partial \mu}\right|_{\hat{\theta}=\hat{\theta},\mu=\hat{\mu}} = \frac{(1+\theta(1+\mu))^2 2n(1+\mu) - n(1+\mu)^2 2\theta(1+\theta(1+\mu))}{(1+\theta(1+\mu))^4} \quad (6.12)$$

$$L_{122} = \left.\frac{\partial^3 L(\theta, \mu)}{\partial \theta \partial \mu^2}\right|_{\hat{\theta}=\hat{\theta},\mu=\hat{\mu}} = \frac{(1+\theta(1+\mu))^2 2n\theta - 2n\theta^2(1+\mu)(1+\theta(1+\mu))}{(1+\theta(1+\mu))^4} \quad (6.13)$$

$$L_{111} = \left.\frac{\partial^3 L(\theta, \mu)}{\partial \theta^3}\right|_{\hat{\theta}=\hat{\theta},\mu=\hat{\mu}} = \frac{4n}{\theta^3} - \frac{2n(1+\mu)^3}{(1+\theta(1+\mu))^3} \quad (6.14)$$

$$L_{222}=\left.\frac{\partial^3 L(\theta,\mu)}{\partial\mu^3}\right|_{\hat\theta=\hat\theta,\mu=\hat\mu}=\frac{-2n\theta^3}{(1+\theta(1+\mu))^3} \tag{6.15}$$

$$\eta_1=\frac{a_1-1}{\hat\theta}-b_1, \tag{6.16}$$

$$\eta_2=\frac{a_2-1}{\hat\mu}-b_2 \tag{6.17}$$

In the following ,we derive the approximate Bayes estimates of θ,μ and R(t) under the considered loss functions(4.1),(4.3) and (4.5).

7 Lindley's Approximate Bayes Estimates Using Squared Error Loss Function

For computing the estimate of θ under SE loss function, let $h(\theta,\mu)=\theta$. Hence, $h_1=1$, $h_2=h_{11}=h_{21}=h_{12}=h_{22}=0$ and we have

$$\hat\theta_{\mathrm{SE}}\approx\hat\theta+\tau_{11}\eta_1+\tau_{21}\eta_2+\frac{1}{2}\left(L_{111}\tau_{11}^2+L_{222}\tau_{22}\tau_{12}+3L_{112}\tau_{12}\tau_{11}\right). \tag{7.1}$$

Similarly, setting $h(\theta,\mu)=\mu$, we have $h_2=1$, $h_1=h_{11}=h_{21}=h_{12}=h_{22}=0$. Therefore, the Bayes estimate of μ under SE loss function becomes

$$\hat\mu_{\mathrm{SE}}\approx\hat\mu+\tau_{22}\eta_2+\tau_{12}\eta_1+\frac{1}{2}\left(L_{111}\tau_{11}\tau_{12}+L_{222}\tau_{22}^2+L_{112}\left(\tau_{11}\tau_{22}+2\tau_{12}^2\right)\right). \tag{7.2}$$

Next, considering $h(\theta,\mu)=R(t)$, the approximate Bayes estimate of $R(t)$ against loss function SE is obtained as

$$\begin{aligned}\hat R_{\mathrm{SE}}(t)\approx{}&\hat R(t;\theta,\mu)+0.5(R_{11}\tau_{11}+R_{22}\tau_{22}+2R_{12}\tau_{12})+\eta_1V_1+\eta_2V_2\\&+0.5\left(L_{111}\tau_{11}V_1+L_{222}\tau_{22}V_2+L_{112}(2\tau_{12}V_1+\tau_{11}V_2)\right),\end{aligned} \tag{7.3}$$

$$V_1=R_1\tau_{11}+R_2\tau_{12}, \tag{7.4}$$

$$V_2=R_1\tau_{12}+R_2\tau_{22} \tag{7.5}$$

$$R_1=\frac{\partial R(t,\theta,\mu)}{\partial\mu}=\frac{(1+\theta(1+t))}{e^{t\theta}}\frac{\theta^2(1+\mu)e^{\theta\mu}}{(1+\theta(1+\mu))^2} \tag{7.6}$$

$$R_2=\frac{\partial R(t,\theta,\mu)}{\partial\theta}=\frac{\theta(\mu^2+2\mu-t^2-2t)+\theta^2(\mu+\mu^2+\mu^2t-t-t^2-\mu t^2)e^{-\theta(t-\mu)}}{(1+\theta(1+\mu))^2} \tag{7.7}$$

$$\begin{aligned}R_{11}={}&\frac{\partial^2 R(t,\theta,\mu)}{\partial\mu^2}=\frac{1+\theta(1+t)}{e^{t\theta}}\\&\times\frac{(1+\theta(1+\mu))^2(\theta^2e^{\theta\mu}+e^{\theta\mu}\theta^3(1+\mu))-2\theta^3(1+\mu)e^{\theta\mu}(1+\theta(1+\mu))}{(1+\theta(1+\mu))^4}\end{aligned} \tag{7.8}$$

$$
\begin{aligned}
R_{22} &= \frac{\partial^2 R(t,\theta,\mu)}{\partial\theta^2} \\
&= \frac{\theta(\mu^2+2\mu-t^2-2t)+\theta^2(\mu+\mu^2+\mu^2 t-t-t^2-\mu t^2)e^{-\theta(t-\mu)}(\mu-t)(1+\theta(1+\mu))^2}{(1+\theta(1+\mu))^4} \\
&+ \frac{(1+\theta(1+\mu))^2 e^{-\theta(t-\mu)}\theta(-2t-2t^2-2t^2\mu+2\mu+2\mu^2+2\mu^2 t)+(-2t-t^2+2\mu+\mu^2)}{(1+\theta(1+\mu))^4} \\
&- \frac{2(\theta(\mu^2+2\mu-t^2-2t)+\theta^2(\mu+\mu^2+\mu^2 t-t-t^2-\mu t^2))e^{-\theta(t-\mu)}(1+\mu)(1+\theta(1+\mu))}{(1+\theta(1+\mu))^4}
\end{aligned}
\tag{7.9}
$$

$$
\begin{aligned}
R_{12} &= \frac{\partial^2 R(t,\theta,\mu)}{\partial\theta\partial\mu} \\
&= \frac{(1+\theta(1+t))}{e^{t\theta}} \\
&\times \frac{((1+\theta(1+\mu))^2(2\theta e^{\theta\mu}(1+\mu)+\theta^2 e^{\theta\mu}\mu(1+\mu))-\theta^2(1+\mu)e^{\theta\mu}2((1+\theta(1+\mu))(1+\mu))}{(1+\theta(1+\mu))^4} \\
&+ \frac{(\theta^2 e^{\theta\mu}(1+\mu))}{(1+\theta(1+\mu))^2}\frac{((1+t)e^{t\theta}+(1+\theta(1+t))te^{t\theta})}{(e^{t\theta})^2}
\end{aligned}
\tag{7.10}
$$

8 Lindley's Approximate Bayes Estimates Using Linex Loss Function

The approximate Bayes estimate of θ under the LE loss function is obtained by choosing $h(\theta,\mu)=e^{-\nu\theta}$. Thus, we have:

$$
\begin{aligned}
\hat{\theta}_{LE} \approx -\frac{1}{\nu}\log(&e^{-\nu\hat{\theta}}+0.5h_{11}\tau_{11}+h_1(\tau_{11}\eta_1+\tau_{21}\eta_2) \\
&+0.5h_1(L_{111}\tau_{11}^2+L_{222}\tau_{22}\tau_{12}+3L_{112}\tau_{12}\tau_{11})),
\end{aligned}
\tag{8.1}
$$

where $h_1=-\nu e^{-\nu\hat{\theta}}$ and $h_{11}=\nu^2 e^{-\nu\hat{\theta}}$.

