Austrian Journal of Statistics
July 2021, Volume 50, 52–76.
http://www.ajs.or.at/
doi:10.17713/ajs.v50i5.1166

Generalized Topp-Leone-Weibull AFT Modelling: A Bayesian Analysis with MCMC Tools Using R and Stan

Md. Ashraf-Ul-Alam
Aligarh Muslim University,
Department of Statistics and
Operations Research

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

Abstract

The generalized Topp-Leone-Weibull (GTL-W) distribution is a generalization of Weibull distribution which is obtained by using generalized Topp-Leone (GTL) distribution as a generator and considering Weibull distribution as a baseline distribution. Weibull distribution is a widely used survival model that has monotone- increasing or decreasing hazard. But it cannot accommodate bathtub shaped and unimodal shaped hazards. As a survival model, GTL-W distribution is more flexible than the Weibull distribution to accommodate different types of hazards. The present study aims at fitting GTL-W model as an accelerated failure time (AFT) model to censored survival data under Bayesian setting using `R` and `Stan` languages. The GTL-W AFT model is compared with its sub-model and the baseline model. The Bayesian model selection criteria `LOOIC` and `WAIC` are applied to select the best model.

Keywords: generalized Topp-Leone-Weibull distribution, baseline distribution, censored data, monotone hazard, accelerated failure time, Stan, LOOIC, WAIC.

1. Introduction

Topp and Leone (1955) introduced a new probability distribution having one parameter on the support (0, 1). Nadarajah and Kotz (2003) discussed this distribution and derived many relevant functions and quantities, for instance, characteristic function, moments. The cumulative distribution function (cdf) and probability density function (pdf) of a random variable `X` having Topp-Leone (TL) distribution with shape parameter $a > 0$ and $0 < x < 1$ are given as follows:

$$F(x|a) = x^a(2-x)^a = \{x(2-x)\}^a$$

$$f(x|a) = 2a(1-x)x^{a-1}(2-x)^{a-1}$$

Since the support of TL distribution is (0, 1), so it can not be used in many areas including in survival analysis. Recently, Sangsanit and Bodhisuwan (2016) and Al-Shomrani, Arif, Shawky, Hanif, and Shahbaz (2016) extended TL distribution as the Topp-Leone generated (TL-G) family of distributions. Suppose, $G(x) = G(x|\xi)$, and $g(x) = g(x|\xi)$ be the baseline cdf and pdf of a continuous random variable $\mathtt{X}$. If TL distribution is considered as a generator then the TL-G family of distributions is obtained by replacing x with $G(x)$ in the TL distribution and is given as below:

$$F(x|a,\xi) = G(x)^a[2-G(x)]^a = \{G(x)[2-G(x)]\}^a \tag{1}$$

$$f(x|a,\xi) = 2ag(x)[1-G(x)]G(x)^{a-1}[2-G(x)]^{a-1} \tag{2}$$

where $f(x) = \frac{d}{dx}F(x)$, $g(x) = \frac{d}{dx}G(x)$, support of $F(x|a,\xi)$ is the support of $G(x)$ and ξ is the vector of parameters of baseline distribution $G(x)$ and we can write $\mathtt{X} \sim$ TL-G (a,ξ). Further generalization of TL-G model is considered in the next section.

1.1. Generalized Topp-Leone-G model

Rezaei, Sadr, Alizadeh, and Nadarajah (2017) generalized the TL-G(a,ξ) distribution by exponentiating the baseline cdf $G(x)$ with an additional shape parameter $b > 0$. Mahdavi (2017) made generalization of TL-G model with different consideration. Motivated by Cordeiro and de Castro (2011) , in this article, the generalization of TL-G family due to Rezaei *et al.* (2017) is called as the generalized Topp-Leone-G (GTL-G) family of distributions. So the GTL distribution is considered as the generator of the new family, GTL-G model. The cdf, survival function (sf) and pdf of GTL-G model can be written as:

$$F(x|a,b,\xi) = \{G(x)^b[2-G(x)^b]\}^a \tag{3}$$

$$S(x|a,b,\xi) = 1-\{G(x)^b[2-G(x)^b]\}^a \tag{4}$$

$$f(x|a,b,\xi) = 2abg(x)[1-G(x)^b]G(x)^{ab-1}[2-G(x)^b]^{a-1} \tag{5}$$

where, shape1 $a > 0$, shape2 $b > 0$, and ξ is the vector of parameters of baseline distribution $G(x)$, support of $F(x|a,b,\xi)$ is the support of $G(x)$ and we can write $\mathtt{X} \sim$ TL-G (a,ξ). The GTL-G model will be tractable if the baseline cdf $G(x)$ and pdf $g(x)$ have simple analytic forms. Considering the cdf of Weibull distribution $G(x)$ as the baseline distribution, the GTL-W distribution is derived and discussed as a survival model with Bayesian perspective.

Random number generation from GTL-G model:
Suppose that $\mathtt{U} \sim$ Uniform(0, 1). Then

$$F(x) = F(x|a,b,\xi) = u$$

$$\{G(x)^b[2-G(x)^b]\}^a = u$$

$$G(x) = \left\{1-\sqrt{(1-u^{1/a})}\right\}^{1/b} \tag{6}$$

This is the general expression for generating random numbers from GTL-G model for any baseline cdf $G(x)$.

2. Weibull survival model

Suppose, survival time $\mathtt{T}$ follows Weibull distribution with shape $\alpha(>0)$ and scale parameter $\lambda(>0)$. Then the pdf $g(t)$, cdf $G(t)$, survival function (sf) $\bar{G}(t)$ and hazard function (hf) $h(t)$ of Weibull distribution are given as below (Lawless 2003):

$$g(t|\alpha,\lambda) = (\alpha/\lambda)(t/\lambda)^{(\alpha-1)}\mathtt{exp}[-(t/\lambda)^\alpha], \quad t > 0$$

$$G(t|\alpha,\lambda) = 1-\mathtt{exp}[-(t/\lambda)^\alpha]$$

$$\bar{G}(t|\alpha,\lambda) = \texttt{exp}[-(t/\lambda)^{\alpha}]$$
$$h(t|\alpha,\lambda) = (\alpha/\lambda)(t/\lambda)^{(\alpha-1)}$$

and we write $\texttt{T}\sim$ Weibull(α,λ). The Weibull distribution is one of the basic lifetime distributions used in survival analysis. The hazard function of Weibull distribution is monotone-increasing or decreasing. Moreover, its survival and hazard functions can be expressed explicitly and as such, it can handle censored data easily and effectively. Because of flexibility and tractability of hazard and survival functions, Weibull model is widely used in modelling reliability and survival data. But the distribution is not suitable for analyzing data having bathtub shaped and unimodal shaped hazard rates. So to capture different types of hazards, more flexible generalization of the Weibull model is essential in fitting skewed survival data.

2.1. Weibull AFT model

The accelerated failure time (AFT) model considers linear regression of logarithm of survival time $\texttt{T}$ on a number of covariates and is used for studying the effect of a covariate that accelerates or decelerates the survival process. The AFT models are parametric models as they depend on the distribution of survival times. The AFT model postulates a direct relationship between failure time and covariates (Kalbfleisch and Prentice 2002). The models are becoming popular as they directly explain reduction or prolongation of the time to event. If survival time $\texttt{T}$ depends on a number of covariates $x_1, x_2, \ldots, x_p$ then the AFT model can be written as:

$$\texttt{log}(T) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \sigma z = \mathbf{x}'\beta + \sigma\mathbf{Z}$$

where, $\beta_j, j = 0, 1, 2, \ldots, p$ are the regression coefficients, $\sigma(>0)$ is a scale parameter, and random error $\mathbf{Z}$ has a specified probability distribution. For instance, if $\mathbf{Z}$ has an extreme value distribution then $\texttt{T}$ has Weibull distribution (Lee and Wang 2013; Collett 2015).

Suppose that a random variable $\mathbf{Z}$ has a standard extreme value distribution with density function $g(z) = \texttt{exp}[z - \texttt{exp}(z)]$ and survival function $S(z) = \texttt{exp}[-\texttt{exp}(z)]$. Substituting $z = (\texttt{log}t - \mathbf{x}'\beta)/\sigma$ in the extreme value distribution, Weibull AFT model, $\texttt{T} \sim \texttt{Weibull}(\frac{1}{\sigma}, \texttt{exp}(\mathbf{x}'\beta))$, is obtained as follows:

$$g(t|\sigma,\beta,\mathbf{x}) = \Big(\frac{\sigma^{-1}}{\texttt{exp}(\mathbf{x}'\beta)}\Big)\Big(\frac{t}{\texttt{exp}(\mathbf{x}'\beta)}\Big)^{(\frac{1}{\sigma}-1)}\texttt{exp}\Big(-\Big(\frac{t}{\texttt{exp}(\mathbf{x}'\beta)}\Big)^{\frac{1}{\sigma}}\Big) \tag{7}$$

$$G(t|\sigma,\beta,\mathbf{x}) = 1 - \texttt{exp}\Big(-\Big(\frac{t}{\texttt{exp}(\mathbf{x}'\beta)}\Big)^{1/\sigma}\Big) \tag{8}$$

$$\bar{G}(t|\sigma,\beta,\mathbf{x}) = 1 - G(t|\sigma,\beta,\mathbf{x}) = \texttt{exp}\Big(-\Big(\frac{t}{\texttt{exp}(\mathbf{x}'\beta)}\Big)^{1/\sigma}\Big) \tag{9}$$

$$h(t|\sigma,\beta,\mathbf{x}) = g(t)/\bar{G}(t) = \Big(\frac{\sigma^{-1}}{\texttt{exp}(\mathbf{x}'\beta)}\Big)\Big(\frac{t}{\texttt{exp}(\mathbf{x}'\beta)}\Big)^{(1/\sigma-1)} \tag{10}$$

Considering the Weibull AFT model as the baseline model, GTL-W AFT model is developed.

