Journal of Modern Applied Statistical Methods

Volume 21 | Issue 2 Article 2

Bayesian Inference and Simulation for Logarithm Transformed (LT) Method Using Stan

Mohammed H AbuJarad
Department of Statistics and Operations Research, Aligarh Muslim University, Aligarh

Eman AbuJarad
Department of Statistics and Operations Research, Aligarh Muslim University, Aligarh

Athar Ali Khan
Department of Statistics and Operations Research, Aligarh Muslim University, Aligarh

Recommended Citation

Mohammed H AbuJarad, Eman AbuJarad, Athar Ali Khan (2022). Bayesian Inference and Simulation for Logarithm Transformed (LT) Method Using Stan. Journal of Modern Applied Statistical Methods, 21(2), https://doi.org/10.56801/Jmasm.V21.i2.2

Journal of Modern Applied Statistical Methods
2022, Vol. 21, No. 2,
Doi: 10.56801/Jmasm.V21.i2.2

ISSN 1538 – 9472

Bayesian Inference and Simulation for Logarithm Transformed (LT) Method Using Stan

Mohammed H AbuJarad
Department of Statistics and Operations Research, AMU, Aligarh

Eman AbuJarad
Department of Statistics and Operations Research, AMU, Aligarh

Athar Ali Khan
Department of Statistics and Operations Research, AMU, Aligarh

In this article, the researchers set up one and two parameters life model called the Logarithm Transformed Exponential Distribution (LTE) and Logarithm Transformed Weibull Distribution (LTW). The researchers give extensive consequence of the survival function and hazard rate function. To fit this model as survival model and hazard rate function the researchers adopted to use Bayesian approach. A real survival data set is used to illustrate this work. Application is done by R and Stan, then a comparison of the two models is conducted by using LOO package to find the best model and simulation. R and Stan codes have been given to actualize censoring mechanism via optimization and also simulation tools.

Keywords: Posterior, Simulation, LTE, LTW, loo, RStan, Bayesian Inference, R.

1. Introduction

S. K. Maurya et al. [12] developed a method for proposing new lifetime distribution which is parsi- monious in parameter and also adds more flexibility in baseline distribution. Here one parameter expo- nential distribution has been taken here as baseline distribution Logarithm Transformed Exponential Distribution(LTE) and Logarithm Transformed Weibull Distribution (LTW). This paper, aim is to fit the Logarithm Transformed Exponential Distribution(LTE) and Logarithm Transformed Weibull Distribution (LTW) using a Bayesian approach and this distribution has an important role in lifetime modelling. Statistical methods for lifetimes data analysis have continued to flourish in the last few decades. Applications of the methods have been seen widened from their historical use in can- cer and reliability research to business, criminology, epidemiology and social sciences. Survival analysis measures the time to certain event, such as failure, death, response, relapse, the development of a given disease, parole or divorce. In many practical situations it has been seen that the survival models are very effectively analyzed in Bayesian paradigm. consequently therefore, with the end goal of Bayesian investigation of this model, two vital techniques are utilized, one of the simulation methods, and the other are strategies for estimating pointwise out-of-sample prediction accuracy from a fitted Bayesian model utilizing the log-probability assessed at the posterior simulations of the parameter values; all are implemented using rstan and loo packages of R. Stan is a new Bayesian statistical software program that implements the powerful and efficient Hamiltonian Monte Carlo (HMC) algorithm. The package rstan [10], is a C + + library for Bayesian modeling and inference to obtain posterior simulation. The package LOO [20], [21] and [25] are employed for estimating pointwise out-of-sample prediction accuracy. A real survival data set is used to illustrate in R. Thus, Bayesian analysis of Logarithm transformed Exponential Distribution(LTE) and Logarithm transformed Weibull Distribution (LTW) have been made with the following objectives:

To define a Bayesian model, that is, specification of likelihood and prior distribution.

To write down the R code for approximating posterior densities with Stan.

To illustrate numeric as well as graphic summaries of the posterior densities.

2. Logarithm transformed (LT) method

S. K. Maurya et al [12] proposed a transformation and called it Logarithm transformed (LT), to obtain a new distribution. If f (x) and F (x) be the probability density function (pdf) and (cdf) of some baseline distribution, then the (pdf) g(x) of new distribution is proposed by:

$$f(x) = 1 - \frac{g(x)}{(2 - G(x))log2} \tag{1}$$

The (cdf) and hazard rate function corresponding to the (pdf) g(x) are given by:

$$F(x) = 1 - \frac{log(2 - G(x))}{log2} \tag{2}$$

$$h(x) = \frac{g(x)}{(2 - G(x)) * log(2 - G(x))} \tag{3}$$

2.1 Logarithm Transformed Exponential Distribution (LTE)

S. K. Maurya et al [12] utilized the method for adding parameter prompts the Exponential distri- bution. In this segment, one parameter exponential distribution as baseline distribution we infer one parameter, then CDF of new distribution by taking LT method define in equation (1) and say it Log- arithmic Transformed Exponential (LTE) distribution. To construct the probability density function (pdf) and cumulative distribution function (cdf) of Exponential distribution which are given by (4) and (3), individually,

$$g(x) = \frac{1}{\theta}exp(\frac{-x}{\theta}) \tag{4}$$

$$G(x) = \left(1 - exp(\frac{-x}{\theta})\right) \tag{5}$$

By inserting (3) and (4) into (1) and (2), we have the pdf, cdf, survival function and hazard func- tion of the Logarithm Transformed Exponential Distribution(LTE) given by respectively, as in Figure(1).

$$f(x) = \frac{\frac{1}{\lambda}e^{-\frac{x}{\lambda}}}{(1+e^{-\frac{x}{\lambda}})log2} \tag{6}$$

$$F(t) = 1 - \frac{log(1+e^{-\frac{x}{\lambda}})}{log2} \tag{7}$$

$$S(t) = 1 - \left(1 - \frac{log(1+e^{-\frac{x}{\lambda}})}{log2}\right) \tag{8}$$

$$h(t) = \frac{f(t)}{S(t)} = \frac{\frac{1}{\lambda}e^{-\frac{x}{\lambda}}}{(1+e^{-\frac{x}{\lambda}})log(1+e^{-\frac{x}{\lambda}})} \tag{9}$$