Similarly, by choosing $h(\theta,\mu)=e^{-\nu\mu}$, we have:

$$
\begin{aligned}
\hat{\mu}_{LE} \approx -\frac{1}{\nu}\log(&e^{-\nu\hat{\mu}}+0.5h_{22}\tau_{22}+h_2(\tau_{22}\eta_2+\tau_{12}\eta_1) \\
&+0.5h_2(L_{111}\tau_{11}\tau_{12}+L_{222}\tau_{22}^2+L_{112}(\tau_{11}\tau_{22}+2\tau_{12}^2))),
\end{aligned}
\tag{8.2}
$$

where $h_2=-\nu e^{-\nu\hat{\mu}}$ and $h_{22}=\nu^2 e^{-\nu\hat{\mu}}$. To evaluate the estimate of R(t) under linex loss,

Let,

$$
\begin{aligned}
h(\theta,\mu) &= e^{-\nu R(t)}, h_1=-\nu e^{-\nu\hat{R}(t;\theta,\mu)}R_1, h_2=-\nu e^{-\nu\hat{R}(t;\theta,\mu)}R_2, \\
h_{11} &= \nu e^{-\nu\hat{R}(t;\theta,\mu)}(\nu R_1^2-R_{11}), h_{22}=\nu e^{-\nu\hat{R}(t;\theta,\mu)}(\nu R_2^2-R_{22}), \\
h_{12} &= \nu e^{-\nu\hat{R}(t;\theta,\mu)}(\nu R_1R_2-R_{12}).
\end{aligned}
\tag{8.3}
$$

Then, we have

$$\hat{R}_{LE}(t;\theta,\mu) \approx -\frac{1}{\nu}\log(e^{-\nu\hat{R}(t;\theta,\mu)} + 0.5(h_{11}\tau_{11} + h_{22}\tau_{22} + 2h_{12}\tau_{12}) \\ + V_1\eta_1 + V_2\eta_2 + 0.5(L_{111}\tau_{11}V_1 + L_{222}\tau_{22}V_2 + L_{112}(2\tau_{12}V_1 + \tau_{11}V_2))), \quad (8.4)$$

where R_1, R_2, R_{11}, R_{22}, and R_{12} are given by (7.6)-(7.10), respectively.

9 Lindley's Approximate Bayes Estimates Using General Entropy Loss function

Setting $h(\theta,\mu) = \theta^{-\omega}$, the approximate Bayes estimate of θ under the GE loss function is obtained as:

$$\hat{\theta}_{GE} \approx [\hat{\theta}^{-\omega} + 0.5h_{11}\tau_{11} + h_1(\tau_{11}\eta_1 + \tau_{12}\eta_2) + 0.5h_1\left(L_{111}\tau_{11}^2 + L_{222}\tau_{12}\tau_{22} + 3L_{112}\tau_{12}\tau_{11}\right)]^{-\frac{1}{\omega}}, \quad (9.1)$$

where $h_1 = -\omega\theta^{-\omega-1}$ and $h_{11} = \omega(\omega+1)\theta^{-\omega-2}$.

Similarly, for the parameter μ, we use $h(\theta,\mu) = \mu^{-\omega}$ and obtain:

$$\hat{\mu}_{GE} \approx [\hat{\mu}^{-\omega} + 0.5h_{22}\tau_{22} + h_2(\tau_{22}\eta_2 + \tau_{12}\eta_1) \\ + 0.5h_2\left(L_{111}\tau_{11}\tau_{12} + L_{222}\tau_{22}^2 + L_{112}(\tau_{11}\tau_{22} + \tau_{12}^2)\right)]^{-\frac{1}{\omega}}, \quad (9.2)$$

where $h_2 = -\omega\mu^{-\omega-1}$ and $h_{22} = \omega(\omega+1)\mu^{-\omega-2}$.

Now, the reliability $R(t)$ can be estimated by considering $h(\theta,\mu) = R(t)^{-\omega}$. Then, we have:

$$\hat{R}_{GE}(t;\theta,\mu) \approx [\hat{R}^{-\omega}(t;\theta,\mu) + 0.5(h_{11}\tau_{11} + h_{22}\tau_{22} + 2h_{12}\tau_{12}) + V_1\eta_1 + V_2\eta_2 \quad (9.3)$$

$$+0.5\left(L_{111}\tau_{11}V_1 + L_{222}\tau_{22}V_2 + L_{112}(2\tau_{12}V_1 + \tau_{11}V_2)\right)]^{-\frac{1}{\omega}} \quad (9.4)$$

where

$$h_1 = -\omega R(t)^{-\omega-1}R_1, \quad (9.5)$$
$$h_2 = -\omega R(t)^{-\omega-1}R_2, \quad (9.6)$$
$$h_{11} = \omega(\omega+1)R(t)^{-\omega-2}R_1^2 - \omega R(t)^{-\omega-1}R_{11}, \quad (9.7)$$
$$h_{22} = \omega(\omega+1)R(t)^{-\omega-2}R_2^2 - \omega R(t)^{-\omega-1}R_{22}, \quad (9.8)$$
$$h_{12} = \omega(\omega+1)R(t)^{-\omega-2}R_1R_2 - \omega R(t)^{-\omega-1}R_{12}. \quad (9.9)$$

10 Simulation Study

The Shifted Lindley distribution (SL) presents challenges in generating samples directly through the inversion of its Cumulative Distribution Function (CDF). However, research has demonstrated that SL(μ,θ) can be represented as a mixture of two shifted gamma distributions. Leveraging this property, a convenient sampling scheme is proposed for generating random samples from SL distributions using R programming. Specifically, the rgamma3 function from the FAdist package is utilized to generate samples from shifted gamma distributions. This summary outlines the steps involved in the sampling scheme, offering a pragmatic approach for data generation in simulation studies.
To estimate the parametersμ and θ, We have generated samples from the Shifted Lindley distribution.Then, using the maximum likelihood estimate for estimate the parameter μ and θ

10.1 Algorithm for Data Simulation

1. Select values of θ and μ
2. Calculate weight

$$w = \frac{(1+\mu)\theta}{1+\theta(1+\mu)}$$

3.Generate U from U(0,1)
4.If $U < w$, generate a sample from $f_{SG}(x; 1, \theta, \mu)$ else from $f_{SG}(x; 2, \theta, \mu)$
using rgamma3(1,shape,scale,thres),

In the earlier segments, we utilized Bayesian techniques to derive estimates for the parameters θ,μ and the reliability function R(t). By exploring various loss functions, we derived Bayes estimators and obtained approximate estimates through both Lindley's method and Gibbs sampling. This approach allowed us to examine different perspectives while refining our parameter estimations.