2.2. Generalized Topp-Leone-Weibull AFT model

Aryal, Ortega, Hamedani, and Yousof (2017) introduced Topp-Leone generated Weibull distribution and illustrated with real data sets using maximum likelihood method of estimation and showed that the model fit the data better than the other competitive models. GTL distribution is considered as a generator of the new family of models and Weibull AFT model is considered as the baseline model G. The generalized Topp-Leone-Weibull (GTL-W) AFT model is obtained by substituting Weibull AFT model in the GTL-G model. The cdf, sf, pdf and hf of the GTL-W AFT model can be written as follows:

$$F(t|\theta,\mathbf{x}) = \{G(t)^b[2-G(t)^b]\}^a \tag{11}$$

$$S(t|\theta,\mathbf{x}) = 1-\{G(t)^b[2-G(t)^b]\}^a \tag{12}$$

$$f(t|\theta,\mathbf{x}) = 2ag(t)[1-G(t)]G(t)^{a-1}[2-G(t)]^{a-1} \tag{13}$$

$$h(t|\theta,\mathbf{x}) = f(t|\theta,\mathbf{x})/S(t|\theta,\mathbf{x}) \tag{14}$$

where, $t>0$, $g(t)=g(t|\sigma,\beta,\mathbf{x})$ and $G(t)=G(t|\sigma,\beta,\mathbf{x})$ are the pdf and cdf of Weibull AFT model (Equation 7, 8), $\theta=(a,b,\sigma,\beta)$, shape1 $a>0$, shape2 $b>0$, shape3 $\alpha>0$, and scale parameters $\lambda>0$; moreover, $\sigma=1/\alpha$, $\lambda=\texttt{exp}(\mathbf{x}'\beta)$ from the AFT model $\texttt{log}(T)=\mathbf{x}'\beta+\sigma\mathbf{Z}$ and we write $T\sim \texttt{GTL-W}(a,b,1/\sigma,\texttt{exp}(\mathbf{x}'\beta))$. Since the GTL-W distribution has explicit forms of survival function and hazard function, so, it can be used properly as a survival model to analyze censored data under Bayesian framework.

Random number generation from GTL-W AFT model:
Suppose, $U\sim$ Uniform(0, 1). Then

$$F(x) = F(x|a,b,\alpha,\lambda) = u$$

$$G(x) = \{1-\sqrt{(1-u^{1/a})}\}^{1/b}$$

$$1-\texttt{exp}[-(x/\lambda)^\alpha] = \{1-\sqrt{(1-u^{1/a})}\}^{1/b}$$

$$x = (\texttt{exp}(\mathbf{x}'\beta))\Big\{-\texttt{log}\Big[1-\{1-\sqrt{(1-u^{1/a})}\}^{1/b}\Big]\Big\}^\sigma \tag{15}$$

This is the expression for generating random numbers from GTL-W AFT model, where, $\sigma=1/\alpha$ and $\lambda=\texttt{exp}(\mathbf{x}'\beta)$.

3. Description of data

Survival times of 80 patients with *diffuse histiocytic lymphoma* according to stage of tumour-stage III and stage IV are given in Table 1. The death times are recorded in days (* indicates censored time). Vidakovic (2011) and Armitage, Berry, and Matthews (2002) discussed the data from different aspects and the data was first reported by McKelvey, Gottlieb, Wilson, Haut, Talley, Stephens, Lane, Gamble, Jones, Grozea *et al.* (1976).
The aim of this study is to fit GTL-W AFT model for comparing the survival patterns of two groups of patients under Bayesian framework and the fitted model is compared with its sub-model and baseline Weibull AFT model on the basis of `LOOIC` and `WAIC`.

4. Bayesian analysis of GTL-W AFT model

The basic assumption of Bayesian statistics is that parameters θ in a probability model are random variables having prior distributions $\pi(\theta)$ while the observed data drawn from a sampling density are considered as fixed. Bayes Theorem is the main instrument of Bayesian statistics that establishes the relationship among the prior distributions, likelihood and the posterior distribution.

4.1. Bayes theorem

Suppose that $f(\mathbf{y}|\theta)$ be the density of data having parameter θ that may be a vector and from this density observed data values $\mathbf{y}=(y_1,y_2,\ldots,y_n)$ are obtained independently. The Bayes Theorem states that (Gelman, Stern, Carlin, Dunson, Vehtari, and Rubin 2013),

Table 1: Survival times (in days) of patients with diffuse histiocytic lymphoma according to stage of tumour

Stage III (19 patients)
6, 19, 32, 42, 42, 43*, 94, 126*, 169*, 207, 211*,
227*, 253, 255*, 270*, 310*, 316*, 335*, 346*
Stage IV (61 patients)
4, 6, 10, 11, 11, 11, 13, 17, 20, 20, 21, 22, 24, 24, 29,
30, 30, 31, 33, 34, 35, 39, 40, 41*, 43*, 45, 46, 50, 56,
61*, 61*, 63, 68, 82, 85, 88, 89, 90, 93, 104, 110, 134,
137, 160*, 169, 171, 173, 175, 184, 201, 222, 235*,
247*, 260*, 284*, 290*, 291*, 302*, 304*, 341*, 345*

$$\pi(\theta|\mathbf{y}) = \frac{f(\mathbf{y},\theta)}{h(\mathbf{y})} = \frac{\pi(\theta)f(\mathbf{y}|\theta)}{\int \pi(\theta)f(\mathbf{y}|\theta)\,d\theta} = \frac{L(\theta|\mathbf{y})\pi(\theta)}{\int \pi(\theta)f(\mathbf{y}|\theta)\,d\theta} \tag{16}$$

where, $\pi(\theta|\mathbf{y})$ is the posterior distribution of parameter θ given the data $\mathbf{y}$, $\pi(\theta)$ is the prior distribution of θ that is selected independent of the data and the likelihood function is given by

$$L(\theta|\mathbf{y}) = f(\mathbf{y}|\theta) = f(y_1, y_2, \ldots, y_n|\theta) = \prod_{i=1}^{n} f(y_i|\theta)$$

$h(\mathbf{y}) = \int \pi(\theta)f(\mathbf{y}|\theta)d\theta$ is the marginal distribution of $\mathbf{y}$ which is independent of θ and considered as normalizing constant. So dropping $h(\mathbf{y})$ from Equation (16), we can write

$$\pi(\theta|\mathbf{y}) \propto L(\theta|\mathbf{y}) \times \pi(\theta)$$

$$\textbf{Posterior} \propto \textbf{Likelihood} \times \textbf{Priors} \tag{17}$$

Equation (17) is the main essence of Bayes Theorem. That is, the posterior distribution of parameters are obtained by combining prior distribution and data. All inferences are based on the posterior distribution.

4.2. Likelihood function of GTL-W AFT model

The likelihood function of GTL-W AFT model for a right censored sample is derived as (Collett 2015; Lawless 2003; Wang, Ryan, and Faraway 2018):

$$\begin{aligned} L(\theta|D) &= \prod_{i=1}^{n} f(t_i|a,b,\sigma,\beta,\mathbf{x})^{\delta_i} S(t_i|a,b,\sigma,\beta,\mathbf{x})^{1-\delta_i} \\ &= \prod_{i=1}^{n} \left(\frac{f(t_i|a,b,\sigma,\beta,\mathbf{x})}{S(t_i|a,b,\sigma,\beta,\mathbf{x})}\right)^{\delta_i} S(t_i|a,b,\sigma,\beta,\mathbf{x}) \\ &= \prod_{i=1}^{n} \Big(h(t_i|a,b,\sigma,\beta,\mathbf{x})\Big)^{\delta_i} S(t_i|a,b,\sigma,\beta,\mathbf{x}) \\ &= \prod_{i=1}^{n} \Big(h(t_i|\theta,\mathbf{x})\Big)^{\delta_i} S(t_i|\theta,\mathbf{x}) \end{aligned} \tag{18}$$

where, $S(t_i|a,b,\sigma,\beta,\mathbf{x})$, $f(t_i|a,b,\sigma,\beta,\mathbf{x})$ and $h(t_i|a,b,\sigma,\beta,\mathbf{x})$ of GTL-W AFT model are given in Equation (12, 13, 14), $\theta = (a,b,\sigma,\beta)$, $D = (\mathbf{t},\delta,\mathbf{X})$, the censoring indicator $\delta = 0$ if the observation is censored and $\delta = 1$ if the observation is failed, and $\mathbf{X}$ is the matrix of covariates which is known as design matrix or model matrix. The contribution of data to the posterior distribution is ensured through the likelihood.