2.2 Logarithm Transformed Weibull Distribution (LTW)

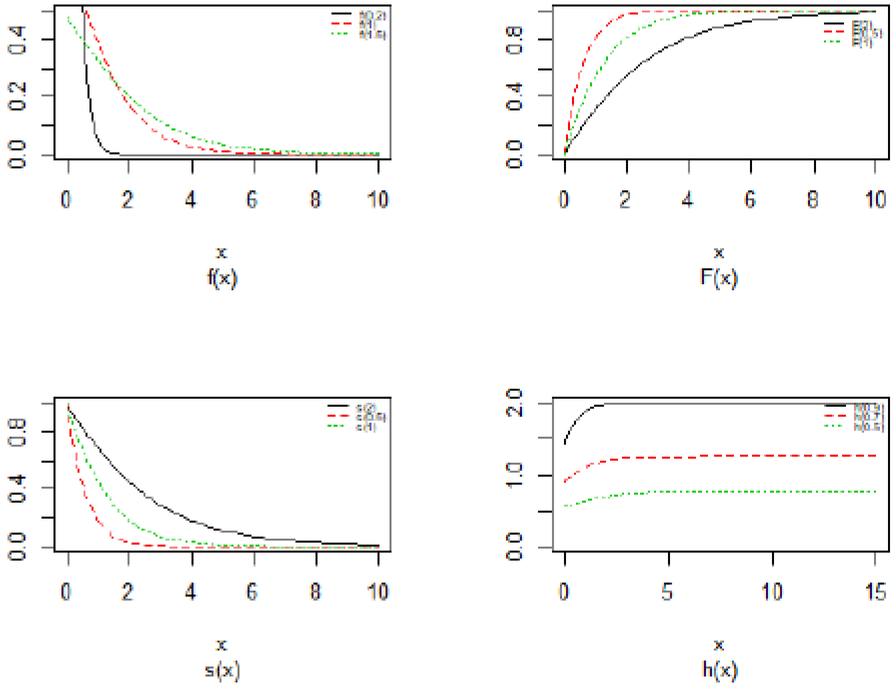
Logarithm Transformed Weibull Distribution (LTW) additional Logarithm Transformed to the two-parameter Weibull distribution. It is clear that the new two-parameter distribution is exceptionally adaptable. At the point when the (pdf), (cdf), survival function and hazard function of LTW appropriation is f (x) $\sim$ w(θ, α), the outcomes are (10), (11), (12) and (13), individually, as in Figure(2)

$$f(x) = \frac{\frac{\theta}{\lambda}(\frac{x}{\lambda})^{\theta-1}e^{-\frac{x}{\lambda}^{\theta}}}{(1+e^{-\frac{x}{\lambda}^{\theta}})log2} \tag{10}$$

$$F(t) = 1 - \frac{log(1+e^{-\frac{x}{\lambda}^{\theta}})}{log2} \tag{11}$$

$$S(t) = 1 - \left(1 - \frac{log(1+e^{-\frac{x}{\lambda}^{\theta}})}{log2}\right) \tag{12}$$

$$h(t) = \frac{f(t)}{S(t)} = \frac{\frac{\theta}{\lambda}(\frac{x}{\lambda})^{\theta-1}e^{-\frac{x}{\lambda}^{\theta}}}{(1+e^{-\frac{x}{\lambda}^{\theta}})log(1+e^{-\frac{x}{\lambda}^{\theta}})} \tag{13}$$



f(x) F(x)

s(x) h(x)

Figure 1: Probability density plots, cdf, survival and hazard curves of logarithm transformed exponential distribution (LTE) for different value.

3. Stan

Stan (named after Stanislas Ullam, see [4]) can be defined [17]. Stan is a computer language for Bayesian inference that, among other approximate techniques, implements the NUTS algorithm to remove hand- tuning. More precisely, as a probabilistic programming language for statistical inference written in C++. It uses the No-U-Turn sampler (NUTS) [10] to obtain posterior simulation given user-specified model and data. The Stan language is used to specify a (Bayesian) statistical model with an imperative program calculating the log probability density function. Using stan, statistical modeling becomes easier and faster, particularly for Bayesian estimation problems. Complex models with large numbers of parameters can be estimated easily using stan, and can generally do it faster than alternative like JAGS/BUGS. Stan manipulates two main tools to efficiently to work out and solve Bayesian problems: Hamiltonian Monte Carlo (HMC) and the no-U-turn sampler (NUTS).Markov chain Monte Carlo (MCMC) algorithms allowing drawing random samples from the posterior. Stan implements Hamiltonian Monte Carlo [14] and its extension, the No-U-Turn Sampler (NUTS) [10]. These algorithms converge much more quickly especially for high-dimensional models regardless of whether the priors are conjugate or not. As an illustration of the acceleration brought by HMC.

4. Leave-One-Out Cross-Validation and WAIC for Bayesian

This package LOO implements the methods described in [20], [21] and [25]. Watanabe-Akaike information criterion (WAIC) proposed by [24] and leave-one-out cross-validation (LOO) both allowing to compare different models applied to the same data (lower WAICs and LOOs indicate better model fit) using the log-likelihood evaluated at the posterior simulations of the parameter values. The WAIC can be seen as an improvement of the popular deviance information criterion (DIC), which has been criticized by several authors [15], [22] in part because of problems arising from fact that the DIC is only a point estimate. In rstan, WAIC and LOO are implemented using the LOO package [20], [21].

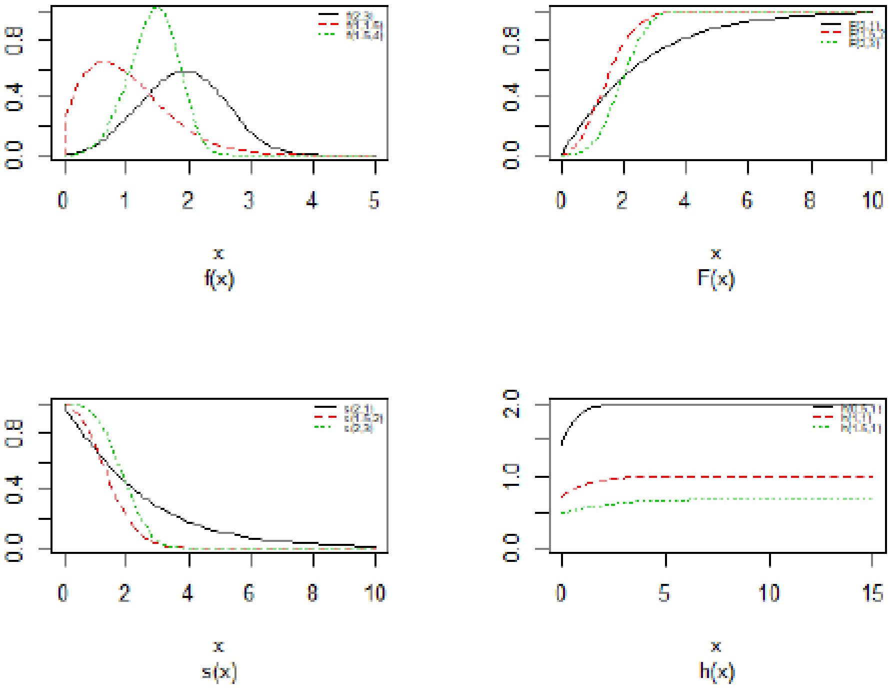


Figure 2: Probability density plots, cdf, survival and hazard curves of Logarithm Transformed Weibull Distribution (LTW) for different values of θ and λ.

