

Bayesian Survival Analysis of Topp-Leone Generalized Family with Stan

Mohammed H. AbuJarad*, Athar Ali Khan

Department of Statistics and Operations Research, AMU, Aligarh, India

Abstract In this article, the discussion has been carried out on the generalization of three distribution by means of exponential, exponentiated exponential and exponentiated extension. We set up three and four parameters life model called the Topp-Leone exponential distribution, Topp-Leone exponentiated exponential distribution and Topp-Leone exponentiated extension distribution. We give extensive consequence of the, survival function and hazard rate function. To fit this model as survival model and hazard rate function we adopted to use Bayesian approach. A real survival data set is used to illustrate. application is done by **R** and **Stan** and suitable illustrations are prepared. **R** and **Stan** codes have been given to actualize censoring mechanism via optimization and also simulation tools.

Keywords Topp-Leone exponential, Topp-Leone exponentiated exponential, Topp-Leone exponentiated extension, Posterior, Simulation, RStan, Bayesian Inference, R, HMC

1. Introduction

In the survival, a number of continuous univariate distributions have been widely utilized for demonstrating information in numerous areas, for example, biology, medicine, engineering, public health, epidemiology and economics. In any case, applied areas, for example, lifetime analysis obviously require expanded types of these distributions. In this way, a few classes of distributions have been built by extending common families of continuous distributions. These generalized distributions give greater adaptability by including "at least one" parameters to the standard model. The Topp-Leone distribution was introduced by Topp and Leone in 1955 (Topp and Leone, 1955). Topp-Leone Generalized family of distributions was inferred by Rezaei et al. (2016). The distribution and density function of proposed family is known by

$$F(t) = [G(t)^b \{2 - G(t)^b\}]^\alpha \quad (1.1)$$

$$f(t) = 2abg(t)[G(t)]^{ab-1}\{1 - G(t)^b\}\{2 - G(t)^b\}^{\alpha-1} \quad (1.2)$$

Where $G(t)$ and $g(t)$ are the cdf and pdf, respectively where the $b > 0$ and $\alpha > 0$. This present article is designed as follows; Section 2, we derive three parameter life model called Topp-Leone exponential distribution, the pdf and cdf expansion, in Section 2.1, we derive four

parameter life model called Topp-Leone exponentiated exponential distribution, the pdf and cdf expansion, in Section 2.2, we derive four parameter life model called Topp-Leone exponentiated extension distribution, the pdf and cdf expansion. The fundamental properties of the proposed demonstrate including, Bayesian methodology has been received to fit this model as survival model and hazard rate function. Survival analysis is the name for an accumulation of statistical techniques used to depict and evaluate time to event data. In survival analysis we utilize the term inability to characterize the event of the interest. In this paper, an endeavor has been made to plot how Bayesian methodology continues to fit Topp-Leone exponential model, Topp-Leone exponentiated exponential and Topp-Leone exponential extension for lifetime data using Stan. The tools and techniques used in this paper are in Bayesian environment, which are implemented using rstan package. Stan is a programming language designed to make statistical modeling easier and faster, especially for Bayesian estimation problems, it can do estimate complex models with large numbers of parameters, and can generally do it faster than alternative like JAGS/BUGS. However, Simulation can also be used as an alternative technique. Simulation based on Markov chain Monte Carlo (MCMC) is used when it is not possible to sample θ directly from posterior $p(\theta|y)$. For a wide class of problems, this is the easiest method to get reliable results (Gelman et al, 2014). Gibbs sampling, Hamiltonian Monte Carlo and Metropolis-Hastings algorithm are the MCMC techniques which render difficult computational tasks quite feasible. To make computation easier, software such as R, Stan is a C++ library for Bayesian modeling and inference that primarily

* Corresponding author:

m.jarad@gu.edu.ps (Mohammed H. AbuJarad)

Published online at <http://journal.sapub.org/statistics>

Copyright © 2018 The Author(s). Published by Scientific & Academic Publishing

This work is licensed under the Creative Commons Attribution International

License (CC BY). <http://creativecommons.org/licenses/by/4.0/>

uses the No-U-Turn sampler (NUTS) (Hoffman and Gelman 2012) to obtain posterior simulation given specified model and data, a variant of Hamiltonian Monte Carlo (HMC) are used, Stan can produce high dimensional proposals that are accepted with high probability without having to spend time tuning. Bayesian analysis of proposal appropriation has 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. The Topp-Leone Exponential Distribution

Corderio et al. (2013) utilized the method for adding parameter prompts the exponentiated type of distribution which was considered by Nadarajah and Kotz (2003). In this segment, we infer three parameter Topp-Leone Exponential distribution. To construct the probability density function (pdf) and cumulative distribution function (cdf) of Exponential distribution which are given by (2.3) and (2.4), individually,

$$f(t) = \frac{1}{\lambda} \exp\left(-\frac{t}{\lambda}\right) \quad (2.3)$$

$$F(t) = (1 - \exp\left(-\frac{t}{\lambda}\right)) \quad (2.4)$$

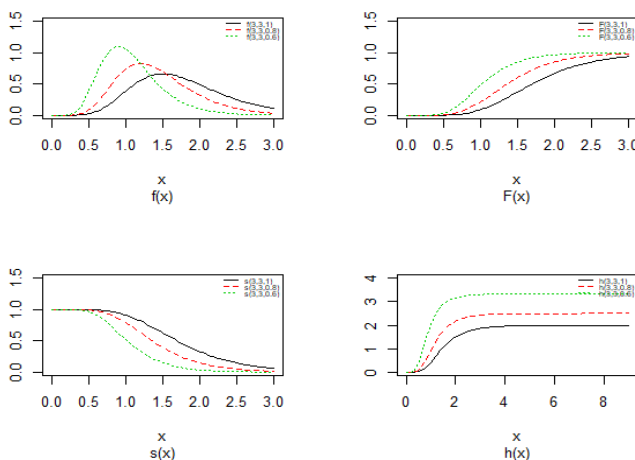


Figure 1. Probability density plots, cdf, survival and hazard curves of Topp-Leone Exponential distribution for different value

By inserting (2.3) and (2.4) into (1.1) and (1.2), we have the pdf, cdf, survival function and hazard function of the proposed model given by respectively, as in Figure (1). Then T follows Topp-Leone Generalized Exponential (TLGE) model (Sangsanit and Bodhisuman, 2016)

$$f(t) = \frac{2\alpha b}{\lambda} \exp\left(-\frac{t}{\lambda}\right) (1 - \exp\left(-\frac{t}{\lambda}\right))^{b\alpha-1} (1 - (1 - \exp\left(-\frac{t}{\lambda}\right))^b) (2 - (1 - \exp\left(-\frac{t}{\lambda}\right))^b)^{\alpha-1} \quad (2.5)$$

$$F(t) = (1 - \exp\left(-\frac{t}{\lambda}\right))^{b\alpha} (2 - (1 - \exp\left(-\frac{t}{\lambda}\right))^b)^{\alpha} \quad (2.6)$$

$$S(t) = 1 - [(1 - \exp\left(-\frac{t}{\lambda}\right))^{b\alpha} (2 - (1 - \exp\left(-\frac{t}{\lambda}\right))^b)^{\alpha}] \quad (2.7)$$

$$h(t) = \frac{f(t)}{S(t)} \quad (2.8)$$

2.1. The Topp-Leone Exponentiated Exponential Distribution

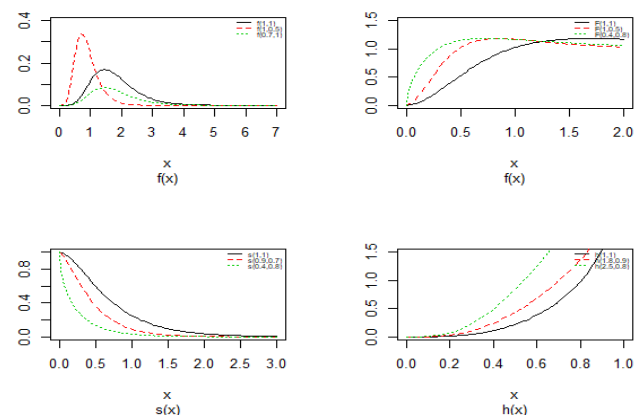


Figure 2. Probability density plots, cdf, survival and hazard curves of Topp-Leone Exponential distribution for different value