Taking logarithm of both sides of likelihood function, the log-likelihood can be written by the following two alternative equations,

$$l(\theta|D) = \sum_{i=1}^{n} \Big(\delta_i(\texttt{log} f(t_i|\theta, \mathbf{x}) - \texttt{log} S(t_i|\theta, \mathbf{x})) + \texttt{log} S(t_i|\theta, \mathbf{x}) \Big)$$

$$l(\theta|D) = \sum_{i=1}^{n} \Big(\delta_i(\texttt{log} h(t_i|\theta, \mathbf{x})) + \texttt{log} S(t_i|\theta, \mathbf{x}) \Big)$$

4.3. Prior distributions of GTL-W AFT model parameters

Selecting prior distributions of the parameters of a probability model of data is very important as they might have significant influence on the posterior distribution of the parameters. GTL-W density has two shape parameter(a, b), a positive scale parameter (σ) and regression coefficients (β). So prior distributions for the parameters $\theta = (a, b, \sigma, \beta)$ are required to be selected. For scale parameter σ, half-Cauchy distribution and for the regression coefficients β, normal distribution are considered as weakly informative prior distribution (Gelman *et al.* 2006). But Cauchy sampling process may not converge to a stable mean and variance (McElreath 2015). So in this article we consider half-t distribution (Wiper, Girón, and Pewsey 2008) as prior for the scale parameter and for the shape parameters as well. Half-t distribution with one degrees of freedom is equivalent to half-Cauchy distribution. The hyperparameters of the prior distributions are chosen so that priors are almost flat (Figure 1), as such, the data dominate the posterior through the likelihood.

Truncated student's t distribution restricted to the interval [0, ∞] is called the half-t distribution. For the scale parameter, $\pi(\sigma)$ =half-t(df, location, scale) = student_t(2, 0, 10) T[restricted interval] = student_t(2, 0, 10) T[0,], for the shape parameters, $\pi(a) = \pi(b) =$ student_t(5, 0, 10) T[0,] and for the regression coefficients, $\pi(\beta_j) = \text{normal}(\mu, \sigma) =$ normal(0, 10) are considered as priors. Stan has provision to write truncated distributions. The parameters are assumed to be independent and their joint prior density function can be written as:

$$\pi(\theta) = p(a, b, \sigma, \beta) = \pi(a) \times \pi(b) \times \pi(\sigma) \times \pi(\beta) \tag{19}$$

4.4. Posterior distribution of GTL-W AFT model parameters

The joint posterior distribution of the parameters $\theta = (a, b, \sigma, \beta) = (a, b, \sigma, \beta_0, \beta_1, \ldots, \beta_p)$ of GTL-W AFT model given the data can be written following Bayes Theorem as:

$$\pi(\theta|D) \propto L(\theta|D) \times \pi(\theta)$$

$$\pi(\theta|D) \propto L(\theta|D) \times \pi(a) \times \pi(b) \times \pi(\sigma) \times \pi(\beta) \tag{20}$$

where the likelihood $L(\theta|D)$ is given by Equation (18), parameters are assumed to be independent.

The normalized joint posterior distribution and the marginal distributions of the parameters are difficult to obtain analytically that requires solving high dimensional integration and, as such, Markov chain Monte Carlo (MCMC) simulation technique is employed to get the estimates and other relevant results.

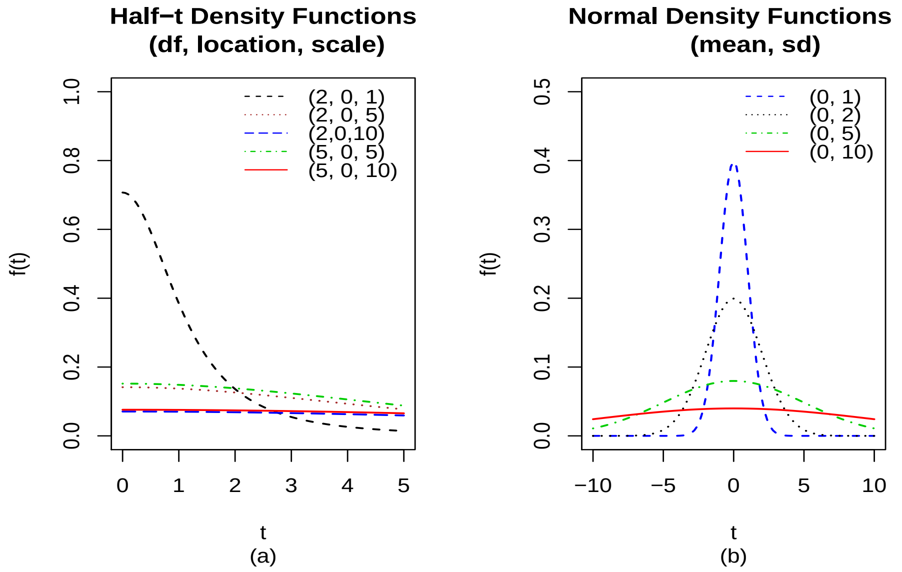

Figure 1: Density curves of half-t and normal distribution for different values of the parameters

4.5. Rationale of using Bayesian methods to survival analysis

Bayesian methods are becoming increasingly popular among the survival analysis researchers. Ibrahim, Chen, and Sinha (2001) and Li and Ma (2013) highlighted some reasons of applying Bayesian methods to survival analysis. As the survival time is non-negative, usually positively skewed and censored, so the least squares method is not applicable for analyzing survival data. The likelihood function is the most important way of handling survival data that considers each of the censored and uncensored observations. But the likelihood function for censored data is complicated and different from likelihood function for a sample of complete observations. The maximum likelihood method of estimation can be applied and sampling distribution can be derived for inference procedures only for simple distributions for some censoring schemes, not for all types of censoring. Large sample asymptotic sampling distribution is difficult to derive for censored data, even that is not applicable for small sized censored sample. Bayesian methods are very useful in survival analysis as it can analyze censored survival data effectively and efficiently irrespective of sample size and can handle complex models with any number of parameters using MCMC algorithms.

4.6. Stan code for fitting Bayesian GTL-W AFT model

Stan is a probabilistic programming language for Bayesian analysis. It uses no-U-turn sampler (NUTS), an adaptive form of Hamiltonian Monte Carlo (HMC) sampling that is more efficient than other Metropolis-Hastings algorithms, specially for high-dimensional models regardless of whether the priors are conjugate or not (Carpenter, Gelman, Hoffman, Lee, Goodrich, Betancourt, Brubaker, Guo, Li, and Riddell 2017; Hoffman and Gelman 2014). It is easier to specify the constraints of the parameters in Stan. A prior can be specified in Stan, but if prior is not specified, Stan uses uniform prior for each parameter. Bayesian softwares BUGS or JUGS is flexible enough to specify a large variety of models, but for large data sets and complex models they might take too long computing time or even may fail to provide solutions. Stan can handle large data sets and complex models in less computing time than BUGS. The number of iterations needed to achieve convergence is lower using Stan. Stan does not allow discrete parameters. As Stan is a recently developed software, not much literature is available on it. Gelman *et al.* (2013), Kruschke (2015), Korner-Nievergelt, Roth, Von Felten, Guélat, Almasi, and Korner-Nievergelt (2015) and Turkman, Paulino, and Müller (2019) discussed Stan to some extent. However, a comprehensive Stan manual is available online. Stan is becoming popular and it is being rapidly developed by the Stan Development Team.

A complete Stan program consists of six code blocks. Sequentially, the blocks are: (i) data block (ii) transformed data block, (iii) parameters block, (iv) transformed parameters block (v) model block (vi) generated quantities block. A sequence of programming statements surrounded by curly braces {} form a block. A statement ends with a semi-colon. A comment in Stan are indicated by a double slash //. The Stan Development Team (2017) called the language as 'Stan' in honor of Stanislaw Ulam, one of the creators of the Monte Carlo methods.

The six code blocks for implementing Bayesian GTL-W AFT model are shown in the appendix. The model block is mandatory that contains the model specification. Priors and likelihood are mentioned in model block. All other blocks are optional. In generated quantities block, the pointwise log-likelihood `log_lik` and posterior prediction `y_rep` are calculated and are stored for future use. Three functions, namely, (i) log survival function, (ii) log hazard function and (iii) log-likelihood function are defined at the beginning, before starting the code blocks. Stan codes are written in the editor `RStudio` . Stan has interface with `R` by `rstan` .

Stan codes are written with comments for fitting the GTL-W AFT model under Bayesian framework and are given in the appendix. The whole code block is saved as `stancode_gtlw` that is used afterwards.