We have used two sets of parameter values,

- Set 1: $(\theta, \mu) = (2, 1)$
- Set 2: $(\theta, \mu) = (1.5, 1.5)$

Choices for ν and w:

- ν: -0.5, 1, 1.5
- w: -0.5, 1, 1.5

To evaluate the Bayes estimates, we take two different sets of hyper-parameter values as- a_1=a_2=b_1=b_2=2.

Then, approximate Bayes estimates of the parameter θ,μ and R(t) are computed by using Lindley method.Note that all the ML and Bayes estimators of R(t) are evaluated at t=1. All the calculations are conducted using R programming.

Table 1: θ=2,μ=1

n		MLE	L_{SE}	L_{LE} ν = -0.5	ν=1	ν=1.5	L_{GE} ω =-0.5	ω =1	ω=1.5
15	$\hat{\theta}$	1.9209	1.9695	1.9715	1.9652	1.9628	1.9684	1.9650	1.9638
	$\hat{\mu}$	1.0169	1.1419	1.1321	1.1649	1.1791	1.1523	1.1927	1.2107
	$\hat{R(t)}$	1.0261	0.8464	0.8609	0.6288	0.6426	0.9495	0.7648	0.7921
20	$\hat{\theta}$	2.0001	2.0364	2.0381	2.0329	2.0310	2.0356	2.0329	2.0319
	$\hat{\mu}$	1.0169	1.1099	1.1037	1.1242	1.1324	1.1166	1.1405	1.1503
	$\hat{R(t)}$	1.0274	0.8880	0.9010	0.7039	0.7101	0.9697	0.8110	0.8327
30	$\hat{\theta}$	2.2666	2.2902	2.2915	2.2877	2.2863	2.2897	2.2880	2.2874
	$\hat{\mu}$	1.0121	1.0732	1.0701	1.0798	1.0834	1.0763	1.0870	1.0911
	$\hat{R(t)}$	1.0227	0.9203	0.9321	0.7720	0.7724	0.9836	0.8520	0.8680
50	$\hat{\theta}$	2.0755	2.0896	2.0903	2.0879	2.0873	2.0892	2.0880	2.0877
	$\hat{\mu}$	1.0004	1.0373	1.0353	1.0413	1.0434	1.0392	1.0456	1.0480
	$\hat{R(t)}$	1.0006	0.9441	0.9508	0.8618	0.8592	0.9749	0.9028	0.9107
70	$\hat{\theta}$	1.9804	1.9905	1.9910	1.9893	1.9888	1.9902	1.9894	1.9891
	$\hat{\mu}$	1.0003	1.0269	1.0254	1.0298	1.0314	1.0284	1.0330	1.0346
	$\hat{R(t)}$	1.0005	0.9617	0.9663	0.9041	0.9012	0.9822	0.9317	0.9371

From table 1 it can be seen that the Bayes estimates of θ is almost similar under different loss function. Similar feature can be seen for Bayes estimate of μ. It can also be seen that the absolute bias in case of MLE of μ is getting reduced as sample size increases. Bayes estimates of $R(1)$ is getting close to 1 as the sample size increased.The MLE always overestimates the $R(1)$ value where the Bayes estimates under all loss functions, underestimates $R(1)$. The smallest estimate of $R(1)$ is 0.7921 which is achieved when genrralized entropy loss function was used with $\omega = 1.5$.

Larger sample sizes result in more stable and accurate estimates across all methods. The L_{GE}estimates (with higher ωvalues) tend to give slightly higher estimates compared to others, especially for smaller sample sizes. For larger sample sizes, all methods converge towards the true values, but $L_{GE}$with ω=1.5 shows slightly higher variability.

From table 2 it can be seen that all methods provide relatively accurate estimates of θ and μ,with slight overestimations. Larger sample sizes result in more stable and accurate estimates across all methods. MLE and L_{SE} estimators provide close and consistent estimates for θ and μ. L_{GE} estimators with higher values of ω and ν provide higher estimates for μ and R(t) compared to other methods. L_{SE} seems to offer reliable estimates across all sample sizes, particularly for θ. No single estimator is universally best for all parameters and sample sizes. For smaller sample sizes,L_{SE} and MLE provide more reliable estimates. For larger sample sizes, all methods converge towards the true values, but L_{GE} with ω=1 or 1.5 shows slightly higher variability. Overall, while MLE and L_{SE} show consistent performance,L_{GE} with adjusted ν and ω parameters can provide useful estimates, especially when tailored to specific sample sizes and conditions.

Table 2: θ=1.5,μ=1.5

n		MLE	L_{SE}	L_{LE} ν=-0.5	ν=1	ν= 1.5	L_{GE} ω=-0.5	ω= 1	ω=1.5
15	$\hat{\theta}$	1.4518	1.5289	1.5290	1.5285	1.5281	1.5288	1.5282	1.5279
	$\hat{\mu}$	1.5225	1.6717	1.6559	1.7108	1.7362	1.6831	1.7249	1.7425
	$\hat{R(t)}$	1.7879	1.5698	1.4495	0.7040	0.6702	2.0913	1.0655	1.1587
20	$\hat{\theta}$	1.5114	1.5693	1.5697	1.5683	1.5678	1.5690	1.5680	1.5677
	$\hat{\mu}$	1.5225	1.6318	1.6216	1.6558	1.6701	1.6391	1.6641	1.6740
	$\hat{R(t)}$	1.8414	1.6678	1.5709	0.8817	0.8126	2.0921	1.1942	1.2730
30	$\hat{\theta}$	1.7122	1.7506	1.7511	1.7495	1.7489	1.7503	1.7493	1.7489
	$\hat{\mu}$	1.5161	1.5847	1.5798	1.5956	1.6017	1.5881	1.5994	1.6035
	$\hat{R(t)}$	2.0170	1.8764	1.7878	1.1185	0.9928	2.2484	1.3971	1.4591
50	$\hat{\theta}$	1.5684	1.5912	1.5916	1.5905	1.5900	1.5910	1.5903	1.5900
	$\hat{\mu}$	1.5004	1.5432	1.5399	1.5501	1.5538	1.5455	1.5526	1.5551
	$\hat{R(t)}$	1.8426	1.7709	1.7367	1.3433	1.2376	1.9406	1.5126	1.5884
70	$\hat{\theta}$	1.4968	1.5131	1.5134	1.5125	1.5122	1.5129	1.5123	1.5121
	$\hat{\mu}$	1.5004	1.5318	1.5293	1.5369	1.5397	1.5333	1.5387	1.5405
	$\hat{R(t)}$	1.7810	1.7332	1.7128	1.4343	1.3473	1.8424	1.5561	1.5811