---

5. Bayesian Inference

1. p(θ) is the set of prior distributions for parameter set θ; for it uses probability as a means of quantifying uncertainty about θ before taking the data into account.
2. p($x|\theta$) is the likelihood or likelihood function, in which all variables are related in a full probability model

3. p($\theta|x$) is the joint posterior distribution that expresses uncertainty about parameter set θ after taking both the prior and the data into account. Calculating posterior probabilities is the main goal of Bayesian statistics.

When prior information is available about θ, it should be included in the prior distribution of θ. then the posterior distribution of θ from the previous model may be used as the prior distribution of θ for the present model. To determine the propriety of a joint posterior distribution, the marginal likelihood must be finite for all x. Then, the marginal likelihood is:

$$p(x) = \int p(x|\theta)p(\theta)d\theta \tag{14}$$

It is popular, for good reasons, to center and scale all continuous predictors [9]. Although centering and scaling predictors are not discussed here, it should be obvious that the potential range of the posterior distribution of θ for a centered and scaled predictor should be small. A popular WIP for a centered and scaled predictor may be

$$\theta \sim N(0, 10000)$$

[8], better properties for scale parameters are yielded with the non-conjugate, proper, halfCauchy distribution, with a general recommendation of scale=25 for a weakly informative scale parameter

$\delta \sim HC(25)$

use the half-Cauchy distribution with scale parameter $\delta = 25$ as prior distribution for scale parameters.

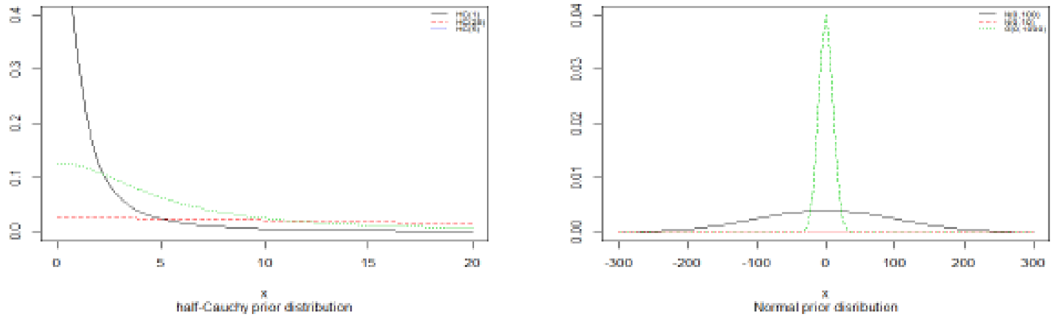


Figure 3

6. Survival Data: Veteran's Administration Lung Cancer Data

In this data, males with advanced inoperable lung cancer were randomized to either a standard or test chemotherapy. Only 9 of the 137 survival times were censored. The data is available in survival package and is presented in [11] and [16]. A portions of the data is analyzed by several other authors ([16]; [5]; [13]; [3]). In this analysis, the 137 subjects who completed the randomized portion of the trial and for whom complete covariate information was available are considered. Six covariates are available which include treatment, age, tumor cell type (adeno, small cell, squamous or large), time between ini- tial diagnosis and enrollment in the trial, Karnofsky performance status, and prior therapy attempted (yes/no).

Treatment: 0 = standard, 1 = test.

Type of tumour: 1 = squamous, 2 = small cell, 3 = adeno, 4 = large cell.

Age in years.

Prior therapy: 0 = no, 1 = yes.

diagtime: Time in months from diagnosis to randomization.

Performance status: Karnofsky performance score (100 = good).

7. Bayesian Analysis of Model

Bayesian analysis is the method to obtain the marginal posterior distribution of the particular param- eters of interest. In principle, the route to achieving this aim is clear; first, we require the joint posterior distribution of all unknown parameters, then, one integrates this distribution over the unknowns param- eters that are not of immediate interest to obtain the desired marginal distribution. Or equivalently, using simulation, we draw samples from the joint posterior distribution, then looks at the parameters of interest and ignores the values of the other unknown parameters. The Bayesian paradigm is based on specifying a probability model for the observed data y, given a vector of unknown parameters λ, leading to the likelihood function L(λ y). Then we assume that λ is random and has a prior distribution denoted by p(λ). Inference concerning λ is then based on the posterior distribution, which is obtained by Bayes theorem. The posterior distribution of λ is given as equation (14).(see, e.g., [2]).

7.1 Bayesian Analysis of Logarithm Transformed Exponential (LTE) Model

As of now, the probability density function (pdf) is indicated through this condition;

$$f(x;\lambda) = \frac{\frac{1}{\lambda}e^{-\frac{x}{\lambda}}}{(1+e^{-\frac{x}{\lambda}})log2}.$$

Additionally, the survival function is set through;

$$S(x;\lambda) = 1 - F(x) = 1 - \left(1 - \frac{log(1+e^{-\frac{x}{\lambda}})}{log2}\right).$$

We have the capacity to condition the likelihood function for right censored (in the meantime similar to our case the data are right censored) as (see, e.g., [2]).

$$
\begin{aligned}
L &= \prod_{i=0}^{n} Pr(x_i, \delta_i) \\
&= \prod_{i=0}^{n} [f(x_i)]^{\delta_i} [S(x_i)]^{1-\delta_i}
\end{aligned}
$$

where δi is an indicator variable which takes value 0 if observation is censored and 1 if observation is uncensored. Subsequently, the likelihood function is indicated by means of

$$
L = \prod_{i=0}^{n} \Bigg[\frac{\frac{1}{\lambda} e^{-\frac{x}{\lambda}}}{(1+e^{-\frac{x}{\lambda}}) log2} \Bigg]^{\delta_i} \Bigg[1 - \Bigg(1 - \frac{log(1+e^{-\frac{x}{\lambda}})}{log2} \Bigg) \Bigg]^{1-\delta_i}. \tag{15}
$$

Subsequently, the posterior distribution of our convictions about the basic inclination is determined in the standard way by applying Bayes' rule [18], the joint posterior density is given by [2]. Here the functions include two parameters.

$$
\begin{aligned}
p(\beta | x, X) &\propto L(t|X, \beta) \times p(\beta) \\
&\propto \prod_{i=0}^{n} \Bigg[\frac{\frac{1}{e^{X\beta}} e^{-\frac{x}{e^{X\beta}}}}{(1+e^{-\frac{x}{e^{X\beta}}}) log2} \Bigg]^{\delta_i} \Bigg[1 - \Bigg(1 - \frac{log(1+e^{-\frac{x}{e^{X\beta}}})}{log2} \Bigg) \Bigg]^{1-\delta_i} \\
&\times \prod_{i=0}^{J} \frac{1}{\sqrt{2\pi \times 10^3}} exp\Bigg(-\frac{1}{2}\frac{\beta_j^2}{10^3} \Bigg).
\end{aligned} \tag{16}
$$

Now the researchers get the Bayesian inference in the Logarithm Transformed Exponential (LTE) Model, as they use the prior distribution for βJs. the researchers discussed the issue associated with specifying prior distributions in section 4. Elementary application of Bayes rule as displayed in (14), is applied to (15), then gives the posterior density for βJs as equation (16). Result for this marginal posterior distribution get high-dimensional integral over all model parameters βJs. The posterior in (16) does not have a closed form and one needs to use numerical integration or Markov chain Monte Carlo (MCMC) methods. These methods can be used to solve the complex numerical integration including censoring mechanism using Stan functions. However, due to the availability of computer software package like rstan, this required model can easily be fitted in Bayesian paradigm using Stan as well as MCMC techniques.(see, e.g., [2]).