Data structure for Stan to fit GTL-W model:
Stan requires a data list that might include a matrix, vector and values. So, data must be prepared to feed into Stan. An object `list` is constructed using `R` and assigned it to `dat1` (say).

```
# survival times in days
y <- c(6,19,32,42,42,43,94,126,169,207,211,227,
       253,255,270,310,316,335,346,4,6,10,11,11,
       11,13,17,20,20,21,22,24,24,29,30,30,31,33,
       34,35,39,40,41,43,45,46,50,56,61,61,63,68,
       82,85,88,89,90,93,104,110,134,137,160,169,
       171,173,175,184,201,222,235,247,260,284,
       290,291,302,304,341,345)
# event=1 indicates death and 0 indicates censored
event <- c(rep(1,5),0,1,0,0,1,0,0,1,rep(0,6),
           rep(1,23),0,0,rep(1,4),0,0,rep(1,12),0,
           rep(1,7),rep(0,10))
# x1=0 if disease stage is III and 1 if stage is IV
x1 <- c(rep(0,19),rep(1,61))
x = cbind(1,x1) # model matrix
N = nrow(x)
M = ncol(x)
dat1 = list(y=y,event=event,x=x,N=N,M=M)
```

Model fitting:
To fit the GTL-W AFT model under Bayesian framework and to simulate from the posterior distribution, the function `stan()` from the package `rstan` is called and a Stanfit object `M1` (say) is created. Stan has default initialization strategy. But initial values can be supplied by the user for avoiding divergent transitions. To decrease divergent transitions, the argument `control=list(adapt_delta=0.9995` is also used. In Stan, default choice for chain and iteration are 4 and 2000 respectively. We have fixed 4 chains and 4000 iterations for each chain, that is, for each of the 4 chains 4000 samples are drawn for each of the parameters. Stan uses half of the iterations as warmup

iterations, so post-warmup draws per chain is 2000. Relevant codes are given in the appendix.

Summarizing output of Stanfit GTL-W AFT model:
Using `print()` command, summary results are obtained from the fitted object `M1` and are reported in Table 2. Trace plots are made for visual convergence checking. For posterior plot `bayesplot` package and for model comparison criteria `LOOIC` and `WAIC`, `loo` package are used and they are reported in Table 5. Codes are explained in the appendix.

Table 2: Summary results of fitted GTL-W AFT model

Parameter	mean	se_mean	sd	2.5%	50%	97.5%	n_ eff	$\hat{R}$
beta[1]	-3.248	0.105	4.379	-12.274	-10.740	4.656	1744	1.001
beta[2]	-1.046	0.007	0.474	-2.000	-1.843	-0.121	4248	1.000
shape1	10.418	0.141	8.344	1.283	1.799	31.942	3494	1.001
shape2	9.607	0.139	7.394	1.037	1.535	28.078	2825	1.001
sigma	7.524	0.052	2.220	3.291	3.918	11.978	1839	1.001

Convergence of MCMC algorithm for GTL-W AFT model:
An MCMC algorithm simulates samples from the complex posterior distribution. So an algorithm is converged to the target posterior distribution means that the Markov chain is stationary and adding more samples will not change the location and shape of the density of the posterior distribution sensibly and hence will not change the estimates and other relevant results. Quantitatively, *potential scale reduction factor*, $\hat{R}$ (Gelman and Rubin 1992), is used for checking convergence of MCMC algorithms which is defined based on between chain variance and within chain variance. Gelman *et al.* (2013) recommended acceptable limit of potential scale reduction factor $\hat{R} < 1.1$ and effective sample size as $n\_eff = 100$.

From the summary features (Table 2), it is evident that $\hat{R}$ is less than 1.1, n_eff is greater than 100; moreover, *Monte Carlo (MC) error*, *se_mean* is less relative to the standard deviations for all of the parameters which is expected, that is, convergence of MCMC algorithm to the posterior distribution has been attained.

Graphically, convergence of an MCMC algorithm can be judged, mainly, by *trace plot* (Gelman *et al.* 2013; Ntzoufras 2009; Hamra, MacLehose, and Richardson 2013). Trace plot (Figure 2) shows no periodicity indicating convergence of the MCMC sampling process to the joint posterior distribution. That is, MCMC algorithm HMC-NUTS performs correctly to explore the target posterior distribution.

Evaluating Bayesian fitting of GTL-W AFT model:
A fitted Bayesian model is accepted as adequate if it predicts the future observations that are consistent with the present data. Model fit is assessed visually by *posterior predictive density (PPD)* plots which are made using `bayesplot` package. From posterior predictive density plots (Figure 3), it is observed that GTL-W AFT model is compatible with the data.

Interpretation of results of fitted GTL-W AFT model:
Summary features of the posterior distribution of the parameters of GTL-W AFT model are shown in Table 2. In regression modelling with Stan, intercept is denoted as $\beta[1]$. The coefficient $\beta[2]$ of covariate `x1`, stage of tumour, is negative which means that death will be accelerated for patients with stage IV tumour, consequently, length of lifetime

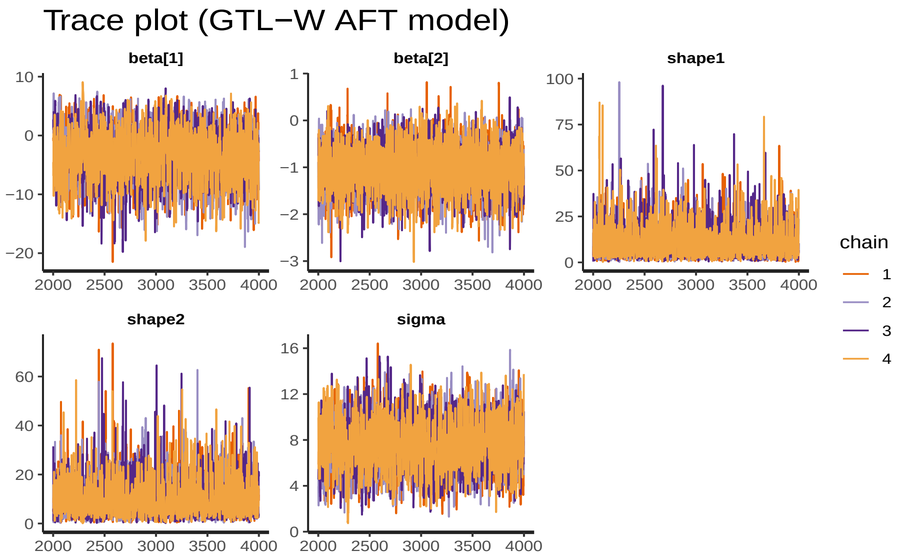

Figure 2: Trace plot of fitted GTL-W AFT model are obtained by plotting parameter values along the Y-axis against their corresponding iterations along the X-axis and there is no tendency of periodicity of the plot indicating convergence of the algorithm

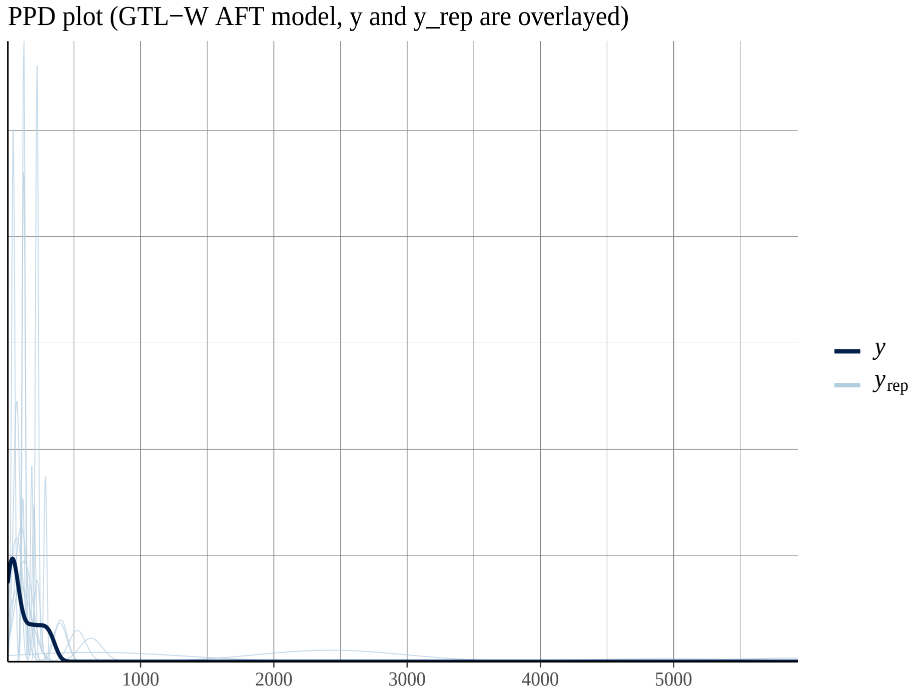

Figure 3: Posterior predictive density (PPD) plot of GTL-W AFT model is done by plotting the data y and then overlaying the density of the predicted values *y_rep*. PPD plot of GTL-W AFT model shows that the posterior predictive density fits the data well

will decrease. Estimated value of the coefficient $\beta[2] = -1.046$ belongs to the 95% credible interval $(-2.000, -0.121)$ which does not include zero value indicates statistical significance. Moreover, it is seen from the caterpillar plot (Figure 4) that zero value is not contained in the 95% credible interval for the coefficient $\beta[2]$, so the coefficient is statistically significant. The acceleration factor is $\texttt{exp}(\beta_2) = \texttt{exp}(-1.046) = 0.35(< 1)$ for a patient with stage IV tumour. Therefore, duration of life of a patient with stage IV tumour would be 0.35 times shorter than that of a patient having tumour of stage III , that is, survival time would be shorter for the patients with stage IV tumour.