11 Data Analysis

The application of the different methods to real data.Let us consider the following data set from Maiti et al. (2021).The data consists of measurements on strength of the sintered silicon nitride.The data is

613.9,623.4,639.3,642.1,653.8,662.4,669.5,672.8,681.3,682.0,699.0,714.5,717.4,725.5, 741.6,744.9,751.0,761.7,763.9,774.2,791.6,795.2,829.8,838.4,856.4,868.3,882.9

Here, considering various loss functions, we compute the approximate Bayes estimates of the unknown parameters. First, from the above data set,we obtained the ML estimates of the parameters and the MLE of the reliability R(t;μ,θ) at t=1,1.5 are computed.All the hyper-Parameters are considered to be 2 and three values -0.5,1,1.5 are assumed for both ν and ω. The values use only for calculating linex and entropy loss functions. Then, the Bayes estimates of the interested parameters under SE,LE and GE loss functions are calculated using Lindely's approximation.

From the table 3, it can be seen that, maximum likelihood estimates of θ , μ are almost similar with the estimates under under loss functions. When t=1,all values of $R(t)$ are approximately same except L_{LE} with ν=1,ν=1.5 . Similarly, for t=1.5, similar observation can be seen under Linex loss function with ν=1 and ν = 1.5. In both the cases, the estimates of $R(t)$ is small pretty small compared to maximum likelihood estimates.

Table 3

		θ	μ	R(t) $t=1$	$t=1.5$
MLE		1.0897	1.1391	1.1106	0.7544
L_{SE}		1.1222	1.2332	1.1356	0.7706
	ν=-0.5	1.1229	1.2193	1.1398	0.7837
L_{LE}	ν= 1	1.1206	1.2653	0.9859	0.6542
	ν=1.5	1.1197	1.2846	0.9746	0.6548
	ω =-0.5	1.1214	1.2464	1.1854	0.8048
L_{GE}	ω =1	1.1191	1.2948	1.0557	0.7201
	ω=1.5	1.1184	1.3158	1.071	0.7316

12 Conclusion

In this work, efforts were made to get Bayesian inference on a newly derived distribution based on different loss functions. The simulation studies showed satisfactory results interm of biases and sd values. From the data analysis, the estimates are found to be reasonable one. Comparing the approximate method with MCMC under the present set up will also be interesting which will be pursued in future.