Topp-Leone additional two shapes parameter to the two-parameter exponentiated exponential distribution. It is seen that the new four-parameter distribution is exceptionally adaptable. At the point when the pdf, cdf, survival function and hazard function of exponentiated exponential appropriation is $g(t) \sim \exp\exp(\theta, \lambda)$, the outcomes are (2.9), (2.10), (2.11) and (2.12), individually, as in Figure(2)

$$f(t) = \frac{2\alpha b \lambda}{\theta} (1 - \exp\left(-\frac{t}{\theta}\right))^{\alpha-1} \exp\left(-\frac{t}{\theta}\right) [(1 - \exp\left(-\frac{t}{\theta}\right))^{\alpha}]^{\alpha b-1} \times (1 - [(1 - \exp\left(-\frac{t}{\theta}\right))^{\alpha}]^b) (2 - [(1 - \exp\left(-\frac{t}{\theta}\right))^{\alpha}]^b)^{\alpha-1} \quad (2.9)$$

$$F(t) = [((1 - \exp\left(-\frac{t}{\theta}\right))^{\alpha})^b] \{2 - [(1 - \exp\left(-\frac{t}{\theta}\right))^{\alpha}]^b\}^{\alpha} \quad (2.10)$$

$$S(t) = 1 - \{[(1 - \exp\left(-\frac{t}{\theta}\right))^{\alpha}]^b \{2 - [(1 - \exp\left(-\frac{t}{\theta}\right))^{\alpha}]^b\}^{\alpha}\} \quad (2.11)$$

$$h(t) = \frac{f(t)}{S(t)} \quad (2.12)$$

2.2. The Topp-Leone Exponential Extension Distribution

When the pdf, cdf, survival function and hazard function of exponential extension distribution is $g(t) \sim \text{expe}(\theta, \lambda)$, the results are (2.13), (2.14), (2.15) and (2.16), respectively, as in Figure(3)

$$f(t) = \frac{2\alpha b \lambda}{\theta} \left(1 + \frac{t}{\theta}\right)^{\lambda-1} \exp\left(1 - \left(1 + \frac{t}{\theta}\right)^{\lambda}\right) \left[1 - \exp\left(1 - \left(1 + \frac{t}{\theta}\right)^{\lambda}\right)\right]^{\alpha b-1} \times (1 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))) (2 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))^b)^{\alpha-1} \quad (2.13)$$

$$F(t) = [(1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda})) \{2 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))^b\}]^{\alpha} \quad (2.14)$$

$$S(t) = 1 - \{[(1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda})) \{2 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))^b\}]^{\alpha}\} \quad (2.15)$$

$$h(t) = \frac{f(t)}{S(t)} \quad (2.16)$$

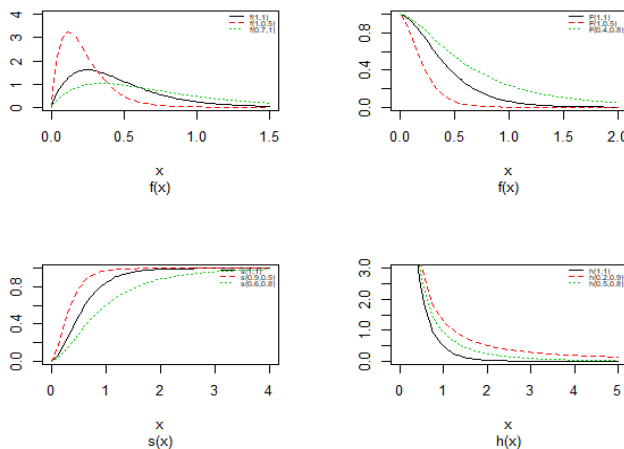


Figure 3. Probability density plots, cdf, survival and hazard curves of Topp-Leone Exponential distribution for different value

3. Bayesian Inference

Gelman et al., (2013) break applied Bayesian modeling into the following three steps:

1. Set up a full probability model for all observable and unobservable quantities. This model should be consistent with existing knowledge of the data being modeled and how it was collected.
2. Calculate the posterior probability of unknown quantities conditioned on observed quantities. The unknowns may include unobservable quantities such as parameters and potentially observable quantities such as predictions for future observations.
3. Evaluate the model fit to the data. This includes evaluating the implications of the posterior.

Typically, this cycle will be repeated until a sufficient fit is achieved in the third step. Stan automates the calculations

involved in the second and third steps (Carpenter et al., 2017).

We have to specify here the most vital in Bayesian inference which are as per the following:

- prior distribution: $p(\theta)$: The parameter θ can set a prior distribution elements that using probability as a means of quantifying uncertainty about θ before taking the data into account.
- Likelihood $p(y|\theta)$: likelihood function for variables are related in full probability model.
- Posterior distribution $p(\theta|y)$: is the joint posterior distribution that expresses uncertainty about parameter θ after considering about the prior and the data, as in equation.

$$P(\theta|y) = p(y|\theta) \times p(\theta) \quad (3.17)$$

4. The Prior Distributions

The Bayesian inference, having the prior distribution, can provide the information concerning an uncertain parameter θ connected through the probability distribution of data. This uncertain parameter is able to help obtain the posterior distribution $p(\theta|y)$. In the case of the Bayesian paradigm, it is very important for prior information to be identified through the value of the specified parameter. The information which are gathered before analyzing the experimental data with using a probability distribution function is referred to as the prior probability distribution (or the prior). In the remain of this paper, The researchers make use of two types of priors: half-Cauchy prior and Normal prior. The simplest types of priors is a conjugate prior which facilitates posterior calculations. In addition, a conjugate prior distribution is intended for an unknown parameter which leads to a posterior distribution for which there is a simple formula for posterior means and variances. (Akhtar and Khan, 2014a) apply the half-Cauchy distribution by scale parameter $\alpha = 25$ while a prior distribution for scale parameter.

Hereinafter we determination talk about the types of prior distribution:

- Half-Cauchy prior.
- Normal prior.

First, the probability density function of half-Cauchy distribution by scale parameter α is specified as a result

$$f(x) = \frac{2\alpha}{\pi(x^2 + \alpha^2)} \quad x > 0, \alpha > 0.$$

Half-Cauchy distribution does not exist for mean and variance, although its mode is equal to 0. The half-Cauchy distribution by scale $\alpha = 25$ is a suggested, default, weakly informative prior distribution used for a scale parameter. On this scale $\alpha = 25$, the density of half-Cauchy is almost flat however not completely (see Figure 4), prior distributions that are not completely flat afford adequate information for

the numerical approximation algorithm to continue to look at the target density; the posterior distribution. The inverse-gamma is often used as a non-informative prior distribution for scale parameter, but; this model creates a trouble for scale parameters close to zero; (Gelman and Hill, 2007) suggest that, the uniform, otherwise if more information is needed, the half-Cauchy is a better option. Consequently, in this paper, the half-Cauchy distribution with scale parameter $\alpha=25$ is used as a weakly informative prior distribution.

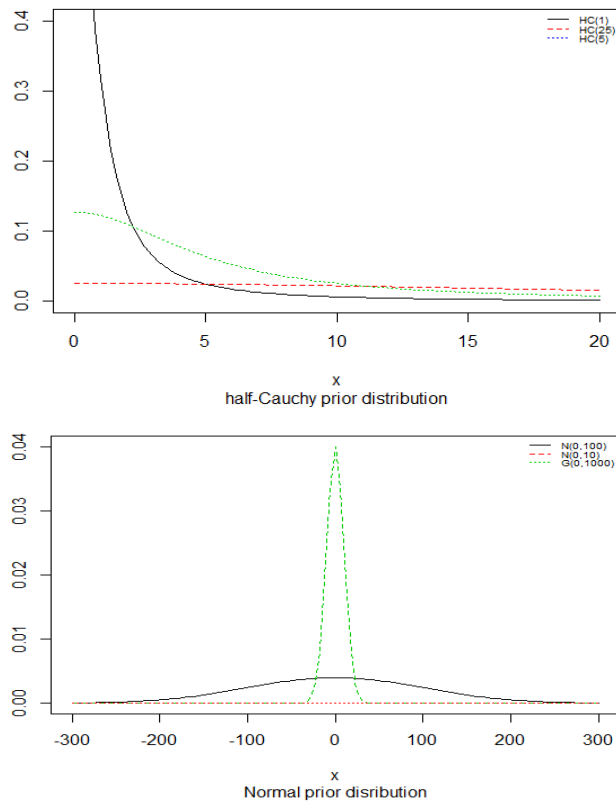


Figure 4

Next, the normal (or Gaussian), every parameters is assigned a weak information Gaussian prior probability distribution. The researcher aims to use the parameters β_i independently in the normal distribution with mean=0 and standard deviation=1000, i.e., $\beta_j \sim N(0,1000)$, for this, we get a flat prior. As of (Figure 4), we see that the large variance indicates a lot of uncertainty about each parameter and hence, a weak informative distribution.