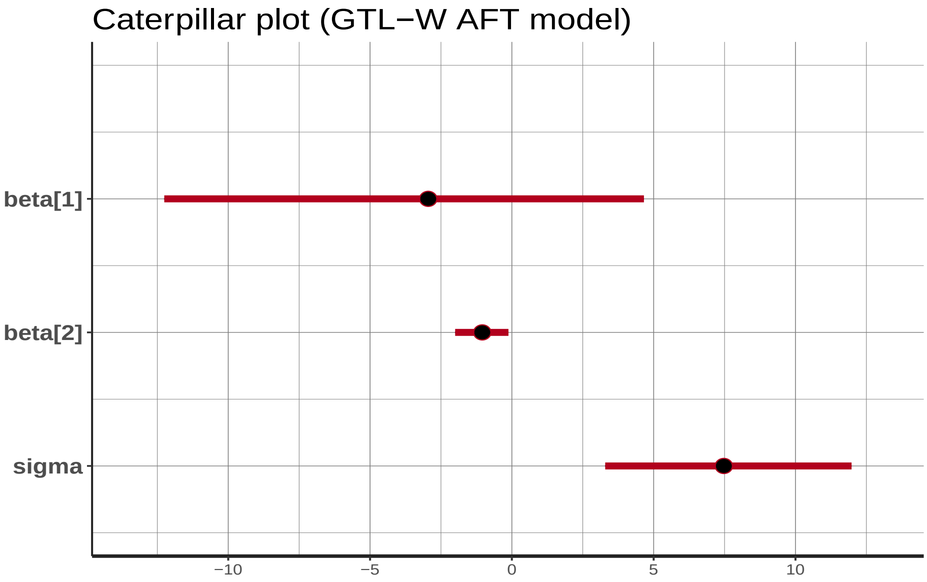

Figure 4: Caterpillar plot of GTL-W AFT model shows that the 95% credible interval of the parameter $\beta[2]$ does not include zero value that means the parameter is statistically significant

5. Sub-models of GTL-W AFT model

If $b = 1$ then GTL-W $(a, b, 1/\sigma, \texttt{exp}(\mathbf{x}'\beta))$ AFT model reduces to TL-W $(a, 1/\sigma, \texttt{exp}(\mathbf{x}'\beta))$ AFT model,

$$F(t|a, \sigma, \beta, \mathbf{x}) = \{G(t|\sigma, \beta, \mathbf{x})(2 - G(t|\sigma, \beta, \mathbf{x}))\}^a \tag{21}$$

where,

$$G(t|\sigma, \beta, \mathbf{x}) = 1 - \texttt{exp}\Big(- \Big(\frac{t}{\texttt{exp}(\mathbf{x}'\beta)}\Big)^{1/\sigma}\Big)$$

is the cdf of baseline Weibull$(1/\sigma, \texttt{exp}(\mathbf{x}'\beta))$ AFT model. So TL-W AFT model and the baseline Weibull AFT model are analyzed and compared with GTL-W AFT model using Bayesian approach.

5.1. Bayesian analysis of sub-models of GTL-W AFT model

Using the same data and same prior distributions the sub-model TL-W AFT and the baseline Weibull AFT model are fitted with little manipulations of Stan code. Necessary results and graphs are shown for comparison.

Fitting TL-W AFT model with Stan and summarizing output:
Stanfit object `M2` (say) is created for TL-W AFT model. The Stan codes for TL-W AFT model are given in the appendix and the whole code block is saved as `stancode_gtlw`.

Summary results are reported in Table 3. Posterior predictive density plot are shown in Figure 5. `LOOIC` and `WAIC` are reported in Table 5.

Table 3: Summary results of fitted TL-W AFT model

Parameter	mean	se_mean	sd	2.5%	50%	97.5%	n_ eff	$\hat{R}$
beta[1]	3.846	0.032	1.283	2.984	3.916	6.193	1604	1.002
beta[2]	-1.132	0.009	0.478	-1.444	-1.124	-0.225	2976	1.001
shape1	15.365	0.237	9.852	8.401	13.089	39.597	1729	1.001
sigma	4.097	0.022	0.941	3.453	4.052	6.048	1752	1.002

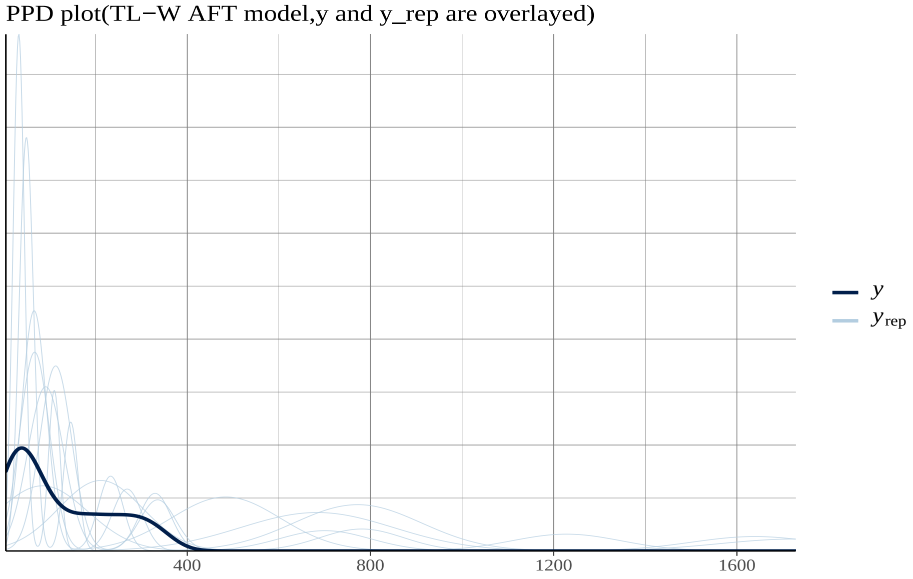

Figure 5: The plot shows that the posterior predictive density of TL-W AFT model fits the data well

Fitting Weibull AFT model and summarizing output:
Stanfit object `M3` is created for Weibull AFT model. The Stan codes for TL-W AFT model are given in the appendix and the whole code block is saved as `stancode_weibull`. Summary results are reported in Table 4. Posterior predictive density plot are shown in Figure 6. `LOOIC` and `WAIC` are reported in Table 5.

Table 4: Summary results of fitted Weibull AFT model

Parameter	mean	se_mean	sd	2.5%	50%	97.5%	n_ eff	$\hat{R}$
beta[1]	6.257	0.008	0.468	5.431	6.221	7.277	3205	1.001
beta[2]	-1.318	0.009	0.510	-2.401	-1.284	-0.406	3304	1.001
sigma	1.274	0.002	0.149	1.021	1.259	1.604	4041	1.000

6. Model comparison

Selecting the best model from among the several competitive models is always crucial in Bayesian statistics and in classical statistics as well. Two information criteria Leave-one-out cross validation (`LOO`) and Widely Applicable or Watanabe Akaike Information Criterion (`WAIC`) (Vehtari, Gelman, and Gabry 2017; McElreath 2015; Gelman *et al.* 2013) are usually used to compare the Bayesian fitted models. Pointwise log-likelihoods