7.2 Bayesian Analysis of Logarithm Transformed Weibull (LTW) Model

Now, the probability density function (pdf) is given by

$$
f(x; \lambda, \theta) = \frac{\frac{\theta}{\lambda}(\frac{x}{\lambda})^{\theta-1} e^{-\frac{x}{\lambda}^{\theta}}}{(1+e^{-\frac{x}{\lambda}^{\theta}}) log2}
$$

Also, the survival function is given by

$$S(x;\lambda,\theta) = 1 - F(x) = 1 - \left(1 - \frac{log(1+e^{-\frac{x}{\lambda}^{\theta}})}{log2}\right)$$

In the presence of censoring, the resulting log-likelihood function is modified to account for the possibility of partially observed data (in correspondence with censoring) We can write the likelihood function for right censored (as is our case the data are right censored) as

$$\begin{aligned} L &= \prod_{i=0}^{n} Pr(x_i, \delta_i) \\ &= \prod_{i=0}^{n} [f(x_i)]^{\delta_i} [S(x_i)]^{1-\delta_i}, \end{aligned}$$

where δi is an indicator variable which takes value 0 if observation is censored and 1 if observation is uncensored. Thus, the likelihood function is given by

$$L = \prod_{i=0}^{n} \left[\frac{\frac{\theta}{\lambda}(\frac{x}{\lambda})^{\theta-1} e^{-\frac{x}{\lambda}^{\theta}}}{(1+e^{-\frac{x}{\lambda}^{\theta}})log2} \right]^{\delta_i} \left[1 - \left(1 - \frac{log(1+e^{-\frac{x}{\lambda}^{\theta}})}{log2} \right) \right]^{1-\delta_i}. \tag{17}$$

Thus, the joint posterior density is given by

$$\begin{aligned} p(\beta,\theta|x,X) &\propto L(x|X,\beta,\theta) \times p(\beta) \times p(\theta) \\ &\propto \prod_{i=0}^{n} \left[\frac{\frac{\theta}{e^{X\beta}}(\frac{x}{e^{X\beta}})^{\theta-1} e^{-\frac{x}{e^{X\beta}}^{\theta}}}{(1+e^{-\frac{x}{e^{X\beta}}^{\theta}})log2} \right]^{\delta_i} \left[1 - \left(1 - \frac{log(1+e^{-\frac{x}{e^{X\beta}}^{\theta}})}{log2} \right) \right]^{1-\delta_i} \\ &\times \prod_{i=0}^{J} \frac{1}{\sqrt{2\pi \times 10^3}} exp\left(-\frac{1}{2}\frac{\beta_j^2}{10^3} \right) \times \frac{2 \times 25}{\pi(\theta^2 + 25^2)}. \end{aligned} \tag{18}$$

To carry out Bayesian inference in the Logarithm Transformed Weibull (LTW) model, we must specify a prior distribution for θ and βJs. We discussed the issue associated with specifying prior distributions was discussed in section 5, but for simplicity at this point, we assume that the prior distribution for θ and βJs is half-Cauchy on the interval [0, 5] and for β is Normal with [0, 5]. Elementary application of Bayes rule as displayed in (5.14), is applied to (7.17), then given the posterior density for θ and βJs as equation (7.18). The result for this marginal posterior distribution gets high-dimensional integral over all model parameters θ and βJs. To resolve this integral, the researchers use the approximated using Markov chain Monte Carlo methods. However, due to the availability of computer software package like rstan, this required model can easily fit in Bayesian paradigm using Stan as well as MCMC techniques.

8. Implementation Using Stan

The main function in the rstan package is Stan, which calls the Stan software program to estimate a specified statistical model. In order to use the Stan function, a Stan program has to be specified either as a separate Stan file ending with ’stan’ as the suffix or as an object in the R environment. The rstan package allows one to conveniently fit Stan models from R [19] and access the output, including posterior inferences and intermediate quantities such as evaluations of the log posterior density and its gradients.

Import, clean, and prepare the data in R Before writing any Stan code; and thus the researchers typically first have to import and clean the data.

Define a statistical model in Stan. This is the meat of the model. We define a statistical model using the Stan language, either in a separate file or as a string in R.

4. Extract draws from posterior and perform inference on the parameters using R After fitting the model in Stan, the researchers again use R to extract draws from the posterior and work with these draws to perform inference on the parameters.

8.1 Model Specification

Presently we will look at the posterior estimates of the parameters when the Analysis of Logarithm Transformed Exponential Distribution (LTE) and Logarithm Transformed Weibull Distribution (LTW) model’s are fitted to the previously mentioned data. Accordingly the importance of the likelihood turns into the highest need for the Bayesian fitting. Indicate statistical models utilizing the Stan modeling language, which is detailed in the manual of Stan [17]. Here, we have likelihood as:(see, e.g., [2]).

$$\begin{aligned} L(\theta|x) &= \prod_{i=1}^{n} f(x_i)^{\delta_i} S(x_i)^{1-\delta_i} \\ &= \prod_{i=1}^{n} \left(\frac{f(x_i)}{S(x_i)}^{\delta_i} S(x_i) \right) \\ &= \prod_{i=1}^{n} h(x_i)^{\delta_i} S(x_i), \end{aligned}$$

along these lines, our log-likelihood progresses toward getting to be

$$logL = \sum_{i=i}^{n} \left(log\left[h(x_i) \right]^{\delta_i} + log(S(x_i)) \right).$$

8.1.1 Logarithm Transformed Exponential (LTE) Model

The first model is Logarithm transformed Exponential (LTE) Model:

$x \sim \text{LTE}(\lambda)$,

where $\lambda = \exp(X\beta)$ a linear combination of explanatory variables, log is the natural log for the time to failure event. The Bayesian system requires the determination and specification of prior distributions for the parameters. Here, we stick to subjectivity and thus introduce weakly informative priors for the parameters. Priors for the βj are taken to be normal as follows:

$\beta_j \sim N(0,5); \quad j = 1,2,3,...J$

```
library(rstan)
model_code1=" functions{
//defined survival
vector log_s(vector t, vector lambda){ vector[num_elements(t)] log_s;
for(i in 1:num_elements(t)){
log_s[i]=log(1-(1-((log(1+exp(-(t[i]/lambda[i]))))/log(2))));
}
return log_s;
}
//define log_ft
vector log_ft(vector t, vector lambda){ vector[num_elements(t)] log_ft;
for(i in 1:num_elements(t)){ log_ft[i]=log(((1/lambda[i])*
exp(-(t[i]/lambda[i])))/((1+exp(-(t[i]/lambda[i])))*log(2)));
}
return log_ft;
}
//define log hazard
vector log_h(vector t, vector lambda){ vector[num_elements(t)] log_h;
vector[num_elements(t)] logft; vector[num_elements(t)] logs;
logft=log_ft(t,lambda); logs=log_s(t,lambda);
log_h=logft-logs; return log_h;
}
//define the sampling distribution
real surv_LTE_lpdf(vector t, vector d, vector lambda){ vector[num_elements(t)]
log_lik;
real prob;
log_lik=d .* log_h(t,lambda)+log_s(t,lambda); prob=sum(log_lik);
return prob;
}
}
```

In this manner, we acquire the survival and hazard of the Logarithm Transformed Exponential (LTE) Model.