5. Stan Modeling

Stan is an abnormal state dialect written in a C++ library for Bayesian demonstrating and (Carpenter et al., 2017) is another Bayesian programming program for induction that essentially utilizes the No-U-Turn sampler (NUTS) (Hoffman and Gelman 2012) to get posterior simulations given a client indicated model and data. Hamiltonian Monte Carlo (HMC; Radford 2011; Betancourt 2017) is one of the calculations having a place with the general class of MCMC

strategies. Practically speaking, HMC can be very complex, because in addition to the specific computation of possibly complex derivatives, it requires tweaking of a few parameters. Hamiltonian Monte Carlo requires a touch of exertion to program and tune. In more entangled settings, however, HMC to be quicker and more dependable than fundamental Markov chain reproduction, Gibbs sampler and the Metropolis algorithm because they explores the posterior parameter space more efficiently. they do so by pairing each model parameter with a momentum variable, which determines HMC exploration behavior of the target distribution based on the posterior density of the current drawn parameter and hence enable HMC to suppress the random walk behavior in the Metropolis algorithm (Gelman, Carlin, Stern, & Rubin, 2014, p. 300). Consequently, Stan is considerably more efficient than the traditional Bayesian software programs. However, the main function in the rstan package is Stan, which calls the Stan software program to estimate a specified statistical model, rstan provides a very clever system in which most of the adaptation is automatic. Statistical model through a conditional probability function $p(\theta|y,x)$ can be classified by Stan program, where θ is a sequence of modeled unknown values, y is a sequence of modeled known values, and x is a sequence of un-modeled predictors and constants (e.g., sizes, hyperparameters). A Stan program imperatively defines a log probability function over parameters conditioned on specified data and constants. Stan provides full Bayesian inference for continuous-variable models through Markov chain Monte Carlo methods (Metropolis et al., 1953), an adjusted form of Hamiltonian Monte Carlo sampling (Duane et al., 1987; Neal, 1994). Stan can be called from R using the rstan package, and through Python using the pystan package. All interfaces support sampling and optimization-based inference with diagnostics and posterior analysis. rstan and pystan also provide access to log probabilities, parameter transforms, and specialized plotting. Stan programs consist of variable type declarations and statements. Variable types include constrained and unconstrained integer, scalar, vector, and matrix types. Variables are declared in blocks corresponding to the variable use: data, transformed data, parameter, transformed parameter, or generated quantities.

6. Bayesian Analysis of Model

Bayesian analysis is the strategy to acquire the marginal posterior distribution of the specific parameters of interest. On a fundamental level, the course to accomplishing this point is clear; first, we require the joint posterior distribution of every obscure parameter, at that point, we integrate this distribution over the unknowns parameters that are not of prompt enthusiasm to acquire the coveted marginal distribution. Or on the other hand identically, utilizing simulation, we draw samples from the joint posterior distribution, at that point, we take at the parameters of interest and disregard the estimations of the other obscure parameters.

6.1. Topp-Leone Exponential Model

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

$$f(t, b, \lambda, \alpha) = \frac{2\alpha b}{\lambda} \exp\left(\frac{-t}{\lambda}\right) (1 - \exp\left(\frac{-t}{\lambda}\right))^{b\alpha-1} \\ (1 - (1 - \exp\left(\frac{-t}{\lambda}\right))^b) (2 - (1 - \exp\left(\frac{-t}{\lambda}\right))^b)^{\alpha-1}$$

Likewise, the survival function is given by

$$S(t; \theta, \lambda) = 1 - F(t) = 1 - [(1 - \exp\left(\frac{-t}{\lambda}\right))^{b\alpha} (2 - (1 - \exp\left(\frac{-t}{\lambda}\right))^b)^{\alpha-1}]$$

We can express the likelihood function for right censored (similar to our case the data are right censored) as

$$L = \prod_{i=0}^n Pr(t_i, \delta_i) = \prod_{i=0}^n [f(t_i)]^{\delta_i} [S(t_i)]^{1-\delta_i}$$

where δ_i is a pointer variable which takes esteem 0 if perception is censored and 1 if perception is uncensored. In this manner, the likelihood function is given by

$$L = \prod_{i=0}^n \left[\frac{2\alpha b}{\lambda} \exp\left(\frac{-t}{\lambda}\right) (1 - \exp\left(\frac{-t}{\lambda}\right))^{b\alpha-1} (1 - (1 - \exp\left(\frac{-t}{\lambda}\right))^b) \right. \\ \left. (2 - (1 - \exp\left(\frac{-t}{\lambda}\right))^b)^{\alpha-1} \right]^{\delta_i} \\ \times [1 - ((1 - \exp\left(\frac{-t}{\lambda}\right))^{b\alpha} (2 - (1 - \exp\left(\frac{-t}{\lambda}\right))^b)^{\alpha-1})]^{1-\delta_i}. \quad (6.18)$$

As a result, the posterior distribution of our beliefs about the underlying bias is derived in the usual way by applying Bayes rule (Statisticat LLC 2015), the joint posterior density is given by (AbuJarad and Khan 2018). Here the functions involve three parameters:

$$p(b, \beta, \alpha | t, X) \propto L(t | X, b, \beta) \times p(\beta) \times p(b) \times p(\alpha) \\ \propto \prod_{i=0}^n \left[\frac{2\alpha b}{e^{x\beta}} \exp\left(\frac{-t}{e^{x\beta}}\right) (1 - \exp\left(\frac{-t}{e^{x\beta}}\right))^{b\alpha-1} \times \right. \\ \left. (1 - (1 - \exp\left(\frac{-t}{e^{x\beta}}\right))^b) (2 - (1 - \exp\left(\frac{-t}{e^{x\beta}}\right))^b)^{\alpha-1} \right]^{\delta_i} \\ \times [1 - ((1 - \exp\left(\frac{-t}{e^{x\beta}}\right))^{b\alpha} (2 - (1 - \exp\left(\frac{-t}{e^{x\beta}}\right))^b)^{\alpha-1})]^{1-\delta_i} \\ \times \prod_{j=0}^J \frac{1}{\sqrt{2\pi \times 10^3}} \exp\left(-\frac{1}{2} \frac{\beta_j^2}{10^3}\right) \times \frac{2 \times 25}{\pi(b^2 + 25^2)} \times \frac{2 \times 25}{\pi(\alpha^2 + 25^2)}. \quad (6.19)$$

To complete Bayesian inference in the Topp-Leone model, we ought to decide an prior distribution for b, α and β 's. We talked about the issue related with determining prior distributions in section 4, however, for effortlessness now, we expect that the prior distribution for b and α is half-Cauchy on the interval $[0, 5]$ and for β is Normal with $[0, 5]$. Rudimentary utilization of Bayes control as showed in (3.17), connected to (6.18), at that point gives the posterior

density for b, α and β as equation (6.19). Result for this marginal posterior distribution get high-dimensional integral over every single model parameters β_j, α and b .

To unravel this integral, we utilize the approximated utilizing Markov Chain Monte Carlo methods techniques. be that as it may, because of the accessibility of computer software package like rstan, this required model can without much of a stretch be fitted in Bayesian paradigm utilizing Stan in addition to MCMC strategies.

6.2. Topp-Leone Exponentiated Exponential Model

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

$$f(t, b, \alpha, \lambda, \theta) = \frac{2\alpha b \lambda}{\theta} (1 - \exp\left(\frac{-t}{\theta}\right))^{\alpha-1} \exp\left(\frac{-t}{\theta}\right) \\ \times [(1 - \exp\left(\frac{-t}{\theta}\right))^{\alpha}]^{\alpha b-1} (1 - [(1 - \exp\left(\frac{-t}{\theta}\right))^{\lambda}]^b) \\ \times (2 - [(1 - \exp\left(\frac{-t}{\theta}\right))^{\lambda}]^b)^{\alpha-1}$$

Also, the survival function is given by

$$S(t, b, \alpha, \lambda, \theta) = 1 - F(t) = 1 - \{[(1 - \exp\left(\frac{-t}{\theta}\right))^{\lambda}]^{\alpha} \} \\ \times \{2 - [(1 - \exp\left(\frac{-t}{\theta}\right))^{\lambda}]^b\}^{\alpha}$$