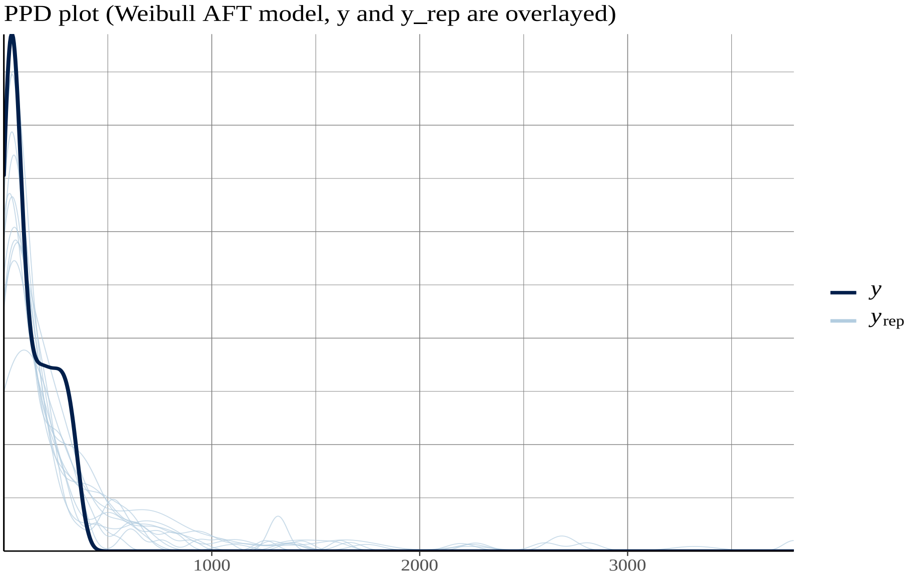

Figure 6: Posterior predictive density (PPD) plot of Weibull AFT model shows that the model is not compatible well with the current data

are calculated in the generated quantities block of `Stan` program and afterwards '`loo`-package' (Vehtari, Gelman, Gabry, and Yao 2018) extracts and uses these quantities to obtain numerical measures `LOOIC` (`LOO` information criterion) or `WAIC` for model comparison and they are presented in the Table 5. A better fitted model has less `LOOIC` or `WAIC` than the others. On the basis of `LOOIC` and `WAIC`, it is evident that the AFT models TL-W and GTL-W are better fitted models than the baseline Weibull AFT model; TL-W and GTL-W AFT models are almost indistinguishable and might be equally acceptable for analyzing the tumour stage data. As generalizations, GTL-W has 4 parameters, TL-W model has 3 parameters whereas Weibull distribution has 2 parameters. The well-known Occam's principle of parsimony is, " If two models describe the observations equally well, we should choose the simpler one." (Lesaffre and Lawson 2012). Accordingly, TL-W AFT model is the most preferred model.

Table 5: LOOIC and WAIC of the fitted model

Model	LOOIC	WAIC
Weibull	661.7	661.7
TL-W	655.3	655.3
GTL-W	655.2	655.1

7. Conclusion

The generalized Topp-Leone-Weibull (GTL-W), Topp-Leone-Weibull (TL-W) and Weibull survival models are fitted as accelerated failure time models under Bayesian framework to the tumour data. For all the models, the coefficient of tumour stage is statistically significant. Comparing posterior predictive density plots, `LOOIC`, `WAIC` and the number of parameters of the models, it can be concluded that the TL-W AFT model is the best suited model for fitting the censored tumour data. The death is accelerated as the acceleration factor is less than one for patients having stage IV tumour, that is, survival time is shorter for the patients with stage IV tumour than that of the patients with stage III tumour (Figure 7).

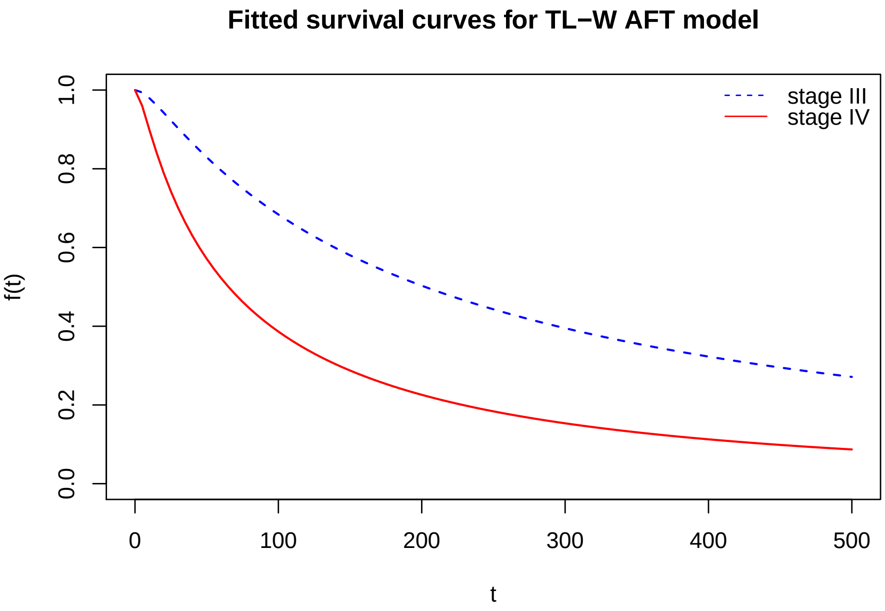

Figure 7: Survival curve for patients with stage III tumour is higher than that of the patients with stage IV tumour

Appendix

A1. Derivation of equation (2)

$$F(x|a,\xi) = G(x)^a[2 - G(x)]^a = \{G(x)[2 - G(x)]\}^a$$
$$f(x|a,\xi) = \frac{d}{dx}F(x|a,\xi) = \frac{d}{dx}\{2G(x) - G(x)^2\}^a$$
$$f(x|a,\xi) = a\{G(x)[2 - G(x)]\}^{a-1}\frac{d}{dx}\{2G(x) - G(x)^2\}$$
$$f(x|a,\xi) = a\{G(x)[2 - G(x)]\}^{a-1}[2g(x) - 2G(x)g(x)]$$
$$f(x|a,\xi) = 2ag(x)[1 - G(x)]G(x)^{a-1}[2 - G(x)]^{a-1}$$

A2. Derivation of equation (6)

$$F(x) = F(x|a,b,\xi) = u$$
$$\{G(x)^b[2 - G(x)^b]\}^a = u$$
$$2G(x)^b - [G(x)^b]^2 = u^{1/a}$$
$$1 - 2G(x)^b + [G(x)^b]^2 = 1 - u^{1/a}$$
$$[1 - G(x)^b]^2 = 1 - u^{1/a}$$
$$G(x) = \left\{1 - \sqrt{(1 - u^{1/a})}\right\}^{1/b}$$

A3. Stan code for GTL-W AFT model

```
library(rstan)
stancode_gtlw = "
functions{
// defines the log survival
vector log_S (vector t,real shape1,real shape2,
                       real shape3,vector scale){
vector[num_elements(t)] log_S ;
   for (i in 1:num_elements(t)){
log_S[i] = log(1-(((weibull_cdf(t[i],shape3,
                  scale[i]))^(shape1*shape2))*
                  ((2-(weibull_cdf(t[i],shape3,
                scale[i]))^(shape2))^(shape1))));
}
return log_S;
}
//defines the log hazard
vector log_h (vector t,real shape1,real shape2,
                           real shape3,vector scale){
vector[num_elements(t)] log_h ;
vector[num_elements(t)] ls ;
ls = log_S(t,shape1,shape2,shape3,scale) ;
  for (i in 1:num_elements(t)){
log_h[i] = (log(2)+log(shape1)+log(shape2)+
             weibull_lpdf(t[i]|shape3,scale[i])+
((shape1*shape2)-1)*weibull_lcdf(t[i]|shape3,scale[i])+
   log(1-(weibull_cdf(t[i],shape3,scale[i]))^(shape2))+
        (shape1-1)*(log(2-(weibull_cdf(t[i],shape3,
               scale[i]))^(shape2))))- ls[i];
}
return log_h;
}
//defines the log-likelihood for right censored data
real surv_gtlw_lpdf(vector t,vector d,real shape1,
            real shape2,real shape3,vector scale){
vector[num_elements(t)] log_lik;
real prob;
log_lik = d .* log_h(t,shape1,shape2,shape3,scale)+
               log_S(t,shape1,shape2,shape3,scale);
prob = sum(log_lik);
return prob;
  }
}
//data block
data{
int N;            // number of observations
vector <lower=0> [N] y;// observed times
vector <lower=0,upper=1> [N] event;//censoring(1=obs.,
                                   //       0=cens.)
int M;      // number of covariates
matrix[N,M] x;//model matrix (N rows, M columns)
```

```
}
//parameters block
parameters{
vector [M] beta;//coef.in the linear predictor
real<lower=0> shape1;// shape parameter
real<lower=0> shape2;// shape parameter
real<lower=0> sigma;//scale parameter sigma=1/shape3
}
// transformed parameters block
transformed parameters{
vector[N] linpred;
vector[N] mu;
linpred = x*beta; //linear predictor
for (i in 1:N){
mu[i] = exp(linpred[i]);
  }
}
// model block
model{
shape1 ~ student_t(5,0,10) T[0, ];//prior for shape1
shape2 ~ student_t(5,0,10) T[0, ];//prior for shape2
sigma ~ student_t(2,0,10) T[0, ];//prior for sigma
beta ~ normal(0,10);//prior for reg. coefficients
y ~ surv_gtlw(event,shape1,shape2,1/sigma,mu);
                            //model for the data
}
// generated quantities block
generated quantities{
          vector[N] y_rep;//posterior predictive value
          vector[N] log_lik;//log-likelihood
                  { for(n in 1:N){
log_lik[n] = (((log(2)+log(shape1)+log(shape2)+
            weibull_lpdf(y[n]|1/sigma,exp(x[n,]*beta))+
         ((shape1*shape2)-1)*weibull_lcdf(y[n]|1/sigma,
      exp(x[n,]*beta))+log(1-(weibull_cdf(y[n],1/sigma,
            exp(x[n,]*beta)))^(shape2))+(shape1-1)*
     (log(2-(weibull_cdf(y[n],1/sigma,exp(x[n,]*beta)))^
        (shape2))))-(log(1-(((weibull_cdf(y[n],1/sigma,
               exp(x[n,]*beta)))^(shape1*shape2))*
       ((2-(weibull_cdf(y[n],1/sigma, exp(x[n,]*beta)))^
                 (shape2))^(shape1))))))*event[n])+
    (log(1-(((weibull_cdf(y[n],1/sigma,exp(x[n,]*beta)))^
         (shape1*shape2))*((2-(weibull_cdf(y[n],1/sigma,
           exp(x[n,]*beta)))^(shape2))^(shape1)))));}
}
              {real u;
                u=uniform_rng(0,1);
                  for (n in 1:N){
  y_rep[n] = (exp(x[n,]*beta))*
             ((-log(1-(1-(sqrt(1-u^(1/shape1))))^
                   (1/shape2)))^(sigma));}
}
```

```
}
"
```

The whole code block is saved as `stancode_gtlw` that is used afterwards.