8.1.2 Logarithm Transformed Weibull (LTW) Model

The second model is Logarithm Transformed Weibull (LTW) Model:

$x \sim LTW\,(\lambda,\ \theta),$

where $\lambda = \exp(X\beta)$. The Bayesian framework requires the specification of prior distributions for the parameters. Here, one sticks to subjectivity and thus introduces weakly informative priors for the parameters. Priors for the β and θ are taken to be normal and half-Cauchy as follows:

$\beta_j \sim \mathrm{N}\,(0, 5);\quad j = 1, 2, 3, \ldots J$

$\theta \sim \mathrm{HC}(0, 25).$

To fit this model in Stan, one first writes the Stan model code and saves it in a separated text-file with name ”model code2”.:

```
library(rstan)
model_code2="
functions{
//defined survival
vector log_s(vector t, real k, vector lambda){
vector[num_elements(t)] log_s;
for(i in 1:num_elements(t)){
log_s[i]=log(1-(1-((log(1+exp(-(t[i]/lambda[i])^k)))/log(2))));
}
return log_s;
}
//define log_ft
vector log_ft(vector t, real k, vector lambda){
vector[num_elements(t)] log_ft;
for(i in 1:num_elements(t)){
log_ft[i]=log(((k/lambda[i])*(t[i]/lambda[i])^(k-1)*
exp(-(t[i]/lambda[i])^k))/((1+exp(-(t[i]/lambda[i])^k))*log(2)));
}
return log_ft;
}
```

```
//define log hazard
vector log_h(vector t, real k, vector lambda){
vector[num_elements(t)] log_h;
vector[num_elements(t)] logft;
vector[num_elements(t)] logs; logft=log_ft(t,k,lambda);
logs=log_s(t,k,lambda);
log_h=logft-logs; return log_h;
}
//define the sampling distribution
real surv_LTW_lpdf(vector t, vector d, real k, vector lambda){
vector[num_elements(t)] log_lik;
real prob;
log_lik=d .* log_h(t,k,lambda)+log_s(t,k,lambda); prob=sum(log_lik);
return prob;
}
}
```

Therefore, we obtain the survival and hazard of the Logarithm Transformed Weibull (LTW) Model.

8.2 Build the Stan

The first necessary component of a Stan model is the data block, in which a researcher specifies the relevant data information and the data itself, in which one includes the number of the observations, observed times, censoring indicator (1=observed, 0=censored), number of covariates, and build the matrix of covariates (with N rows and M columns). Following the data block is the parameters block in which model parameters are specified. The model block, where the priors and the model are specified, is the most essential component of a Stan program. The generated quantities block is an optional component of the Stan code, and it is usually used when there is a need to compute new variables and obtain their corresponding posterior distributions. We save this work in a file to use it in rstan package.(see, e.g., [2] and [1]).

8.2.1 Logarithm Transformed Exponential (LTE) Model

```
//data block data {
int N;   // number of observations
vector<lower=0>[N] y;          // observed times
```

```
vector<lower=0,upper=1>[N] censor;//censoring indicator (1=observed, 0=censored)
int M;  // number of covariates
matrix[N, M] x;          // matrix of covariates (with N rows and M columns)
}
parameters {
vector[M] beta; // Coefficients in the linear predictor (including intercept)
}
transformed parameters { vector[N] linpred; vector[N] lambda; linpred = x * beta;
for (i in 1:N) {
lambda[i] = exp(linpred[i]);
}
}
model {
beta ~ normal(0, 1000);
~ surv_LTE (censor, lambda);
}
generated quantities{ vector [N] log_lik; for(n in 1:N)
log_lik[n]=log(((1/exp(x[n,]*beta))*
exp(-(y[n]/exp(x[n,]*beta))))/((1+exp(-(y[n]/exp(x[n,]*beta))))*log(2)));
}
"
```

8.2.2 Logarithm Transformed Weibull (LTW) Model

```
//data block
data {
int N;   // number of observations
vector<lower=0>[N] y;          // observed times
vector<lower=0,upper=1>[N] censor;//censoring indicator (1=observed, 0=censored)
int M;  // number of covariates
matrix[N, M] x;          // matrix of covariates (with N rows and M columns)
}
parameters {
vector[M] beta; // Coefficients in the linear predictor (including intercept)
real<lower=0> k;       // shape parameter
```

```
}
transformed parameters { vector[N] linpred; vector[N] lambda; linpred = x * beta;
for (i in 1:N) {
lambda[i] = exp(linpred[i]);
}
}
model {
k          ~ cauchy(0,5); beta ~ normal(0, 1000);
y ~ surv_LTW (censor,k, lambda);
}
generated quantities{ vector [N] log_lik; for(n in 1:N)
log_lik[n]=log(((k/exp(x[n,]*beta))*(y[n]/exp(x[n,]*beta))^(k-1)*
exp(-(y[n]/exp(x[n,]*beta))^k))/((1+exp(-(y[n]/exp(x[n,]*beta))^k))*log(2)));
}
"
```

8.3 Creation of Data for Stan

The function Stan requires data that are specified in a list. For illustrative purpose, a real survival data set called veteran that is provided with the survival package is used. The survival data, called, veteran contains six regressor variable i.e celltype, karno, diagtime, age, prior and trt, and its vector have been defined by objects names x1, x2, x3, x4, x5 and x6, respectively, using an extraction operator $.

```
library(survival) data(veteran)
y<-veteran$time
x1<-veteran$karno
x2<-veteran$celltype
x3<-veteran$diagtime
x4<-veteran$age
x5<-veteran$prior
x6<-veteran$trt
censor<-veteran$status
N<-137
x<-cbind(1,x1,x2,x3,x4,x5,x6)
M<-7
```

```
N = nrow(x) M = ncol(x) event=censor
```

```
dat <- list( y=y, x=x, event=event, N=N, M=M)
```

Total number of observations is 137, censoring is taken into account, where 0 stands for censored and 1 for uncensored values. Finally, all these things are combined in a list of dat.