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

$$L = \prod_{i=0}^n Pr(t_i, \delta_i) = \prod_{i=0}^n [f(t_i)]^{\delta_i} [S(t_i)]^{1-\delta_i}$$

where δ_i 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

$$L = \prod_{i=0}^n \left[\frac{2\alpha b \lambda}{\theta} (1 - \exp\left(\frac{-t}{\theta}\right))^{\alpha-1} \exp\left(\frac{-t}{\theta}\right) [(1 - \exp\left(\frac{-t}{\theta}\right))^{\alpha}]^{\alpha b-1} \right. \\ \times [1 - ((1 - \exp\left(\frac{-t}{\theta}\right))^{\lambda})^b] (2 - [(1 - \exp\left(\frac{-t}{\theta}\right))^{\lambda})^b]^{\alpha-1} \left. \right]^{\delta_i} \\ \times [1 - \{[(1 - \exp\left(\frac{-t}{\theta}\right))^{\lambda}\} \{2 - [(1 - \exp\left(\frac{-t}{\theta}\right))^{\lambda}\}^b\}^{\alpha}\}]^{1-\delta_i}. \quad (6.20)$$

Thus, the joint posterior density is given by

$$p(b, \beta, \alpha | t, X) \propto L(t | X, b, \beta) \times p(\beta) \times p(b) \times p(\alpha) \\ \propto \prod_{i=0}^n \left[\frac{2\alpha b \lambda}{e^{x\beta}} (1 - \exp\left(\frac{-t}{e^{x\beta}}\right))^{\alpha-1} \exp\left(\frac{-t}{e^{x\beta}}\right) \right. \\ \times [(1 - \exp\left(\frac{-t}{e^{x\beta}}\right))^{\alpha}]^{\alpha b-1} (1 - [(1 - \exp\left(\frac{-t}{e^{x\beta}}\right))^{\lambda}]^b) \left. \right]^{\delta_i}$$

$$\begin{aligned}
& \times \{2 - [(1 - \exp(-\frac{t}{e^{x\beta}}))^{\lambda}]^b\}^{\alpha-1} \}^{\delta_i} \\
& \times [1 - \{[(1 - \exp(-\frac{t}{e^{x\beta}}))^{\lambda}]^b\}^{\alpha}\}^{\delta_i}]^{1-\delta_i} \\
& \times \prod_{i=0}^J \frac{1}{\sqrt{2\pi \times 10^3}} \exp(-\frac{1}{2} \frac{\beta_j^2}{10^3}) \times \frac{2 \times 25}{\pi(\alpha^2 + 25^2)} \\
& \times \frac{2 \times 25}{\pi(b^2 + 25^2)} \times \frac{2 \times 25}{\pi(\lambda^2 + 25^2)} \quad (6.21)
\end{aligned}$$

To carry out Bayesian inference in the Topp-Leone exponentiated exponential model, we must specify a prior distribution for b, α, λ and β 's. We discussed the issue associated with specifying prior distributions in section 4, but for simplicity at this point, we assume that the prior distribution for α, λ and b is half-Cauchy on the interval $[0, 5]$ and for β is Normal with $[0, 5]$. Elementary application of Bayes rule as displayed in (3.17), applied to (6.20), then gives the posterior density for α, b, λ and β as equation (6.21). The result for this marginal posterior distribution get high-dimensional integral over all model parameters β_j, b, λ and α . To resolve this integral we 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.

6.3. Topp-Leone Exponential Extension Model

The probability density function (pdf) given by

$$\begin{aligned}
f(t, \alpha, \lambda, \theta, b) &= \frac{2\alpha b \lambda}{\theta} (1 + \frac{t}{\theta})^{\lambda-1} \exp(1 - (1 + \frac{t}{\theta})^{\lambda}) \\
& \times [1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda})]^{\alpha b-1} \{1 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))\} \\
& \times \{2 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))^b\}^{\alpha-1}
\end{aligned}$$

The survival function is given by

$$\begin{aligned}
S(t, \alpha, \lambda, \theta, b) &= 1 - F(t) = 1 - \{[(1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda})) \\
& \{2 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))^b\}^{\alpha}\}^{\delta_i}
\end{aligned}$$

We can state the likelihood function for right censored (as is our case the data are right censored) as

$$L = \prod_{i=0}^n Pr(t_i, \delta_i) = \prod_{i=0}^n [f(t_i)]^{\delta_i} [S(t_i)]^{1-\delta_i}$$

where δ_i 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

$$\begin{aligned}
L &= \prod_{i=0}^n \left[\frac{2\alpha b \lambda}{\theta} (1 + \frac{t}{\theta})^{\lambda-1} \exp(1 - (1 + \frac{t}{\theta})^{\lambda}) \right. \\
& \times [1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda})]^{\alpha b-1} \{1 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))\} \\
& \times \{2 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))^b\}^{\alpha-1} \}^{\delta_i} \\
& \times [1 - \{[(1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda})) \\
& \{2 - (1 - \exp(1 - (1 + \frac{t}{\theta})^{\lambda}))^b\}^{\alpha}\}^{\delta_i}]^{1-\delta_i} \}. \quad (6.22)
\end{aligned}$$

Thus, the joint posterior density is given by

$$\begin{aligned}
p(b, \beta, \alpha, \lambda | t, X) &\propto L(t | X, b, \beta) \times p(\beta) \times p(\alpha) \times p(\lambda) \times p(b) \\
&\propto \prod_{i=0}^n \left[\frac{2\alpha b \lambda}{e^{x\beta}} (1 + \frac{t}{e^{x\beta}})^{\lambda-1} \exp(1 - (1 + \frac{t}{e^{x\beta}})^{\lambda}) \right. \\
& \times [1 - \exp(1 - (1 + \frac{t}{e^{x\beta}})^{\lambda})]^{\alpha b-1} \{1 - (1 - \exp(1 - (1 + \frac{t}{e^{x\beta}})^{\lambda}))\} \\
& \times \{2 - (1 - \exp(1 - (1 + \frac{t}{e^{x\beta}})^{\lambda}))^b\}^{\alpha-1} \}^{\delta_i} \\
& \times [1 - \{[(1 - \exp(1 - (1 + \frac{t}{e^{x\beta}})^{\lambda})) \\
& \{2 - (1 - \exp(1 - (1 + \frac{t}{e^{x\beta}})^{\lambda}))^b\}^{\alpha}\}^{\delta_i}]^{1-\delta_i} \\
& \times \prod_{i=0}^J \frac{1}{\sqrt{2\pi \times 10^3}} \exp(-\frac{1}{2} \frac{\beta_j^2}{10^3}) \times \frac{2 \times 25}{\pi(\alpha^2 + 25^2)} \\
& \times \frac{2 \times 25}{\pi(b^2 + 25^2)} \times \frac{2 \times 25}{\pi(\lambda^2 + 25^2)}. \quad (6.23)
\end{aligned}$$

To carry out Bayesian inference in the Topp-Leone exponential extension model, we must specify a prior distribution for α, b, λ and β 's. We discussed the issue associated with specifying prior distributions in section 4, but for simplicity at this point, we assume that the prior distribution for α, λ and b is half-Cauchy on the interval $[0, 5]$ and for β is Normal with $[0, 5]$. Elementary application of Bayes rule as displayed in (3.17), applied to (6.22), then gives the posterior density for α, b, λ and β as equation (6.23). The result for this marginal posterior distribution get high-dimensional integral over all model parameters β_j, b, λ and α . To resolve this integral we 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.

6.4. The Data: Chemotherapy in Ovarian Cancer Patients

Following careful treatment of ovarian malignancy, patients may experience a course of chemotherapy. The two distinct types of chemotherapy treatment, Edmunson *et al.* (1979) looked at the counter tumor impacts of cyclophosphamide alone and cyclophosphamide joined with adriamycin. The preliminary included 26 ladies with insignificant remaining illness and who had encountered careful extraction of all tumor masses more prominent than 2 cm in diameter. Following surgery, the patients were additionally ordered by whether the lingering illness was totally or mostly extracted. The age of the patient and their execution status were likewise recorded toward the beginning of the preliminary. The reaction variable was the survival time in day following randomisation to one or other of the two chemotherapy medications. The data, which were obtained from Therneau (1986), are given in Table (1):

Table 1

Time	Status	Treat	Age	Rdisease	Pref
156	1	1	66	2	2
1040	0	1	38	2	2
59	1	1	72	2	1
421	0	2	53	2	1
329	1	1	43	2	1
769	0	2	59	2	2
365	1	2	64	2	1
770	0	2	57	2	1
1227	0	2	59	1	2
268	1	1	74	2	2
475	1	2	59	2	2
1129	0	2	53	1	1
464	1	2	56	2	2
1206	0	2	44	2	1
638	1	1	56	1	2
563	1	2	55	1	2
1106	0	1	44	1	1
431	1	1	50	2	1
855	0	1	43	1	2
803	0	1	39	1	1
115	1	1	74	2	1
744	0	2	50	1	1
477	0	1	64	2	1
448	0	1	56	1	2
353	1	2	63	1	2
377	0	2	58	1	1

Time: Survival time in days,

Status: Event indicator (0=censored, 1=uncensored),

Treat: Treatment (1=single, 2=combined),

Age: Age of patient in years,

Rdisease: Extent of residual disease (1=incomplete, 2=complete),

Perf: Performance status (1=good, 2=poor)

7. Implementation Using Stan

Bayesian modeling of Topp-Leone models in *rstan* package includes the creation of blocks, data, transformed data, parameter, transformed parameter, or generated quantities. To use the method for Topp-Leone exponential model, Topp-Leone exponentiated exponential, and Topp-Leone exponential extension, we will follow the following steps; starting with build a function for the model containing the accompanying items:

- Define the log survival.
- Define the log hazard.
- Define the sampling distributions for right censored data.