13 Appendix: R Codes to Perform Analysis Reported in This Document

```
##install.packages("PearsonDS")
##install.packages("nleqslv")

library(PearsonDS)
library(nleqslv)

n1=50###change
mu=1.5### Change
theta=1.5###Change
count1=0;
count2=0;
a=matrix(0,1,n1);
b=matrix(0,1,n1);
w1=((1+mu)*theta)/(1+theta*(1+mu))

for(i in 1:n1)
{
  if(runif(1)<w1)
  {
count1=count1+1
a[count1]=rpearsonIII(1, shape=1, location=mu, scale=1/theta)
  } else
  {
count2=count2+1
b[count2]=rpearsonIII(1, shape=2, location=mu, scale=1/theta)
  }
}
c1=c(a,b)
d1=c1[c1!=0]
d1
min_value <- min(d1)
print(min_value)
mu1=min_value
mu1
message <- sprintf("The M.L.E of mu1 is = %f", min_value)
print(message)

########
samp=d1
```

```
m31=min_value
x=n1*mu1*(1+mu1)-sum(d1)*(1+mu1)
x
y=n1+n1*mu1-sum(d1)+n1*mu1
y
# Define the coefficients
a <- x
b <- y
c <- 2 * n1

# Calculate the discriminant
discriminant <- b^2 - 4 * a * c
discriminant

# Check if there are real roots
if (discriminant < 0) {
  print("No real roots")
} else if (discriminant == 0) {
  # Calculate the single real root
  root <- -b / (2 * a)
  print(paste("Single real root:", root))
} else {
  # Calculate two real roots
  root1 <- (-b + sqrt(discriminant)) / (2 * a)
  root2 <- (-b - sqrt(discriminant)) / (2 * a)
  print(paste("Root 1:", root1))
  print(paste("Root 2:", root2))

}
# Select the positive root
if (root1 > 0) {
  positive_root <- root1
} else {
  positive_root <- root2
}
# Print the positive root
print(positive_root)
theta1=positive_root
theta1
```

```
t=1
###true survival function
R_t_MLE <- function(t, theta1, mu1) {
  result <- ((1 + theta1 * (1 + t)) / (1 + theta1 * (1 + mu1)))
   * exp(-theta1* (t - mu1))
}
result_value_11<- R_t_MLE(t,theta1,mu1)
print(result_value_11)
R_t_MLE=result_value_11
R_t_MLE

#####We take the value of theta1 is 1.133, since theta1 is
always positive
L11=-(2*n1/theta1^2)+((n1*(1+mu1)^2)/(1 + theta1 *
(1 + mu1))^2)
L11
L22=((n1 * theta1^2) / (1 + theta1 * (1 + mu1))^2)
L22
##The value of L12=L21
L21=-(n1/(1 + theta1 * (1 + mu1))^2)+n1
L21
L12=-(n1/(1 + theta1 * (1 + mu1))^2)+n1
L12

L111=(4*n1/theta1^3)-((2*n1*(1+mu1)^3)/(1+theta1*(1+mu1))^3)
L111
L222 <- -((2 * n1 * theta1^3) / (1 + theta1 * (1 + mu1))^3)
L222

L112 <- ((2 * n1 * (1 + mu1) * (1 + theta1 * (1 + mu1))^2)
- (2 * n1 * theta1 * (1 + mu1)^2 * (1 + theta1 * (1 + mu1))))
/ ((1 + theta1 * (1 + mu1))^4)
L112

# Create the matrix
matrix<- matrix(c(-L11, -L12, -L21, -L22), nrow = 2,
byrow = TRUE)

# Print the matrix
print(matrix)

# Find the inverse of the matrix
inverse_matrix <- solve(matrix)
```

```
# Print the inverse matrix
print(inverse_matrix)
T11=inverse_matrix[1,1]
T12=inverse_matrix[1,2]
T21=inverse_matrix[2,1]
T22=inverse_matrix[2,2]

#########change

###T11=0.0001797935
##T12=-0.001927182
##T21=-0.001927182
##T22=-0.002337156
suppressWarnings({a1 <- a2 <- b1 <- b2 <- 2})
eta1=((a1-1)/theta1)-b1
eta2=((a2-1)/mu1)-b2


##The squared error loss function of theta is
theta_SE=theta1+T11*eta1+T21*eta2+0.5*(L111*T11^2+L222*T22*T12
+3*L112*T12*T11)
theta_SE
mu_SE <- mu1 + T22 * eta2 + T12 * eta1 + 0.5 * (L111 *
T11 * T12
+ L222 * T22^2 + L112 * (T11 * T22 + 2 * T12^2))
mu_SE
##############
##Reliability Function (The ML and Bayes estimators of R(t)are
evaluated at t=1)
R <- function(t, theta1, mu1) {
  result <- ((1 + theta1 * (1 + t)) / (1 + theta1 * (1 + mu1)))
  * exp(-theta1 * (t - mu1))
}
result_value1 <- R(t,theta1,mu1)
print(result_value1)
R=result_value1
R
#########################
R1 <- function(t, theta1, mu1) {
  numerator <- (1 + theta1 * (1 + t)) * (theta1^2 * (1 + mu1)
  * exp(mu1 * theta1))
  denominator <- exp(t * theta1) * (1 + theta1 * (1 + mu1))^2
```

```
  result2 <- numerator / denominator

  return(result2)
}
result2 <- R1(t,theta1, mu1)
print(result2)
R1=result2
R1
##############
R11 <- function(t, theta1, mu1) {
  numerator1 <- (1 + theta1 * (1 + t))
  denominator1 <- exp(t * theta1)

  numerator2 <- (1 + theta1 * (1 + mu1))^2 *
  (theta1^2 * exp(theta1 * mu1) + exp(theta1 * mu1) * theta1^3
  * (1 + mu1)) - 2 * theta1^3 * (1 + mu1) * exp(theta1*mu1)
  * (1 + theta1 * (1 + mu1))
  denominator2 <- (1 + theta1* (1 + mu1))^4

  result3 <- (numerator1 / denominator1) *
  (numerator2 / denominator2)

  return(result3)
}
result3 <-R11(t, theta1,mu1)
print(result3)
R11=result3
R11

######################
R12<- function(t, theta1, mu1) {
  numerator1 <- (1 + theta1* (1 + t))
  denominator1 <- exp(t * theta1)
  numerator2 <- ((1 + theta1* (1 + mu1))^2) *
  (2 * theta1* exp(theta1* mu1) * (1 + mu1) +
  theta1^2 * exp(theta1* mu1) * mu1* (1 + mu1)) -
    theta1^2 * (1 + mu1) * exp(theta1* mu1) *
    2 * ((1 + theta1* (1 + mu1)) * (1 + mu1))
  denominator2 <- (1 + theta1* (1 + mu1))^4

  numerator3 <- (theta1^2 * exp(theta1* mu1) * (1 + mu1))
  denominator3 <- (1 + theta1* (1 + mu1))^2
```

```
  numerator4 <- ((1 + t) * exp(t * theta1) +
  (1 + theta1 * (1 + t)) * t * exp(t * theta1))
  denominator4 <- (exp(t * theta1))^2

  result4 <- ((numerator1 / denominator1) *
  (numerator2 / denominator2))+
  ((numerator3 / denominator3) * (numerator4 / denominator4))

  return(result4)
}
result4 <- R12(t,theta1, mu1)
print(result4)
R12=result4
R12
###############
R2<- function(t, theta1, mu1) {
numerator <- theta1 * (mu1^2 + 2 * mu1 - t^2 - 2 * t) +
theta1^2 * (mu1 + mu1^2 +