8.4 Runing the Model Using Stan for Logarithm Transformed Exponential (LTE) Model

Now we run Stan with 2 chains for 5000 iterations and display the results numerically and graphically:

regression coefficient with log(y) as a guess to initialize

```
beta1=solve(crossprod(x),crossprod(x,log(y))) #convert matrix to a vector
```

```
beta1=c(beta1)
```

```
M1<-stan(model_code=model_code1,init=list(list(beta=beta1),list(beta=2*beta1))
```

```
,data=dat,iter=5000,chains=2)
```

8.4.1 Summarizing Output

A summary of the parameter distributions can be obtained by using print(M1), which provides pos- terior estimates for each of the parameters in the model. Before any inferences can be made, however, it is critically important to determine whether the sampling process has converged to the posterior distri- bution. The column named n eff lists the effective number of simulation draws, which can be viewed as the effective sample for a posterior distribution on which inferences are based. The last column Rhat lists the Gelman and Rubin’s convergence diagnostic [6], a popular statistic that computes the potential scale reduction factor (PSRF). A PSRF value close to 1 usually indicates model convergence, [7], both of which are outputs in the summary statistics with print(M1). The function rstan approximates the posterior density of the fitted model and posterior summaries can be seen in the following tables. Table (1), which contains summaries for all chains merged and individual chains, respectively. Included in the summaries are (quantiles),(means), standard deviations (sd), effective sample sizes (n eff), and split (Rhats) (the potential scale reduction derived from all chains after splitting each chain in half and treating the halves as chains). For the summary of all chains merged, Monte Carlo standard errors (se mean) are also reported.

The function rstan for this regression model, simulates the data from the posterior density with Hamiltonian Monte Carlo algorithm and summaries of results are reported in the following table:

Table 1. Summary of the simulated results using rstan function with Mean stands for posterior mean, se mean, sd for posterior standard deviation, LB, Median, UB are 2.5%, 50%, 97.5% quantiles, n eff for

	mean	se mean	sd	2.5%	25%	50%	75%	97.5%	n eff	Rhat
beta[0]	3.15	0.01	0.67	1.88	2.70	3.14	3.61	4.50	2383	1
beta[1]	0.03	0.00	0.00	0.02	0.03	0.03	0.04	0.04	3436	1
beta[2]	-0.14	0.00	0.07	-0.29	-0.19	-0.14	-0.10	-0.01	3764	1
beta[3]	0.00	0.00	0.01	-0.02	-0.01	0.00	0.00	0.02	3274	1
beta[4]	0.00	0.00	0.01	-0.02	-0.01	0.00	0.01	0.02	2702	1
beta[5]	0.01	0.00	0.02	-0.03	0.00	0.01	0.03	0.05	3497	1
beta[6]	-0.19	0.00	0.17	-0.53	-0.31	-0.20	-0.08	0.15	3630	1

The inference of the posterior density after fitting the (Logarithm Transformed Exponential Model) for (Veteran's Administration Lung Cancer Data) using stan are reposted in Table 1. It may noted that posterior mean of parameters β0, β1, β2, β3, β4, β5 and β6 are 3.15 ± 0.67, 0.03 ± 0.00, −0.14 ± 0.07, 0.00 ± 0.01, 0.00 ± 0.01, 0.01 ± 0.02 and −0.19 ± 0.17 respectively. According to 95% credible intervals, β0, β1 and β2 are found to statistically significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. The effective sample size given an indication of the underlying autocorrelation in the MCMC samples values are close to the total number of iterations. The selection of appropriate regressor variables can also be done by using a caterpillar plot. Caterpillar plots are popular plots in Bayesian inference for summarizing the quantiles of posterior samples. one can see in this (Figure 4).

```
library(coda)

conflict.fit.coda<-mcmc.list(lapply(1:ncol(M2),function(X)
mcmc(as.array(M2)[,X,]))) plot(conflict.fit.coda[,c(1:3)])
```

This (Figure 4) shows the traces of the parameters on the left, each color represent a different chain, we had 2 chains (the default) and you want all chain to converge to similar values (ie no divergence in the values on the right side of the plot). On the right side of the plot are the posterior distributions of the parameters.

8.5 Runing the Model Using Stan for Logarithm Transformed Weibull (LTW) Model

Now we run Stan with 2 chains for 5000 iterations and display the results numerically and graphically:

```
M3<-stan(model_code=model_code2,data=dat,iter=5000,

chains=2, control = list(adapt_delta = 0.99))

print(M3, c("beta","alpha","b"),digits=2)
```

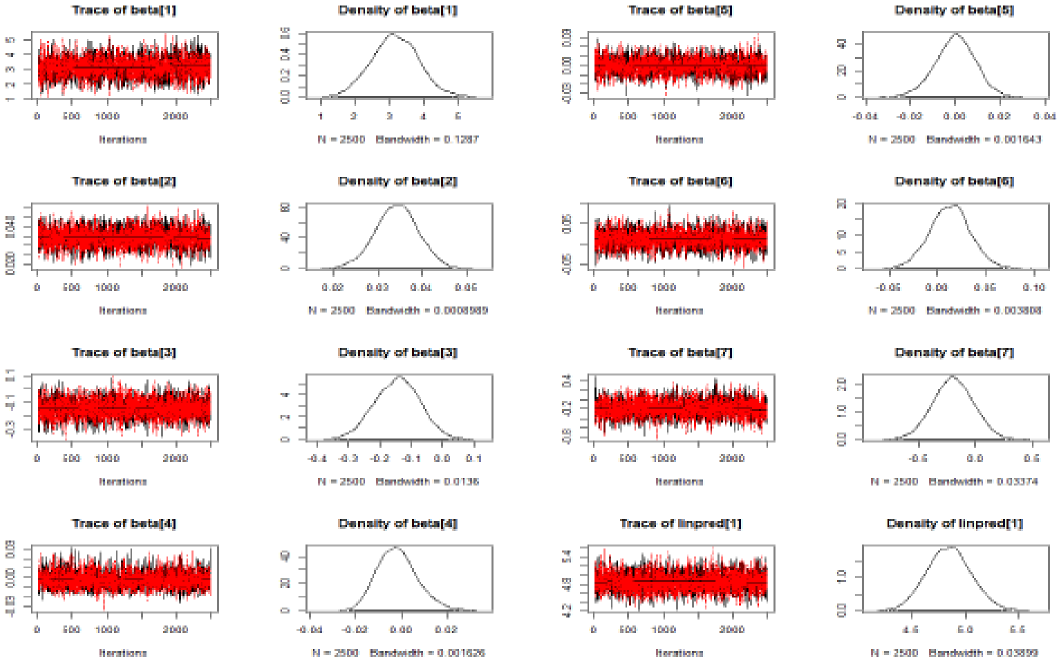


Figure 4: Checking model convergence using rstan,through inspection of the traceplots or the autocorrelation plot