At that point the distribution ought to be built on the function definition blocks. The function definition block contains user defined functions. The data block states the needed data for the model. The transformed data block permits the definition of constants and transforms of the data. The parameters block declares the model parameters. The transformed parameters block allows variables to be defined in terms of data and parameters that may be used later and will be saved. The model block is where the log probability function is defined.

```
stan(file, model_name = "anon_model", model_code = "",
fit = NA, data = list(), pars = NA, chains = 4,
iter = 2000, warmup = floor(iter/2), thin = 1,
init = "random", algorithm = c("NUTS", "HMC",
"Fixed_param"),)
```

7.1. Model Specification

Presently we will look at the posterior estimates of the parameters when the Topp-Leone exponential, Topp-Leone exponentiated exponential and Topp-Leone exponential expansion model's are fitted to the previously mentioned data (information). Accordingly the importance of the likelihood (probability) 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 (The Stan Development Team 2014c). Here, we have likelihood as:

$$L(\theta|t) = \prod_{i=1}^n f(t_i)^{\delta_i} S(t_i)^{1-\delta_i}$$

$$= \prod_{i=1}^n \left(\frac{f(t_i)}{S(t_i)} \right)^{\delta_i} S(t_i) = \prod_{i=1}^n h(t_i)^{\delta_i} S(t_i),$$

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

$$\log L = \sum_{i=1}^n (\log[h(t_i)]^{\delta_i} + \log(S(t_i))).$$

7.1.1. Topp-Leone Exponential Model

The first model is Topp-Leone exponential:

$$y \sim TLexp(\alpha, b, \lambda),$$

where $\theta = \exp(X\beta)$ 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 α , b and β_j are taken to be half-Cauchy and normal as follows:

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

$$b, \alpha \sim HC(0,25).$$

The rstan package allows a model to be coded in a text file, here we write the Stan model code and save it in a separated text-file with name "model code1":

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

In this manner, we acquire the survival and hazard of the Topp-Leone Exponential model.

7.1.2. Topp-Leone Exponentiated Exponential Model

The second model is Topp-Leone exponentiated exponential model:

$$y \sim TLEXP(\alpha, b, \lambda, \theta),$$

where $\theta = \exp(X\beta)$. The Bayesian framework requires the specification of prior distributions for the parameters. Here, we stick to subjectivity and thus introduce weakly informative priors for the parameters. Priors for the β , α , λ , and b are taken to be normal and half-Cauchy as follows:

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

$$\alpha, b, \lambda \sim HC(0,25).$$

To fit this model in Stan, we first write the Stan model code and save it in a separated text-file with name "model code2":

```
library(rstan)
model_code2="
functions{
//defined survival
vector log_s(vector t, real shapel, real shape2, real
shape3, vector rate){
vector[num_elements(t)] log_s;
for(i in 1:num_elements(t)){
log_s[i]=log(1-(((1-exp(-t[i]/rate[i]))^(shape3))^
(shape2)*(2-((1-exp(-t[i]/rate[i]))^shape3)^shape2
))^shapel));
return log_s;}
//define log_ft
vector log_ft(vector t, real shapel, real shape2, real
shape3, vector rate){
vector[num_elements(t)] log_ft;
for(i in 1:num_elements(t)){
log_ft[i]=log(2*shapel*shape2*((shape3/rate[i]))*((
1-exp(-t[i]/rate[i]))^(shape3-1))*exp(-t[i]/rate[i
]))*((1-exp(-t[i]/rate[i]))^(shape3))^((shapel*sh
ape2)-1))*((1-((1-exp(-t[i]/rate[i]))^(shape3))^
(s hape2)))*((2-((1-exp(-t[i]/rate[i]))^shape3)^s
hape2)^(shapel-1)))));
return log_ft;}
//define log hazard
vector log_h(vector t, real shapel, real shape2, real
shape3, vector rate){
vector[num_elements(t)] log_h;
vector[num_elements(t)] logft;
vector[num_elements(t)] logs;
logft=log_ft(t, shapel, shape2, shape3, rate);
logs=log_s(t, shapel, shape2, shape3, rate);
log_h=logft-logs;
return log_h;}
//define the sampling distribution
real surv_TLEXP_lpdf(vector t, vector d, real
shapel, real shape2, real shape3, vector rate){
vector[num_elements(t)] log_lik;
```



```

real prob;
log_lik=d.*log_h(t,shapel,shape2,shape3,rate)+log_
s(t,shapel, shape2, shape3, rate);
prob=sum(log_lik);
return prob;}}

```

Therefore, we obtain the survival and hazard of the Topp-Leone exponentiated exponential model.

7.1.3. Topp-Leone Exponential Extension Model

The third model is Topp-Leone exponential extension model:

$$y \sim TLexpext(\alpha, b, \lambda, \theta),$$

where $\theta = \exp(X\beta)$. The Bayesian framework requires the specification of prior distributions for the parameters. Here, we stick to subjectivity and thus introduce weakly informative priors for the parameters. Priors for the β , α , λ , and b are taken to be normal and half-Cauchy as follows:

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

$$\alpha, b, \lambda \sim HC(0,25).$$

To fit this model in Stan, we first write the Stan model code and save it in a separated text-file with name "model code3":

```

library(rstan)
model_code3="
functions{
//defined survival
vector log_s(vector t, real shapel, real shape2, real
shape3, vector rate){
vector[num_elements(t)] log_s;
for(i in 1:num_elements(t)){
log_s[i]=log(1-(((1-exp(1-(1+t[i]/rate[i])^(shape
3)))^(shape2))* (2-(1-exp(1-(1+t[i]/rate[i])^shape3
))^(shape2))^(shapel))));
return log_s;
//define log_ft
vector log_ft(vector t, real shapel, real shape2,real
shape3, vector rate){
vector[num_elements(t)] log_ft;
for(i in 1:num_elements(t)){
log_ft[i]=log(2*shapel*shape2*((shape3/rate[i])*(1
+t[i]/rate[i])^(shape3-1)*exp(1-(1+t[i]/rate[i])^s
hape3))
*((1-exp(1-(1+t[i]/rate[i])^(shape3)))^( (shapel*sh
ape2)-1))* (1-(1-exp(1-(1+t[i]/rate[i])^(shape3)))^(
(shape2))* (2-(1-exp(1-(1+t[i]/rate[i])^shape3))^(sh
ape2)^(shapel-1))));
return log_ft;
//define log hazard
vector log_h(vector t, real shapel, real shape2, real
shape3, vector rate){
vector[num_elements(t)] log_h;

```

```

vector[num_elements(t)] logft;
vector[num_elements(t)] logs;
logft=log_ft(t,shapel,shape2,shape3,rate);
logs=log_s(t,shapel,shape2,shape3,rate);
log_h=logft-logs;
return log_h;
//define the sampling distribution
real surv_TLEXPEXT_lpdf(vector t, vector d, real
shapel, real shape2,real shape3, vector rate){
vector[num_elements(t)] log_lik;
real prob;
log_lik=d.*log_h(t,shapel,shape2,shape3,rate)+log_
s(t,shapel, shape2, shape3, rate);
prob=sum(log_lik);
return prob;}}

```

Therefore, we obtain the survival and hazard of the Topp-Leone exponential extension model.

7.2. Build the Stan

Stan contains an arrangement of blocks as stated previously; in the first block we will define the data block, in which we include 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). Then we create the parameter in block parameters, since we have more one parameter, we will do some changes for the parameters in side transformed parameters block. Finally, we arrange the model in blocks model. In these blocks, we put the prior for the parameters and the likelihood to get the posterior distribution for these model. We save this work in a file to use it in rstan package.