mu1^2 * t - t - t^2 - mu1 * t^2) *
exp(-theta1 * (t - mu1))
denominator <- (1 + theta1 * (1 + mu1))^2

  result5 <- numerator / denominator

  return(result5)
}
result5 <-R2(t,theta1,mu1)
print(result5)
R2=result5
R2
##############
R22<- function(t, theta1, mu1) {
term1_numerator <- theta1 * (mu1^2 + 2 * mu1 - t^2 - 2 * t) +
theta1^2 * (mu1 + mu1^2 + mu1^2 * t - t - t^2 - mu1 * t^2) *
exp(-theta1 * (t - mu1)) * (mu1 - t) *
(1 + theta1 * (1 + mu1))^2
term1_denominator <- (1 + theta1 * (1 + mu1))^4

term2_numerator <- (1 + theta1 * (1 + mu1))^2 *

exp(-theta1*(t - mu1)) *theta1 * (-2 * t - 2 * t^2 - 2 *
```

```
t^2 * mu1 + 2 * mu1 +
2 * mu1^2 + 2 * mu1^2 * t) + (-2 * t - t^2 + 2 * mu1 + mu1^2)
term2_denominator <- (1 + theta1 * (1 + mu1))^4
term3_numerator <- 2 * (theta1 * (mu1^2 + 2 * mu1 - t^2 -
2 * t) +
theta1^2 * (mu1 + mu1^2 + mu1^2 * t - t - t^2 - mu1* t^2)) *
exp(-theta1 * (t - mu1)) * (1 + mu1) * (1 + theta1 * (1 + mu1))
term3_denominator <- (1 + theta1 * (1 + mu1))^4
result6 <- (term1_numerator / term1_denominator) +
(term2_numerator / term2_denominator) -
(term3_numerator / term3_denominator)
return(result6)
}
result6 <- R22(t,theta1,mu1)
print(result6)
R22=result6
R22
##################################################
v1=R1*T11+R2*T12
v1
v2=R1*T12+R2*T22
v2
####Squared errorloss function
####
R_SE=R+0.5*(R11*T11+R22*T22+2*R12*T12)+eta1*v1+eta2*v2+
0.5*(L111*T11*v1+L222*T22*v2+L112*(2*T12*v1+T11*v2))
R_SE

#### Llnex loss function#######
nu=1######change#########
h1_LE= -nu*exp(-nu*theta1)
h11_LE=(nu)^2*exp(-nu*theta1)

T_LE <- function(nu,theta1, h11_LE, T11, h1_LE,
eta1,eta2, L111, L222, L112, T22,T12){
  result12 <- -1/nu * log(exp(-nu*theta1) +
  0.5 * h11_LE * T11 + h1_LE *
(T11 *eta1 + T21 * eta2) + 0.5 * h1_LE *
(L111 * T11^2 + L222 * T22 * T12 + 3 * L112 * T12 * T11))
  return(result12)
}
result12<-T_LE(nu,theta1, h11_LE, T11, h1_LE,
eta1,eta2, L111, L222, L112, T22,T12)
```

```
print(result12)
theta_LE=result12
theta_LE
##########################################
h2_LE= -nu*exp(-nu*mu1)
h22_LE=(nu)^2*exp(-nu*mu1)

m_LE <- function(nu,mu1, h22_LE, T11, h2_LE,
eta1,eta2, L111, L222, L112, T22,T12){
result22 <- -1/nu * log(exp(-nu*mu1) + 0.5 * h22_LE * T22 +
h2_LE *
(T22 *eta2 + T12* eta1) + 0.5 * h2_LE * (L111 * T11*T12 +
L222 * T22^2  + L112*(T11*T22+2*T12^2)))
return(result22)
}
result22<-m_LE(nu,mu1, h22_LE, T11, h2_LE,
eta1,eta2, L111, L222, L112, T22,T12)
print(result22)
mu_LE=result22
mu_LE
######################################################
h_1_LE=-nu*exp(-nu*R)*R1
h_2_LE=-nu*exp(-nu*R)*R2
h_11_LE=nu*exp(-nu*R)*(nu*R1^2-R11)
h_22_LE=nu*exp(-nu*R)*(nu*R2^2-R22)
h_12_LE=nu*exp(-nu*R)*(nu*R1*R2-R12)

R_LE <- function(nu,R,h_11_LE,
T11,h_22_LE,h_12_LE,eta1,eta2, v1,v2,L111, L222, L112,
T22,T12){
  result33<- -1/nu * log(exp(-nu*R) +
0.5 * (h_11_LE *T11 + h_22_LE*T22+2 * h_12_LE *T12) +
v1 * eta1 + v2 * eta2 +
0.5 * (L111 * T11 * v1 + L222 *T22 * v2 + L112 *
(2 *T12 * v1 +T11 * v2)))
return(result33)
}
result33<-R_LE(nu,R,h_11_LE, T11,h_22_LE,h_12_LE,eta1,eta2,
v1,v2,L111, L222, L112, T22,T12)
print(result33)
R_LE=result33
R_LE
```

```
######################
###true survival function
R_t <- function(t, theta, mu) {
 result <- ((1 + theta * (1 + t)) / (1 + theta * (1 + mu)))
* exp(-theta* (t - mu))
}
 result_value_11<- R_t(t,theta,mu)
 print(result_value_11)
R_t=result_value_11
R_t
####################
##General entropy loss function
w=1##################change#
h1_GE= -w*theta1^(-w-1)
h11_GE=w*(w+1)*theta1^(-w-2)

T_GE <- function(w,theta1, h11_GE, T11, h1_GE,eta1,eta2,
L111, L222, L112, T22,T12){
 result_12 <- (theta1^-w+ 0.5 * h11_GE * T11 + h1_GE * (T11
*eta1 + T21 * eta2) +
 0.5 * h1_GE * (L111 * T11^2 + L222 * T22 * T12 +
 3 * L112 * T12 * T11))^(-1/w)
 return(result_12)
}
 result_12<-T_GE(w,theta1, h11_GE, T11, h1_GE,eta1,eta2,
L111, L222, L112, T22,T12)
 print(result_12)
 theta_GE=result_12
 theta_GE
########################################
h2_GE=-w*mu1^(-w-1)
h22_GE=w*(w+1)*mu1^(-w-2)

m_GE <- function(w,mu1, h22_GE, T11, h2_GE,eta1,eta2,
L111,L222, L112, T22,T12){
 result_22 <-(mu1^-w + 0.5 * h22_GE * T22 + h2_GE *
(T22 *eta2 + T12* eta1) + 0.5 * h2_GE *
(L111 * T11*T12 + L222 * T22^2  + L112*
(T11*T22+2*T12^2)))^(-1/w)
  return(result_22)
}
 result_22<-m_GE(w,mu1, h22_GE, T11, h2_GE,eta1,eta2, L111,
```

```
L222, L112, T22,T12)
print(result_22)
mu_GE=result_22
mu_GE
##################################
h_1_GE=-w*R^(-w-1)*R1
h_2_GE=-w*R^(-w-1)*R2
h_11_GE=w*(w+1)*R^(-w-2)*R1^2-w*R^(-w-1)*R11
h_22_GE=w*(w+1)*R^(-w-2)*R2^2-w*R^(-w-1)*R22
h_12_GE=w*(w+1)*R^(-w-2)*R1*R2-w*R^(-w-1)*R12

R_GE <- function(w,R,h_11_GE,

T11,h_22_LE,h_12_LE,eta1,eta2, v1,v2,
L111, L222, L112, T22,T12){
result_333<- (R^-w+ 0.5 * (h_11_GE *T11 + h_22_GE*T22+
2 * h_12_GE *T12) +
v1 * eta1 + v2 * eta2 +
0.5 * (L111 * T11 * v1 + L222 *T22 * v2 + L112 *
(2 *T12 * v1 +T11 * v2)))^(-1/w)
  return(result_333)
}
result_333<-R_GE(w,R,h_11_LE,

T11,h_22_LE,h_12_LE,eta1,eta2, v1,v2,L111, L222, L112, T22,T12)
print(result_333)
R_GE=result_333
R_GE
#################
n1
######################
####For MLE
theta1
mu1
R_t_MLE
#####For squared error loss function
theta_SE
mu_SE
R_SE
############For Linex loss function
nu
theta_LE
mu_LE
```

```
R_LE
#####FOR General entropy Loss function
w
theta_GE
mu_GE
R_GE
```