8.5.1 Summarizing Output

The function rstan approximates the posterior density of the fitted model, and posterior summaries can be seen in the following tables. Table (2) contains summaries for for all chains merged and individual chains, respectively. Included in the summaries are (quantiles), (means), standard deviations (sd), effective sample sizes (n eff), and split (Rhats) (the potential scale reduction derived from all chains after splitting each chain in half and treating the halves as chains). For the summary of all chains merged, Monte Carlo standard errors (se mean) are also reported. The inference of the posterior density after fitting the (Logarithm Transformed Weibull Model) for (Veteran's Administration Lung Cancer Data) using stan are reposted in Table (2). It may noted that posterior mean of parameters β0, β1, β2, β3, β4, β5, β6 and k are 3.01±0.73, 0.03±0.01, −0.14±0.08, 0.00±0.01, 0.00±0.01, 0.01±0.02, −0.18±0.19 and 0.90 0.06 respectively. According to 95% credible intervals, β0, β1 and k are found to statistically significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence,. The effective sample size given an indication of the underlying autocorrelation in the MCMC samples values are close to the total number of iterations. The selection of appropriate regressor variables can also be done by using a caterpillar plot. Caterpillar plots are popular plots in Bayesian inference for summarizing the quantiles of posterior samples. one can see in this (Figure 5).

Table 2. Summary of the simulated results using rstan function with Mean stands for posterior mean, se mean, sd for posterior standard deviation, LB, Median, UB are 2.5%, 50%, 97.5% quantiles, n eff for number effective sample size, and Rhat, respectively

	mean	se mean	sd	2.5%	25%	50%	75%	97.5 %	n eff	Rhat
beta[0]	3.01	0.02	0.73	1.59	2.52	3.02	3.51	4.43	2228	1
beta[1]	0.03	0.00	0.01	0.02	0.03	0.03	0.04	0.04	3015	1
beta[2]	-0.14	0.00	0.08	-0.29	-0.19	-0.14	-0.09	0.01	3777	1
beta[3]	0.00	0.00	0.01	-0.02	-0.01	0.00	0.00	0.02	3192	1
beta[4]	0.00	0.00	0.01	-0.02	-0.01	0.00	0.01	0.02	2775	1
beta[5]	0.01	0.00	0.02	-0.03	0.00	0.01	0.03	0.06	3450	1
beta[6]	-0.18	0.00	0.19	-0.56	-0.31	-0.18	-0.06	0.18	4007	1
k	0.90	0.00	0.06	0.78	0.86	0.90	0.95	1.03	3412	1

```
library(coda)
```

```
conflict.fit.coda<-mcmc.list(lapply(1:ncol(M3),function(X)
mcmc(as.array(M3)[,X,]))) plot(conflict.fit.coda[,c(1:3)])
```

```
stan_plot(M3,c("beta","alpha","b"))
```

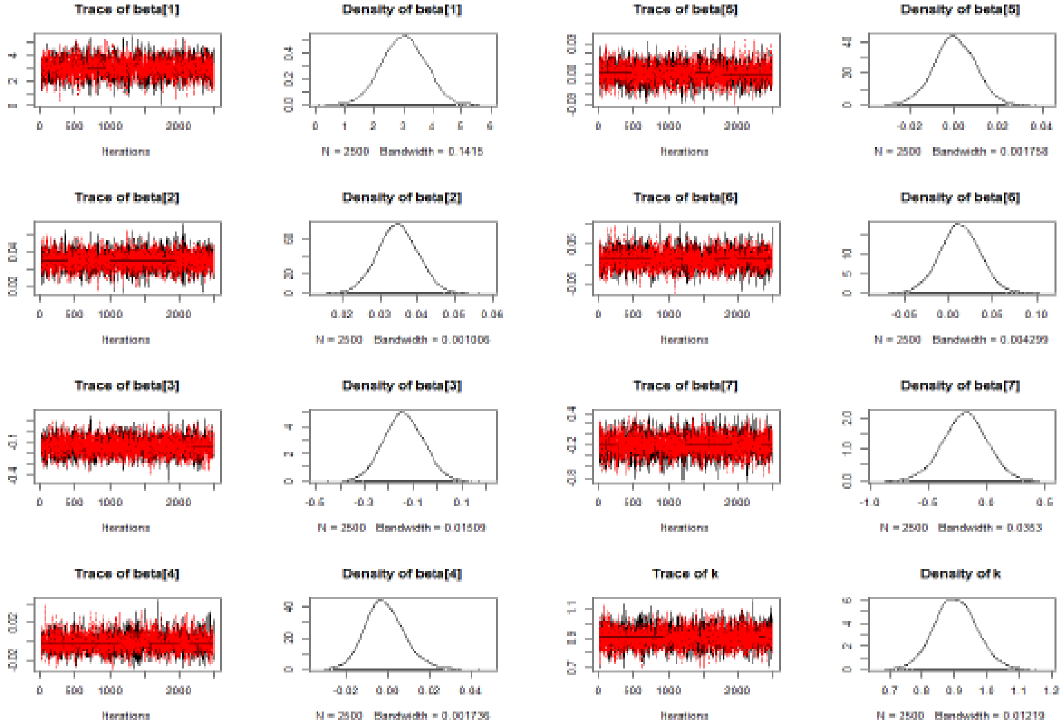


Figure 5: Checking model convergence using rstan,through inspection of the traceplots or the autocorrelation plot

9. Using Leave-One-Out Cross-Validation and WAIC From Stan

We have set up a code to implement LOO, WAIC the widely applicable or Watanabe-Akaike information criterion (WAIC; [24]) can be viewed as an, improvement on the deviance information criterion (DIC) for Bayesian models, in R and Stan so that users will have a quick and convenient way to assess and compare model fits. Implementation is not automatic, though, because of the need to compute the separate factors $p(yi\ \theta)$ in the likelihood, stan works with the joint density and in its usual computations does not know which parts come from the prior and which from the likelihood. Sample code for carrying out this procedure using Stan and the LOO R package [20] and [21]. We have defined the log likelihood as a vector named log lik in the generated quantities block so that the individual terms will be saved by Stan. After running Stan, log lik can be extracted (using the extract log lik function provided in the LOO package).

9.1 Computing Approximate Leave-one-out cross-Validation Usig PSIS-LOO and WAIC

We can use the R package LOO provides the functions LOO() and waic() for efficiently computing PSIS-LOO and WAIC for fitted Bayesian models. Below, we provide R code for preparing and running the Logarithm Transformed Exponential Distribution(LTE) and Logarithm Transformed Weibull Distribution (LTW) for the (Veteran’s Administration Lung Cancer Data) in Stan. After fitting the model we then use the LOO package to compute LOO and WAIC.