7.2.1. Topp-Leone Exponential 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 H columns)}
parameters {
vector[M] beta; // Coefficients in the linear
predictor (including intercept)
real<lower=0> shape; // shape parameter
real<lower=0> scale;}
transformed parameters {
vector[N] linpred;
vector[N] rate;
linpred = x*beta;
for (i in 1:N) {
rate[i] = exp(linpred[i]);}}
model {
shape ~ cauchy(0,25);

```

```

scale ~ cauchy(0,25);
beta ~ normal(0,1000);
y ~ surv_TLEXP(censor, shape, scale, rate);}
generated quantities{
  real dev;
  dev=0;
  dev=dev+(-2)*surv_TLEXP_lpdf(y|censor,shape,scale,
rate);}
"

```

7.2.2. Topp-Leone Exponentiated Exponential 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 H columns)}
parameters {
  vector[M] beta; // Coefficients in the linear
predictor (including intercept)
  real<lower=0> shapel; // shape parameter
  real<lower=0> shape2;
  real<lower=0> shape3;}
transformed parameters {
  vector[N] linpred;
  vector[N] rate;
  linpred = x*beta;
  for (i in 1:N) {
    rate[i] = exp(linpred[i]);}}
model {
  shapel ~ cauchy(0,25);
  shape2 ~ cauchy(0,25);
  shape3 ~ cauchy(0,25);
  beta ~ normal(0,1000);
  y ~ surv_TLEXP(censor, shapel,shape2, shape3,
rate);}
generated quantities{
  real dev;
  dev=0;
  dev=dev+(-2)*surv_TLEXP_lpdf(y|censor,shapel,sh
ape2,shape3,rate);}
"

```

7.2.3. Topp-Leone Exponential Extension 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 H columns)}

```

```

parameters {
  vector[M] beta; // Coefficients in the linear
predictor (including intercept)
  real<lower=0> shapel; // shape parameter
  real<lower=0> shape2;
  real<lower=0> shape3;}
transformed parameters {
  vector[N] linpred;
  vector[N] rate;
  linpred = x*beta;
  for (i in 1:N) {
    rate[i] = exp(linpred[i]);}}
model {
  shapel ~ cauchy(0,25);
  shape2 ~ cauchy(0,25);
  shape3 ~ cauchy(0,25);
  beta ~ normal(0,1000);
  y ~ surv_TLEXPExt(censor, shapel,shape2, shape3,
rate);}
generated quantities{
  real dev;
  dev=0;
  dev=dev+(-2)*surv_TLEXPExt_lpdf(y|censor,shapel,sh
ape2,shape3,rate);}
"

```

7.3. Creation of Data for Stan

Within this part, we force organize the data that we have to utilize for analysis, data arrangement requires model matrix X, number of predictors M, information regarding censoring and response variable. The number of observations is specified by N, that is, 26. Censoring is taken into account, where 0 stands for censored and 1 for uncensored values. Finally, every one of these things are consolidated in a recorded list as dat.

```

#Time
y<-c(156,1040,59,421,329,769,365,770,1227,268,475,
1129,464,1206,638,563,1106,431,855,803,115,744,477
,448,353,377)
censor<-c(1,0,1,0,1,0,1,0,0,1,1,0,1,0,1,1,0,1,0,0,
1,0,0,0,1,0)
#Treat
x1<-c(1,1,1,2,1,2,2,2,2,1,2,2,2,2,1,2,1,1,1,1,2,
1,1,2,2)
#Age
x2<-c(66,38,72,53,43,59,64,57,59,74,59,53,56,44,56
,55,44,50,43,39,74,50,64,56,63,58)
#Rdisease
x3<-c(2,2,2,2,2,2,2,2,1,2,2,1,2,2,1,1,1,2,1,1,2,1,
2,1,1,1)
#Perf
x4<-c(2,2,1,1,1,2,1,1,2,2,2,1,2,1,2,2,1,1,2,1,1,1,
1,2,2,1)
x2<-x2-mean(x2)
x<-cbind(1,x1,x2,x3,x4)
N = nrow(x)

```

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

7.4. Runing the Model Using Stan for Topp-Leone Exponential 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
betal=solve(crossprod(x),crossprod(x,log(y)))
#convert matrix to a vector
betal=c(betal)
S0<-stan(model_code=model_code1,init=list(list(bet
a=betal),list(beta=2*betal)),
data=dat,iter=5000,chains=2)
print(S0,c("beta","shape","scale","lp__","dev"),di
gits=2)
```

7.4.1. Summarizing Output

A summary of the parameter distributions can be obtained by using `print(S0)`, which provides posterior 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 distribution. Convergence can be diagnosed in several different ways. One way is to look at convergence statistics such as the potential scale reduction factor, Rhat (Gelman & Rubin, 1992), and the effective number of samples, `n_eff` (Gelman et al., 2013), both of which are outputs in the summary statistics with `print(S0)`. The function `rstan` approximates the posterior density of the fitted model and posterior summaries can be seen in the following tables. Table 2, which contain 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.

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]	7.63	0.04	1.73	4.93	6.45	7.42	8.50	11.79	1775	1
beta[1]	0.56	0.01	0.46	-0.33	0.29	0.56	0.82	1.51	3694	1
beta[2]	-0.08	0.00	0.03	-0.14	-0.09	-0.07	-0.06	-0.04	2746	1
beta[3]	-0.64	0.01	0.56	-1.85	-0.95	-0.60	-0.28	0.34	2595	1
beta[4]	-0.13	0.01	0.44	-1.08	-0.38	-0.11	0.15	0.67	2932	1
shape	2.13	0.07	3.79	0.15	0.54	1.08	2.32	11.03	2559	1
scale	5.40	0.50	18.25	0.16	0.92	2.18	4.98	26.90	1347	1
lp	-90.59	0.07	2.55	-96.90	-91.97	-90.13	-88.71	-87.04	1176	1

The inference of the posterior density after fitting the (Topp-Leone Exponential model) for ovarian cancer patients data using stan are reposted in Table 2. The posterior estimate for β_0 is 7.63 ± 1.73 and 95% credible interval is $(4.93, 11.79)$, which is statistically significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_1 is 0.56 ± 0.46 and 95% credible interval is $(-0.33, 1.51)$, which is statistically not significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_2 is -0.08 ± 0.00 and 95% credible interval is $(-0.14, -0.04)$, which is statistically significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_3 is -0.64 ± 0.56 and 95% credible interval is $(-1.85, 0.34)$, which is statistically not significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus

approximate convergence, posterior estimate for β_4 is -0.13 ± 0.44 and 95% credible interval is $(-1.08, 0.67)$, which is statistically not significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. The table displays the output from Stan. Here, the coefficient β_0 is the intercept, while the coefficient $\beta_j, j=1, \dots, 4$ is the effect of the covariate included in the model. The effective sample size given an indication of the underlying autocorrelation in the MCMC samples values 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. we can see in this (Figure 5) that the caterpillar plot is a horizontal plot of 3 quantiles of selected distribution, in this plot, credible intervals (by default 80%) for all the parameters, and the median of each chain are displayed. In addition, under the lines representing intervals, small colored areas are used to indicate which range the

value of the split \hat{R} statistic is in. This may be used to produce a caterpillar plot of posterior samples. In MCMC estimation, it is important to thoroughly assess convergence as it in (Figure 6) the rstan contains specialized function to visualise the model output and assess convergence.

```
stan_ac(S0,"beta")
traceplot(S0,"beta")
```

Usually Bayesian practitioners like to check whether a model estimation converges well or not, we want to plot a distribution (histogram) of each estimated parameter. The simplest way is using library coda.