R Code For Data Analysis

```
library(PearsonDS)
library(nleqslv)
set.seed(23)
n1=27
d1=c(613.9,623.4,639.3,642.1,653.8,662.4,669.5,672.8,681.3,
682.0,699.0,714.5,717.4,725.5,741.6,744.9,751.0,761.7,
763.9,774.2,791.6,795.2,829.8,838.4,856.4,868.3,882.9)
d1
min_value <- min(d1)
print(min_value)
mu1=min_value
mu1
message <- sprintf("The M.L.E of mu1 is = %f", min_value)
print(message)
###############
samp=d1
m31=min_value
x=n1*mu1*(1+mu1)-sum(d1)*(1+mu1)
x
y=n1+n1*mu1-sum(d1)+n1*mu1
y
# Define the coefficients
a <- x
b <- y
c <- 2 * n1

# Calculate the discriminant
discriminant <- b^2 - 4 * a * c
discriminant

# Check if there are real roots
if (discriminant < 0) {
  print("No real roots")
} else if (discriminant == 0) {
```

```
  # Calculate the single real root
  root <- -b / (2 * a)
  print(paste("Single real root:", root))
} else {
  # Calculate two real roots
  root1 <- (-b + sqrt(discriminant)) / (2 * a)
  root2 <- (-b - sqrt(discriminant)) / (2 * a)
  print(paste("Root 1:", root1))
  print(paste("Root 2:", root2))

}
# Select the positive root
if (root1 > 0) {
  positive_root <- root1
} else {
  positive_root <- root2
}
# Print the positive root
print(positive_root)
theta1=positive_root
theta1
################
t=1
############
###true survival function
R_t_MLE <- function(t, theta1, mu1) {
  result <- ((1 + theta1 * (1 + t)) / (1 + theta1 * (1 + mu1)))
  * exp(-theta1* (t - mu1))
}
result_value_11<- R_t_MLE(t,theta1,mu1)
print(result_value_11)
R_t_MLE=result_value_11
R_t_MLE

#####We take the value of theta1 is 1.133, since
theta1 is always positive
L11=-(2*n1/theta1^2)+((n1*(1+mu1)^2)/(1 + theta1 *
(1 + mu1))^2)
L11
L22=((n1 * theta1^2) / (1 + theta1 * (1 + mu1))^2)
L22
##The value of L12=L21
L21=-(n1/(1 + theta1 * (1 + mu1))^2)+n1
```

```
L21
L12=-(n1/(1 + theta1 * (1 + mu1))^2)+n1
L12
L111=(4*n1/theta1^n1)-((2*n1*(1+mu1)^3)/(1+theta1*(1+mu1))^3)
L111
L222 <- -((2 * n1 * theta1^3) / (1 + theta1 * (1 + mu1))^3)
L222

L112 <- ((2 * n1 * (1 + mu1) * (1 + theta1 * (1 + mu1))^2) -
(2 * n1 * theta1 * (1 + mu1)^2
* (1 + theta1 * (1 + mu1)))) / ((1 + theta1 * (1 + mu1))^4)
L112

# Create the matrix
matrix<- matrix(c(-L11, -L12, -L21, -L22), nrow = 2, byrow
= TRUE)

# Print the matrix
print(matrix)

# Find the inverse of the matrix
inverse_matrix <- solve(matrix)

# Print the inverse matrix
print(inverse_matrix)
T11=inverse_matrix[1,1]
T12=inverse_matrix[1,2]
T21=inverse_matrix[2,1]
T22=inverse_matrix[2,2]

#########change

###T11=0.0001797935
##T12=-0.001927182
##T21=-0.001927182
##T22=-0.002337156
suppressWarnings({a1 <- a2 <- b1 <- b2 <- 2})
eta1=((a1-1)/theta1)-b1
eta2=((a2-1)/mu1)-b2


##The squared error loss function of theta is
theta_SE=theta1+T11*eta1+T21*eta2
```

```
+0.5*(L111*T11^2+L222*T22*T12+3*L112*T12*T11)
theta_SE
mu_SE <- mu1 + T22 * eta2 + T12 * eta1
+ 0.5 * (L111 * T11 * T12 + L222 * T22^2 + L112 *
(T11 * T22 + 2 * T12^2))
mu_SE

#########
##Reliability Function (The ML and Bayes estimators
of R(t)are evaluated at t=1)
t=1
R <- function(t, theta1, mu1) {
  result <- ((1 + theta1 * (1 + t))
  / (1 + theta1 * (1 + mu1))) * exp(-theta1 * (t - mu1))
}
result_value1 <- R(t,theta1,mu1)
print(result_value1)
R=result_value1
R

#############
R1 <- function(t, theta1, mu1) {
  numerator <- (1 + theta1 * (1 + t))
  * (theta1^2 * (1 + mu1) * exp(mu1 * theta1))
  denominator <- exp(t * theta1) * (1 + theta1 * (1 + mu1))^2

  result2 <- numerator / denominator

  return(result2)
}
result2 <- R1(1,theta1, mu1)
print(result2)
R1=result2
R1
#############################################
R11 <- function(t, theta1, mu1) {
numerator1 <- (1 + theta1 * (1 + t))
denominator1 <- exp(t * theta1)
numerator2 <- (1 + theta1 * (1 + mu1))^2
* (theta1^2 * exp(theta1 * mu1) +
exp(theta1 * mu1) * theta1^3 * (1 + mu1)) -
2 * theta1^3 * (1 + mu1) * exp(theta1*mu1) *
(1 + theta1 * (1 + mu1))
```

```
denominator2 <- (1 + theta1* (1 + mu1))^4
result3 <- (numerator1 / denominator1)
*(numerator2/ denominator2)
return(result3)
}
result3 <-R11(t, theta1 ,mu1)
print(result3)
R11=result3
R11

##############################################################
R12<- function(t, theta1, mu1) {
numerator1 <- (1 + theta1* (1 + t))
denominator1 <- exp(t * theta1)
numerator2 <- ((1 + theta1* (1 + mu1))^2)
*(2 * theta1* exp(theta1* mu1) * (1 + mu1)
+ theta1^2 * exp(theta1* mu1) * mu1* (1 + mu1)) -
theta^2 * (1 + mu1) * exp(theta1* mu1) * 2 *
((1 + theta1* (1 + mu1)) * (1 + mu1))
denominator2 <- (1 + theta1* (1 + mu1))^4
numerator3 <- (theta1^2 * exp(theta1* mu1) * (1 + mu1))
denominator3 <- (1 + theta1* (1 + mu1))^2
numerator4 <- ((1 + t) * exp(t * theta1) +
(1 + theta1 * (1 + t))* t * exp(t * theta1))
denominator4 <- (exp(t * theta1))^2
result4 <- (
(numerator1 / denominator1) *

(numerator2 /denominator2)) +
((numerator3 / denominator3) * (numerator4 / denominator4))
return(result4)
}
result4 <- R12(t,theta1, mu1)
print(result4)
R12=result4
R12
##########
R2<- function(t, theta1, mu1) {
  numerator <- theta1 * (mu1^2 + 2 * mu1 - t^2 - 2 * t)
  + theta1^2 * (mu1 + mu1^2 + mu1^2 * t
  - t - t^2 - mu1 * t^2) * exp(-theta1 * (t - mu1))
  denominator <- (1 + theta1 * (1 + mu1))^2
```

```
  result5 <- numerator / denominator

  return(result5)
}
result5 <-R2(1,theta1,mu1)
print(result5)
R2=result5
R2
###################
R22<- function(t, theta1, mu1) {
term1_numerator <- theta1 * (mu1^2 + 2 * mu1 - t^2 - 2 * t)