A3.1. Fitting GTL-W AFT model with Stan

```
#selecting initial values
require(survival)
#regression coefficient of log(y) as initial values
beta1 = solve(crossprod(x),crossprod(x,log(y)))
#convert matrix to a vector
beta1 = c(beta1)
#beta1 is used as initial values for 4 chains

#fitting with stan
M1 <- stan(model_code = stancode_gtlw,data=dat1,
          init=list(list(beta=beta1),list(beta=beta1),
          list(beta=beta1),list(beta=beta1)),iter=4000,
           chains=4,control=list(adapt_delta=0.9995))
```

A3.2. Summarizing output of fitted GTL-W AFT

```
print(M1,c("beta","shape1","shape2","sigma"),
            probs=c(0.025,0.05,0.975),digits=3)
require(bayesplot)
require(loo)

stan_trace(M1, pars=c("beta","sigma"))+
          ggtitle("Trace plot (GTL-W AFT model)")
#posterior predictive check
# posterior predictive value y_rep
y_rep <- as.matrix(M1, pars="y_rep")
ppc_dens_overlay(y,y_rep[50:70,])+grid_lines()+
            ggtitle("PPD plot (GTL-W AFT model,
                    y and y_rep are overlayed)")

#Caterpillar plot for showing credible intervals
stan_plot(M1,pars=c("beta","sigma"),ci_level=.95)+
                          grid_lines()+
         ggtitle("Caterpillar plot (GTL-W AFT model)")

#calculating looic and waic using loo package
log_lik_1 <- extract_log_lik(M1,parameter_name="log_lik",
                             merge_chains = TRUE)
loo_1 <- loo(log_lik_1,r_eff = NULL,save_psis = FALSE)
print(loo_1)
waic1 <- waic(log_lik_1)
print(waic1)
```

A4. Stan code for TL-W AFT model

```
stancode_tlw = "
functions{
// defines the log survival
vector log_S (vector t, real shape1, real shape2,
                                   vector scale){
vector[num_elements(t)] log_S ;
 for (i in 1:num_elements(t)){
log_S[i] = log(1-(((weibull_cdf(t[i],shape2,scale[i]))^
        (shape1))*((2-
        (weibull_cdf(t[i],shape2,scale[i])))^(shape1))));
}
return log_S;
}
//defines the log hazard
vector log_h (vector t,real shape1,real shape2,
                                  vector scale){
vector[num_elements(t)] log_h ;
vector[num_elements(t)] ls ;
ls = log_S(t,shape1,shape2, scale) ;
  for (i in 1:num_elements(t)){
log_h[i] = (log(2)+log(shape1)+weibull_lpdf(t[i]|
        shape2,scale[i])+
        weibull_lccdf(t[i]|shape2,scale[i])+(shape1-1)*
        weibull_lcdf(t[i]|shape2,scale[i])+(shape1-1)*
        (log(2-weibull_cdf(t[i],shape2,scale[i]))))-
        ls[i];
}
return log_h;
}
//defines the log-likelihood for right censored data
real surv_tlw_lpdf(vector t,vector d,
            real shape1,real shape2, vector scale){
vector[num_elements(t)] log_lik;
real prob;
log_lik = d .* log_h(t, shape1, shape2, scale) +
               log_S(t, shape1, shape2, scale);
prob = sum(log_lik);
return prob;
  }
}
//data block
data{
int N;                   // number of observations
vector <lower=0> [N] y;  // observed times
vector <lower=0,upper=1> [N] event;//censoring(1=obs.,
                                    //       0=cens.)
int M;      // number of covariates
matrix[N, M] x;//model matrix (N rows, M columns)
}
//parameters block
```

```
parameters{
vector [M] beta;// coeff.in the linear predictor
real<lower=0> shape1; // shape parameter
real<lower=0> sigma;//scale parameter sigma=1/shape2
}
// transformed parameters block
transformed parameters{
vector[N] linpred;
vector[N] mu;
linpred = x*beta;//linear predictor
for (i in 1:N){
mu[i] = exp(linpred[i]);
  }
}
// model block
model{
shape1 ~ student_t(5,0,10) T[0, ];//prior for shape1
sigma ~ student_t(2,0,10) T[0, ];//prior for sigma
beta ~ normal(0,10);//prior for reg. coefficients
y ~ surv_tlw(event, shape1, 1/sigma, mu);
                              //model for the data
}
// generated quantities block
generated quantities{
          vector[N] y_rep;//posterior predictive value
          vector[N] log_lik;//log-likelihood
              { for(n in 1:N){
log_lik[n] = (((log(2)+ log(shape1)+
          weibull_lpdf(y[n]|1/sigma, exp(x[n,]*beta))+
          weibull_lccdf(y[n]|1/sigma,exp(x[n,]*beta))+
          (shape1-1)*weibull_lcdf(y[n]|1/sigma,exp(x[n,]*
          beta))+ (shape1-1)*(log(2-weibull_cdf(y[n],
          1/sigma,exp(x[n,]*beta)))))-(log(1-
          (((weibull_cdf(y[n],1/sigma,exp(x[n,]*beta)))^
          (shape1))*((2-(weibull_cdf(y[n],1/sigma,
          exp(x[n,]*beta))))^(shape1))))))*event[n])+
          (log(1-(((weibull_cdf(y[n],1/sigma,exp(x[n,]*
          beta)))^(shape1))*((2-(weibull_cdf(y[n],1/sigma,
          exp(x[n,]*beta))))^(shape1)))));}
}
                {real u;
                 u=uniform_rng(0,1);
                   for (n in 1:N){
  y_rep[n] = (exp(x[n,]*beta))*
              ((-log(sqrt(1-u^(1/shape1))))^(sigma));}
}
}
"
```

The whole code block is saved as `stancode_tlw`.

A4.1. Fitting TL-W AFT model with Stan and summarizing output

```
#selecting initial values
require(survival)
#regression coefficient of log(y) as initial values
beta1 = solve(crossprod(x),crossprod(x,log(y)))
beta1 = c(beta1) #convert matrix to a vector
#beta1 is used as initial values for 4 chains

# fitting with stan
M2 <- stan(model_code=stancode_tlw,data=dat1,
      init=list(list(beta=beta1),list(beta=beta1),
      list(beta=beta1),list(beta=beta1)),iter=4000,
      chains=4,control=list(adapt_delta=0.9995))
print(M2, c("beta", "shape1", "sigma"),
             probs=c(0.25,0.5,0.975),digits=3)

require(bayesplot)
require(loo)

#posterior predictive check
#posterior predictive value y_rep
y_rep <- as.matrix(M2, pars="y_rep")
ppc_dens_overlay(y,y_rep[550:560,])+ grid_lines()+
  ggtitle("PPD plot(TL-W AFT model,
           y and y_rep are overlayed)")

#calculating looic and waic using loo package
log_lik_1 <- extract_log_lik(M2,parameter_name="log_lik",
             merge_chains = TRUE)
loo_1 <- loo(log_lik_1, r_eff = NULL,save_psis = FALSE)
print(loo_1)
waic1 <- waic(log_lik_1)
print(waic1)
```