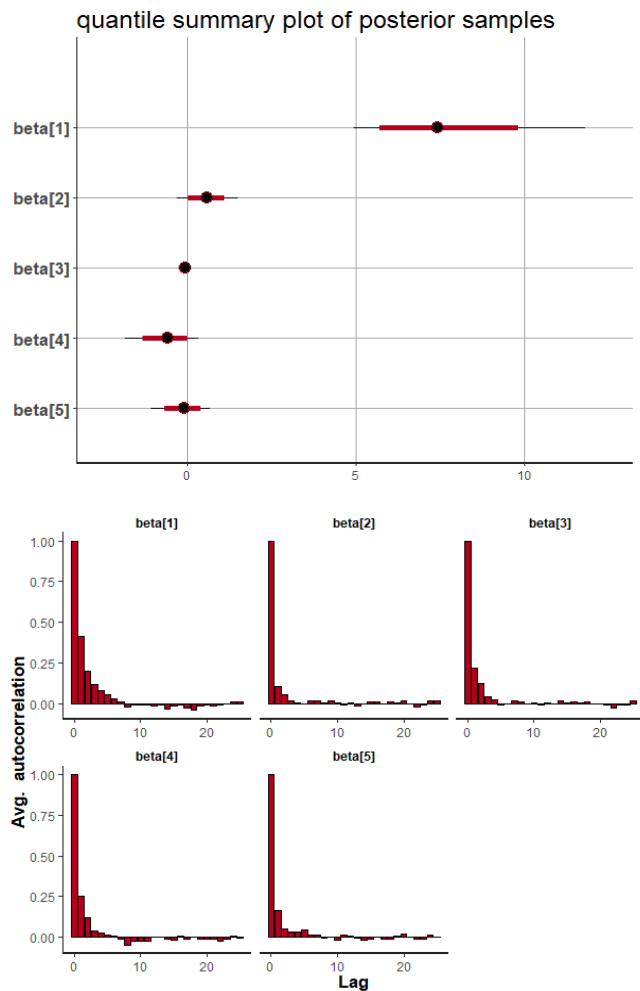


Figure 5. Caterpillar plot for Topp-Leone Exponential model

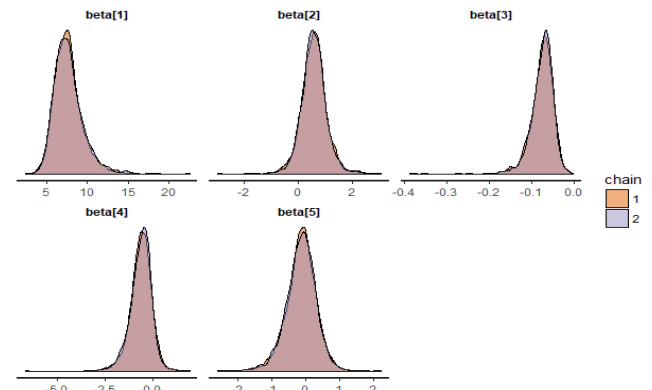
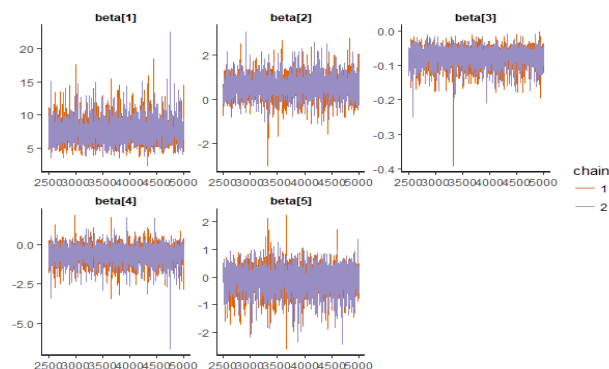


Figure 6. Checking model convergence using rstan, through inspection of the traceplots or the autocorrelation plot

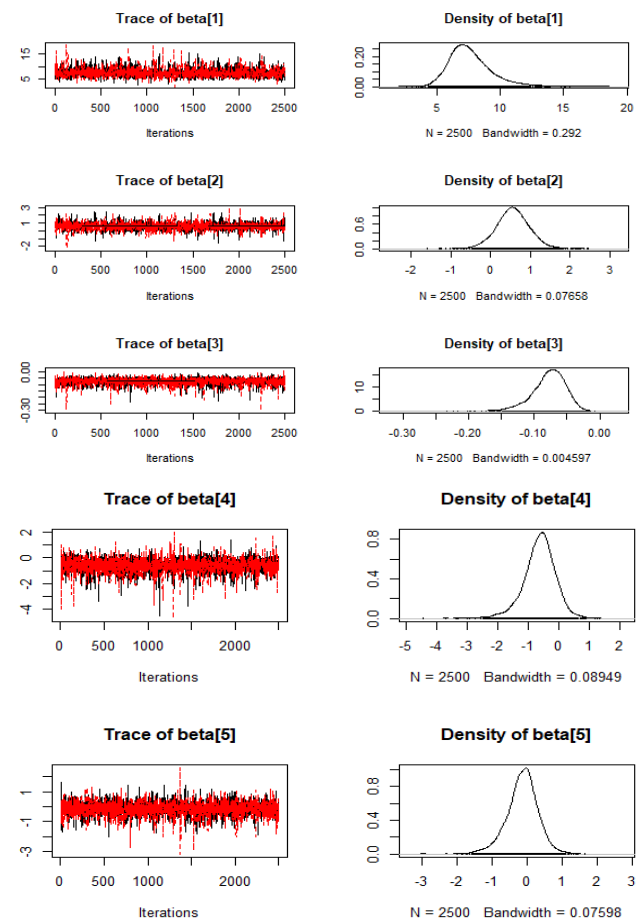


Figure 7. Checking model convergence using coda, through inspection of the simulated posterior density plots with trace plots of regressor variables obtained by HMC

7.5. Runing the Model Using Stan for Topp-Leone Exponentiated Exponential 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)
S1<-stan(model_code=model_code2,init=list(list(beta
a=beta1),list(beta=2*beta1),
data=dat,iter=5000,chains=2)
print(S1,c("beta","shape1","shape2","shape3","lp__
","dev"),digits=2)

```

7.5.1. Summarizing Output

The function `rstan` approximates the posterior density of

Table 3. 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]	7.77	0.05	1.91	4.90	6.50	7.52	8.67	12.29	1702	1
beta[1]	0.56	0.01	0.47	-0.38	0.28	0.56	0.84	1.48	3714	1
beta[2]	-0.08	0.00	0.03	-0.14	-0.09	-0.08	-0.06	-0.03	2460	1
beta[3]	-0.66	0.01	0.58	-1.94	-0.97	-0.62	-0.30	0.38	2597	1
beta[4]	-0.16	0.01	0.48	-1.21	-0.43	-0.13	0.14	0.71	2873	1
shape1	6.58	0.50	22.04	0.05	0.76	2.33	5.66	37.52	1949	1
shape2	2.90	0.09	5.20	0.05	0.44	1.22	3.31	15.94	3175	1
shape3	2.88	0.11	6.19	0.05	0.39	1.20	3.15	15.31	3189	1
lp	-91.32	0.07	2.68	-97.73	-92.83	-90.86	-89.39	-87.43	1333	1

The inference of the posterior density after fitting the (Topp-Leone exponential exponential model) for ovarian cancer patients data using `stan` are reposted in Table 3. The posterior estimate for β_0 is 7.77 ± 1.91 and 95% credible interval is $(4.90, 12.29)$, which is statistically significant. `Rhat` is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_1 is 0.56 ± 0.47 and 95% credible interval is $(-0.38, 1.48)$, which is statistically not significant. `Rhat` is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_2 is -0.08 ± 0.03 and 95% credible interval is $(-0.14, -0.03)$, which is statistically significant. `Rhat` is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_3 is -0.66 ± 0.58 and 95% credible interval is $(-1.94, 0.38)$, which is statistically not significant. `Rhat` is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. The selection of appropriate regressor variable can also be done by using a caterpillar plot. Caterpillar plots are popular plots in Bayesian inference for summarizing the quantiles of posterior samples. we can see in this (Figure 8) that the caterpillar plot is a horizontal plot of 3 quantiles of selected distribution. In MCMC estimation, it is important to thoroughly assess convergence as in (Figure 9) the `rstan`

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 is 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.

contains specialized function to visualise the model output and assess convergence.