+ theta1^2 * (mu1 + mu1^2 + mu1^2 * t - t - t^2 - mu1 *

t^2) * exp(-theta1 * (t - mu1)) *
(mu1 - t) * (1 + theta1 *(1 + mu1))^2
term1_denominator <- (1 + theta1 * (1 + mu1))^4

term2_numerator <- (1 + theta1 * (1 + mu1))^2 *
exp(-theta1 * (t - mu1)) * theta1 *
(-2 * t - 2 * t^2 - 2 * t^2 * mu1 +
2 * mu1 + 2 * mu1^2 +
2 * mu1^2 * t) + (-2 * t - t^2 + 2 * mu1 + mu1^2)
term2_denominator <- (1 + theta1 * (1 + mu1))^4
term3_numerator <- 2 * (theta1 * (mu1^2 + 2 * mu1 - t^2 -

2* t) +
theta1^2 * (mu1 + mu1^2 + mu1^2 * t - t - t^2 - mu1* t^2))

*exp(-theta1 * (t - mu1)) * (1 + mu1) * (1 + theta1 *
(1 + mu1))
  term3_denominator <- (1 + theta1 * (1 + mu1))^4

result6 <- (term1_numerator / term1_denominator) +

(term2_numerator / term2_denominator) +
(term3_numerator / term3_denominator)

  return(result6)
}
result6 <- R22(1,theta1,mu1)
print(result6)
R22=result6
```

```
R22
######################
v1=R1*T11+R2*T12
v1
v2=R1*T12+R2*T22
v2
####Squared errorloss function
R_SE=R+0.5*(R11*T11+R22*T22+2*R12*T12)+eta1*v1+eta2*v2
+0.5*(L111*T11*v1+L222*T22*v2+L112*(2*T12*v1+T11*v2))
R_SE
#### Llnex loss function#
nu=-0.5####change####################
h1_LE= -nu*exp(-nu*theta1)
h11_LE=(nu)^2*exp(-nu*theta1)

T_LE <- function(nu,theta1, h11_LE, T11, h1_LE,eta1,eta2,
L111, L222, L112, T22,T12){
result12 <- -1/nu * log(exp(-nu*theta1) +
0.5 * h11_LE * T11 + h1_LE * (T11 *eta1 + T21 * eta2) +
0.5 * h1_LE * (L111 * T11^2 + L222 * T22 * T12 +
3 * L112 * T12 * T11))
  return(result12)
}
result12<-T_LE(nu,theta1, h11_LE, T11, h1_LE,eta1,eta2,

L111, L222, L112, T22,T12)
print(result12)
theta_LE=result12
theta_LE
#####################################
h2_LE= -nu*exp(-nu*mu1)
h22_LE=(nu)^2*exp(-nu*mu1)

m_LE <- function(nu,mu1, h22_LE, T11, h2_LE,eta1,eta2,
L111, L222,
L112, T22,T12){
result22 <- -1/nu * log(exp(-nu*mu1) +
0.5 * h22_LE * T22 +
h2_LE * (T22 *eta2 + T12* eta1) + 0.5 * h2_LE *
(L111 * T11*T12 + L222 * T22^2  + L112*(T11*T22+2*T12^2)))
return(result22)
}
result22<-m_LE(nu,mu1, h22_LE, T11, h2_LE,eta1,eta2,
```

```
L111, L222,
L112, T22,T12)
print(result22)
mu_LE=result22
mu_LE
##################################################
h_1_LE=-nu*exp(-nu*R)*R1
h_2_LE=-nu*exp(-nu*R)*R2
h_11_LE=nu*exp(-nu*R)*(nu*R1^2-R11)
h_22_LE=nu*exp(-nu*R)*(nu*R2^2-R22)
h_12_LE=nu*exp(-nu*R)*(nu*R1*R2-R12)

R_LE <- function(nu,R,h_11_LE,T11,
h_22_LE,h_12_LE,eta1,eta2,v1,v2,
L111, L222, L112, T22,T12){
result33<- -1/nu * log(exp(-nu*R) +
0.5 * (h_11_LE *T11 + h_22_LE*T22+2 * h_12_LE *T12) +
v1 * eta1 + v2 * eta2 + 0.5 * (L111 * T11 * v1 + L222 *T22
* v2 + L112 * (2 *T12 * v1 +T11 * v2)))
return(result33)
}
result33<-R_LE(nu,R,h_11_LE, T11,h_22_LE,h_12_LE,
eta1,eta2, v1,v2,L111, L222, L112, T22,T12)
print(result33)
R_LE=result33
R_LE
####################
###true survival function
R_t <- function(t, theta, mu) {
result <- ((1 + theta * (1 + t)) / (1 + theta * (1 +mu)))*

exp(-theta* (t - mu))
}
result_value_11<- R_t(t,theta,mu)
print(result_value_11)
R_t=result_value_11
R_t
###############
##General entropy loss function
w=-0.5###################change
h1_GE= -w*theta1^(-w-1)
h11_GE=w*(w+1)*theta1^(-w-2)
```

```
T_GE <- function(w,theta1, h11_GE, T11, h1_GE,eta1,eta2,
L111, L222, L112, T22,T12){
result_12 <- (theta1^-w+ 0.5 * h11_GE * T11 + h1_GE *
(T11 *eta1 + T21 * eta2) +
0.5 * h1_GE * (L111 * T11^2 + L222 * T22 * T12 +
3 * L112 * T12 * T11))^(-1/w)
return(result_12)
}
result_12<-T_GE(w,theta1, h11_GE, T11, h1_GE,eta1,eta2, L111,
L222, L112, T22,T12)
print(result_12)
theta_GE=result_12
theta_GE
################
h2_GE=-w*mu1^(-w-1)
h22_GE=w*(w+1)*mu1^(-w-2)

m_GE <- function(w,mu1, h22_GE, T11, h2_GE,eta1,eta2, L111,
L222,L112, T22,T12){
result_22 <-(mu1^-w + 0.5 * h22_GE * T22 + h2_GE * (T22 *eta2 +
T12* eta1) + 0.5 * h2_GE * (L111 * T11*T12 + L222 * T22^2
+ L112*(T11*T22+2*T12^2)))^(-1/w)
return(result_22)
}
result_22<-m_GE(w,mu1, h22_GE, T11, h2_GE,eta1,eta2,
L111, L222, L112, T22,T12)
print(result_22)
mu_GE=result_22
mu_GE
##################################
h_1_GE=-w*R^(-w-1)*R1
h_2_GE=-w*R^(-w-1)*R2
h_11_GE=w*(w+1)*R^(-w-2)*R1^2-w*R^(-w-1)*R11
h_22_GE=w*(w+1)*R^(-w-2)*R2^2-w*R^(-w-1)*R22
h_12_GE=w*(w+1)*R^(-w-2)*R1*R2-w*R^(-w-1)*R12

R_GE <- function(w,R,h_11_GE, T11,h_22_LE,h_12_LE,eta1,eta2,
v1,v2,L111, L222, L112, T22,T12){
result_333<- (R^-w+ 0.5 * (h_11_GE *T11 + h_22_GE*T22+
2 * h_12_GE *T12) +
v1 * eta1 + v2 * eta2 + 0.5 * (L111 * T11 * v1 +
L222 *T22 * v2 + L112 *
(2 *T12 * v1 +T11 * v2)))^(-1/w)
```

```
  return(result_333)
}
result_333<-R_GE(w,R,h_11_LE, T11,h_22_LE,h_12_LE,eta1 ,eta2 ,
v1,v2,L111, L222, L112, T22,T12)
print(result_333)
R_GE=result_333
R_GE
####For MLE
theta1
mu1
R_t_MLE
#####For squared error loss function
theta_SE
mu_SE
R_SE
############For Linex loss function
nu
theta_LE
mu_LE
R_LE
#####FOR General entropy Loss function
w
theta_GE
mu_GE
R_GE
```

References

Alduais, F. (2021). Comparison of classical and bayesian estimators to estimate the parameters in weibull distribution under weighted general entropy loss function. *Int. J. Adv. Appl. Sci*, 8:57–62.

Maiti, S. I., Rana, S., and Chakraborty, A. (2021). Estimation in shifted lindley distribution. *Statistics and Applications*, 20:247–263.

Pak, A., Ghitany, M., and Mahmoudi, M. R. (2019). Bayesian inference on power lindley distribution based on different loss functions.