9.1.1 For Logarithm Transformed Exponential (LTE) Model

WAIC and LOO are computed with the functions waic and loo, respectively

```
library("loo")

# Extract pointwise log-likelihood and compute LOO

log_lik_1 <- extract_log_lik(M1, merge_chains = FALSE)

# as of loo v2.0.0 we can optionally provide relative effective sample sizes

# when calling loo, which allows for better estimates of the PSIS effective

# sample sizes and Monte Carlo error

r_eff1 <- relative_eff(exp(log_lik_1))

loo_1 <- loo(log_lik_1, r_eff = r_eff1, cores = 2)

print(loo_1)

# compare the waic

waic1 <- waic(log_lik_1)

print(waic1)
```

Table 3

	Estimate	SE
elpd_loo	-781.1	15.2
p_loo	9.0	1.7
Looic	1563.8	29.8
elpd_waic	-781.4	15.3
p_waic	8.9	1.8
waic	1562.8	30.5

The printed output from the LOO function shows the estimates elpd loo (expected log predictive density), p loo (effective number of parameters), and looic=-2elpd loo (the LOO information criterion).

9.1.2 For Logarithm Transformed Weibull (LTW) Model

```
ibrary("loo")
# Extract pointwise log-likelihood and compute LOO
log_lik_3 <- extract_log_lik(M3, merge_chains = FALSE)
# as of loo v2.0.0 we can optionally provide relative effective sample sizes
# when calling loo, which allows for better estimates of the PSIS effective
# sample sizes and Monte Carlo error
r_eff <- relative_eff(exp(log_lik_3))
loo_3 <- loo(log_lik_3, r_eff = r_eff, cores = 2)
print(loo_3)
# compare the waic
waic3 <- waic(log_lik_3)
print(waic3)
```

Table 4

	Estimate	SE
elpd_loo	-780.8	14.8
p_loo	8.1	1.4
Looic	1561.7	29.8
elpd_waic	-780.7	14.9
p_waic	8.0	1.5
waic	1561.5	29.8

The printed output from the loo function shows the estimates elpd loo (expected log predictive density), p loo (effective number of parameters), and looic=-2elpd loo (the LOO information criterion).

In tables(4 & 5) above, the WAIC for each model as well as the difference of the WAICs each with its corresponding standard error SE is shown. Both, WAIC and LOO are approximately normal if the number of observations is large so that the standard errors can be very helpful in evaluating differences in the information criteria. However, for small sample sizes, standard errors should be interpreted with care [23].

10. Conclusion

We can now compare the models on LOO using the compare function. This object, comp, contains the estimated difference of expected leave-one-out prediction errors between the two models, along with the standard error.

```
# Compare

comp1 <- compare(loo_1, loo_2)

print(comp1)

comp2 <-compare(waic1, waic2)

print(comp2)
```

Table 5. compare the different between Logarithm Transformed Exponential (LTE) Model and Logarithm Transformed Weibull (LTW) Model

	elpd diff	se
Looic	0.7	1.5
Waic	0.8	1.4

The positive difference in elpd and its scale relative to the standard error, indicates a preference for the second model (Logarithm Transformed Weibull (LTW) Model).

Table 6. Model comparison of Logarithm Transformed Exponential (LTE) Model and Logarithm Trans- formed Weibull (LTW) Model for the Intrauterine device (IUD) data. It is evident from this table that Logarithm Transformed Weibull (LTW) Model is much better than Logarithm Transformed Exponential (LTE) Model.

Models	Stan Deviance	WAIC	loo
Logarithm Transformed Exponential (LTE) Model	1458.5	1562.8	1563.8
Logarithm Transformed Weibull (LTW) Model	1457.1	1561.5	1561.7

Table 6 lists the WAIC and LOO values for the two Logarithm Transformed (LT) models, both of which lead to the same conclusion that the Logarithm Transformed Weibull (LTW) model has the smallest values of LOO and WAIC. Consequently, it was concluded that the Logarithm Transformed Weibull (LTW) model was the best-fitting model among the Logarithm Transformed Exponential (LTE).

References

AbuJarad, M. H., & Khan, A. A. (2018a). Exponential Model: A Bayesian Study with Stan.Int J Recent Sci Res, 9(8), 28495-28506.

AbuJarad, M. H., & Khan, A. A. (2018b). Bayesian Survival Analysis of Topp-Leone Generalized Family with Stan.

Bennett, S. (1983). Log-logistic regression models for survival data.Applied Statistics 32, 165-171. [4]Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., ... & Riddell,

A. (2017).Stan: A probabilistic programming language.Journal of statistical software, 76(1).

Chen, K., Jin, Z., and Ying, Z. (2002). Semiparametric analysis of transformation models with cen- sored data.Biometrika 89, 659-668.

Gelman, A., & Rubin, D. B. (1992). A single series from the Gibbs sampler provides a false sense of security.Bayesian statistics, 4, 625-631.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). Bayesian data analysis.Chapman and Hall/CRC.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). Bayesian analysis, 1(3), 515-534.

Gelman, A. (2008). Scaling regression inputs by dividing by two standard deviations.Statistics in medicine, 27(15), 2865-2873.

Hoffman, M. D., & Gelman, A. (2014), The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo., Journal of Machine Learning Research 15(1), 1593-1623.

Kalbfleisch, J. D., & Prentice, R. L. (1980). The statistical analysis of failure time data. 2002. Hobo- ken, NJ.

Maurya, S. K., Kaushik, A., Singh, R. K., Singh, S. K., & Singh, U. (2016), A new method of proposing distribution and its application to real data., Imperial Journal of Interdisciplinary Research 2(6). 65-70.

Murphy, S. A., Rossini, A. J., and Vaart, A. W. V. D. (1997). Maximum likelihood estimation in the proportional odds model.Journal of the American Statistical Association 92, 968-976.

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo, 2(11), 2.

Plummer, M. (2008). Penalized loss functions for Bayesian model comparison.Biostatistics, 9(3), 523-539.

Prentice, R. L. (1973). Exponential survivals with censoring and explanatory variables.Biometrika 60, 279-284.

Stan Development Team. (2016). Stan modeling language users guide and reference manual,version 2.14. 0. Technical report.

Statisticat, L. L. C. (2015). LaplacesDemon: complete environment for Bayesian inference. R package version, 15(01).

Team, R. C. (2014). R: A language and environment for statistical computing.Vienna, Austria: R Foundation for Statistical Computing; 2014.

Vehtari, A., Gelman, A., & Gabry, J. (2017a), Practical Bayesian model evaluation using leave-one- out cross-validation and WAIC., Statistics and Computing, 27(5), 1413-1432.

Vehtari, A., Gelman, A., and Gabry, J. (2017b), Pareto smoothed importance sampling.,arXiv preprint: http://arxiv.org/abs/1507.02646/

Van Der Linde, A. (2005). DIC in variable selection.Statistica Neerlandica, 59(1), 45-56. [23]Vehtari, A., Gelman, A., & Gabry, J. (2015). Efficient implementation of leave-one-out cross-

validation and WAIC for evaluating fitted Bayesian models. arXiv preprint arXiv:1507.04544.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable in- formation criterion in singular learning theory.Journal of Machine Learning Research, 11(Dec), 3571-3594.

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018), Using stacking to average Bayesian predictive distributions. Bayesian Analysis,advance publication, doi:10.1214/17-BA1091.