A5. Stan code for Weibull AFT model

```
stancode_weibull = "
functions{
// defines the log survival
vector log_S (vector t,real shape,vector scale){
vector[num_elements(t)] log_S;
  for (i in 1:num_elements(t)){
log_S[i] = weibull_lccdf(t[i]|shape,scale[i]);
}
return log_S;
}
//defines the log hazard
vector log_h (vector t,real shape,vector scale){
vector[num_elements(t)] log_h ;
vector[num_elements(t)] ls ;
```

```
ls = log_S(t,shape, scale) ;
  for (i in 1:num_elements(t)){
log_h[i] = (weibull_lpdf(t[i]|shape,scale[i]))-ls[i];
  }
return log_h;
}
//defines the log-likelihood for right censored data
real surv_weibull_lpdf(vector t,vector d,real shape,
                      vector scale){
vector[num_elements(t)] log_lik;
real prob;
log_lik = d .* log_h(t,shape,scale)+
               log_S(t,shape,scale);
prob = sum(log_lik);
return prob;
  }
}
//data block
data{
int N;                // number of observations
vector <lower=0> [N] y; // observed times
vector <lower=0,upper=1> [N] event;//censoring (1=obs.,
                                   //          0=cens.)
int M;             // number of covariates
matrix[N,M] x;//model matrix (N rows, M columns)
}
//parameters block
parameters{
vector [M] beta;  //coeff. in the linear predictor
real<lower=0> sigma;//scale parameter sigma=1/shape
}
// transformed parameters block
transformed parameters{
vector[N] linpred;
vector[N] mu;
linpred = x*beta;//linear predictor
  for (i in 1:N){
mu[i] = exp(linpred[i]);
  }
}
// model block
model{
sigma ~ student_t(2,0,10) T[0,];//prior for sigma
beta ~ normal(0,10); //prior for reg. coefficients
y ~ surv_weibull(event, 1/sigma, mu);
                             //model for the data
}
// generated quantities block
generated quantities{
            vector[N] y_rep;//posterior predictive value
            vector[N] log_lik;//log-likelihood
                       {  for(n in 1:N){
```

```
log_lik[n] = (((weibull_lpdf(y[n]|1/sigma,exp(x[n,]*beta)))-
             (weibull_lccdf(y[n]|1/sigma,exp(x[n,]*beta))))*
                 event[n])+(weibull_lccdf(y[n]|1/sigma,
                                exp(x[n,]*beta)));}
}
                     {  for (n in 1:N){
y_rep[n]= weibull_rng( 1/sigma, exp((x[n,]*beta)));}
                     }
}
"
```

The whole code block is saved as `stancode_weibull`.

A5.1. Fitting Weibull AFT model with Stan and summarizing output

```
# fitting with stan
M3 <- stan(model_code=stancode_weibull,data=dat1,
                  chains=4,iter=4000)
print(M3,c("beta","sigma"),digits=3,probs=c(0.025,0.50,0.975))

require(bayesplot)
require(loo)

#posterior predictive check
#posterior predictive value y_rep
y_rep <- as.matrix(M3,pars="y_rep")
ppc_dens_overlay(y,y_rep[110:120,])+grid_lines()+
        ggtitle("PPD plot (Weibull AFT model,
                y and y_rep are overlayed)")

#calculating looic and waic using loo package
log_lik_1 <- extract_log_lik(M3,parameter_name="log_lik",
                             merge_chains = TRUE)
loo_1 <- loo(log_lik_1,r_ef=NULL,save_psis=FALSE)
print(loo_1)
waic1 <- waic(log_lik_1)
print(waic1)
```

References

Al-Shomrani A, Arif O, Shawky A, Hanif S, Shahbaz MQ (2016). “Topp–Leone Family of Distributions: Some Properties and Application.” *Pakistan Journal of Statistics and Operation Research*, **12**(3), 443–451.

Armitage P, Berry G, Matthews JNS (2002). *Statistical Methods in Medical Research*. John Wiley & Sons.

Aryal GR, Ortega EM, Hamedani G, Yousof HM (2017). “The Topp-Leone Generated Weibull Distribution: Regression Model, Characterizations and Applications.” *International Journal of Statistics and Probability*, **6**(1), 126–141.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017). "Stan: A Probabilistic Programming Language." *Journal of Statistical Software*, **76**(1).

Collett D (2015). *Modelling Survival Data in Medical Research.* Chapman and Hall/CRC.

Cordeiro GM, de Castro M (2011). "A New Family of Generalized Distributions." *Journal of Statistical Computation and Simulation*, **81**(7), 883–898.

Gabry J, Mahr T (2017). "bayesplot: Plotting for Bayesian Models." *R package version*, **1**.

Gelman A, Rubin DB (1992). "Using Multiple Sequences." *Statistical Science*, **7**(4), 457–472.

Gelman A, Stern HS, Carlin JB, Dunson DB, Vehtari A, Rubin DB (2013). *Bayesian Data Analysis.* Chapman and Hall/CRC.

Gelman A, *et al.* (2006). "Prior Distributions for Variance Parameters in Hierarchical Models (Comment on Article by Browne and Draper)." *Bayesian analysis*, **1**(3), 515–534.

Hamra G, MacLehose R, Richardson D (2013). "Markov Chain Monte Carlo: An Introduction for Epidemiologists." *International Journal of Epidemiology*, **42**(2), 627–634.

Hoffman MD, 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.

Ibrahim JG, Chen MH, Sinha D (2001). *Bayesian Survival Analysis.* Springer Science & Business Media.

Kalbfleisch JD, Prentice RL (2002). *The Statistical Analysis of Failure Time Data*, volume 360. John Wiley & Sons.

Korner-Nievergelt F, Roth T, Von Felten S, Guélat J, Almasi B, Korner-Nievergelt P (2015). *Bayesian Data Analysis in Ecology Using Linear Models with R, BUGS, and Stan.* Academic Press.

Kruschke J (2015). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan.* Academic Press.

Lawless JF (2003). *Statistical Models and Methods for Lifetime Data*, volume 362. John Wiley & Sons.

Lee ET, Wang J (2013). *Statistical Methods for Survival Data Analysis.* John Wiley & Sons.

Lesaffre E, Lawson AB (2012). *Bayesian Biostatistics.* John Wiley & Sons.

Li J, Ma S (2013). *Survival Analysis in Medicine and Genetics.* CRC Press.

Mahdavi A (2017). "Generalized Topp-Leone Family of Distributions." *Journal of Biostatistics and Epidemiology*, **3**(2), 65–75.

McElreath R (2015). *Statistical Rethinking: A Bayesian Course with Examples in R and Stan.* Chapman and Hall/CRC.

McKelvey EM, Gottlieb JA, Wilson HE, Haut A, Talley RW, Stephens R, Lane M, Gamble JF, Jones SE, Grozea PN, *et al.* (1976). "Hydroxyldaunomycin (Adriamycin) Combination Chemotherapy in Malignant Lymphoma." *Cancer*, **38**(4), 1484–1493.

Nadarajah S, Kotz S (2003). "Moments of Some J-shaped Distributions." *Journal of Applied Statistics*, **30**(3), 311–317.

Ntzoufras I (2009). *Bayesian Modeling Using WinBUGS*, volume 698. John Wiley & Sons.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rezaei S, Sadr BB, Alizadeh M, Nadarajah S (2017). "Topp–Leone Generated Family of Distributions: Properties and Applications." *Communications in Statistics-Theory and Methods*, **46**(6), 2893–2909.

RStudio Team (2015). *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA. URL http://www.rstudio.com/.

Sangsanit Y, Bodhisuwan W (2016). "The Topp-Leone Generator of Distributions: Properties and Inferences." *Songklanakarin Journal of Science & Technology*, **38**(5).

Stan Development Team (2017). "Stan Modeling Language Users Guide and Reference Manual." Version 2.16.0, URL http://mc-stan.org.

Stan Development Team (2018). "RStan: the R Interface to Stan." R package version 2.17.3, URL http://mc-stan.org/.

Topp CW, Leone FC (1955). "A Family of J-shaped Frequency Functions." *Journal of the American Statistical Association*, **50**(269), 209–219.

Turkman MAA, Paulino CD, Müller P (2019). *Computational Bayesian Statistics: An Introduction*, volume 11. Cambridge University Press.

Vehtari A, Gelman A, Gabry J (2017). "Practical Bayesian Model Evaluation Using Leave-one-out Cross-validation and WAIC." *Statistics and Computing*, **27**(5), 1413–1432.

Vehtari A, Gelman A, Gabry J, Yao Y (2018). "loo: Efficient Leave-one-out Cross-validation and WAIC for Bayesian Models. R package version 2.0. 0."

Vidakovic B (2011). *Statistics for Bioengineering Sciences: with MATLAB and WinBUGS Support*. Springer Science & Business Media.

Wang X, Ryan YY, Faraway JJ (2018). *Bayesian Regression Modeling with INLA*. Chapman and Hall/CRC.

Wiper MP, Girón FJ, Pewsey A (2008). "Objective Bayesian Inference for the Half-Normal and Half-t Distributions." *Communications in Statistics—Theory and Methods*, **37**(20), 3165–3185.

Affiliation:

Md. Ashraf-Ul-Alam
Department of Statistics and Operations Research
Aligarh Muslim University
202002 Aligarh, India
E-mail: mdashrafulal@gmail.com

Athar Ali Khan
Department of Statistics and Operations Research
Aligarh Muslim University
202002 Aligarh, India
E-mail: atharkhan1962@gmail.com

Austrian Journal of Statistics http://www.ajs.or.at/
published by the Austrian Society of Statistics http://www.osg.or.at/
Volume 50 *Submitted:* 2020-05-13
July 2021 *Accepted:* 2020-10-13