```

stan_ac(S1,"beta")
traceplot(S1,"beta")

```

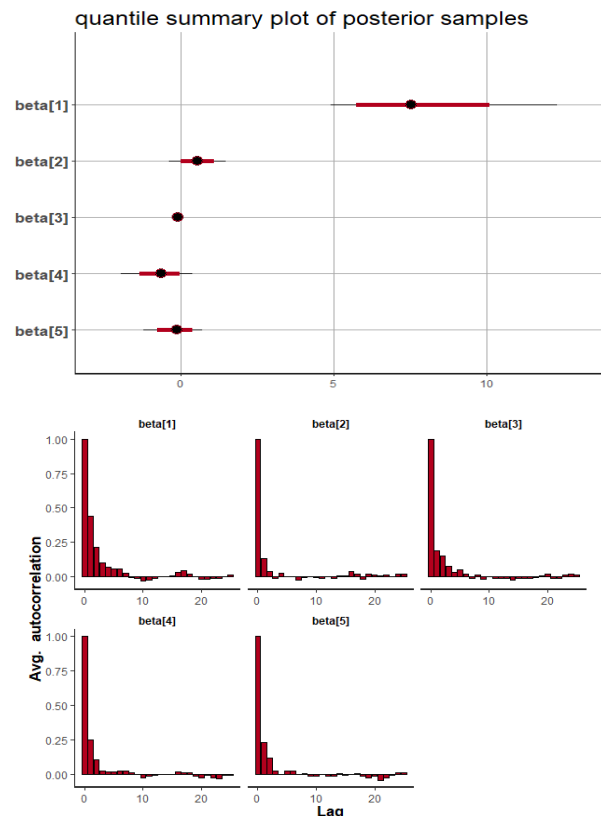


Figure 8. Caterpillar plot for Topp-Leone Exponentiated Exponential model

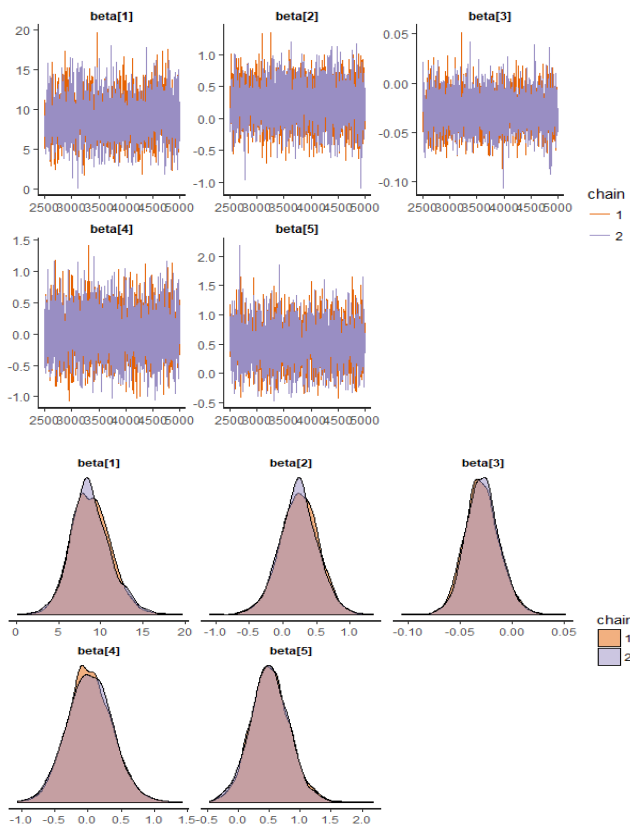


Figure 9. Checking model convergence using rstan, through inspection of the traceplots or the autocorrelation plot

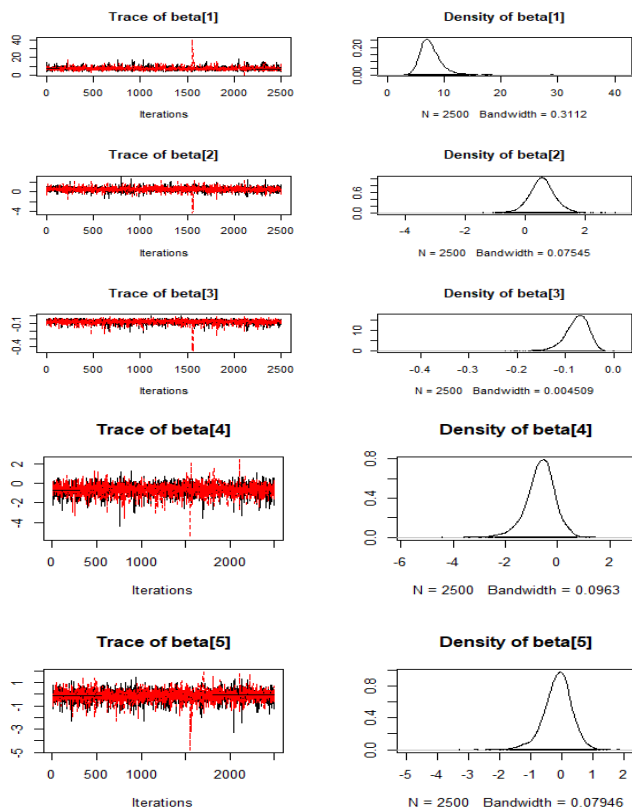


Figure 10. Checking model convergence using coda, through inspection of the simulated posterior density plots with trace plots of regressor variables obtained by HMC

Usually Bayesian practitioners like to check whether a model estimation converges well or not, we want to plot a distribution (histogram) of each estimated parameter. The simplest way is using library coda.

7.6. Runing the Model Using Stan for Topp-Leone Exponential Extension 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
betal=solve(crossprod(x),crossprod(x,log(y)))
#convert matrix to a vector
betal=c(betal)
S2<-stan(model_code=model_code3,init=list(list(bet
a=betal),list(beta=2*betal)),
data=dat,iter=5000,chains=2)
print(S2,c("beta","shape1","shape2","shape3","lp__
","dev"),digits=2)
```

7.6.1. Summarizing Output

The function rstan approximates the posterior density of the fitted model, and posterior summaries can be seen in the following tables. Table 4, 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 (Topp-Leone exponential extension model) for ovarian cancer patients data using stan are reposted in Table 4. The posterior estimate for β_0 is 6.13 ± 4.54 and 95% credible interval is $(-2.76, 13.90)$, which is statistically not significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_1 is 0.59 ± 0.48 and 95% credible interval is $(-0.35, 1.56)$, which is statistically not significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_2 is -0.08 ± 0.03 and 95% credible interval is $(-0.15, -0.04)$, which is statistically significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_3 is -0.66 ± 0.56 and 95% credible interval is $(-1.87, 0.41)$, which is statistically not significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. posterior estimate for β_4 is -0.14 ± 0.48 and 95% credible interval is $(-1.18, 0.76)$, which is statistically not significant. Rhat is close to 1.0, indication of good mixing of the three chains and thus approximate convergence. The selection of

appropriate regressor variable can also be done by using a caterpillar plot. Caterpillar plots are popular plots in Bayesian inference for summarizing the quantiles of posterior samples. We can see in (Figure 11) that the caterpillar plot is a horizontal plot of 3 quantiles of selected distribution. In MCMC estimation, it is important to

thoroughly assess convergence as it in (Figure 12) the `rstan` contains specialized function to visualise the model output and assess convergence

```
stan_ac(S2,"beta")
traceplot(S2,"beta")
```

Table 4. 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]	6.13	0.39	4.54	-2.76	2.86	6.61	9.50	13.90	133	1.01
beta[1]	0.59	0.01	0.48	-0.35	0.34	0.60	0.85	1.56	2702	1.00
beta[2]	-0.08	0.00	0.03	-0.15	-0.09	-0.07	-0.06	-0.04	1053	1.00
beta[3]	-0.66	0.01	0.58	-1.87	-0.95	-0.66	-0.31	0.41	2376	1.00
beta[4]	-0.14	0.01	0.48	-1.18	-0.39	-0.13	0.15	0.76	2365	1.00
shape1	7.17	0.92	14.26	0.19	1.37	3.36	7.66	29.84	239	1.01
shape2	3.68	0.18	6.27	0.16	0.64	1.71	4.67	16.77	1243	1.00
shape3	31.17	18.24	876.33	0.14	0.26	0.67	3.78	57.53	2309	1.00
lp	-90.65	0.24	2.83	-97.45	-92.13	-90.18	-88.55	-86.83	142	1.01

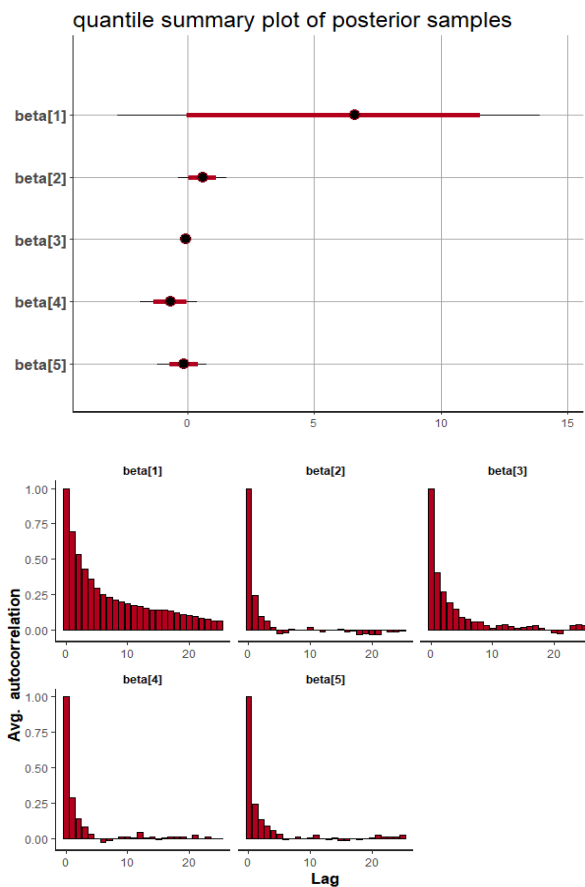


Figure 11. Caterpillar plot for Topp-Leone Exponential Extension model

Usually Bayesian practitioners like to check whether a model estimation converges well or not, we want to plot a distribution (histogram) of each estimated parameter. The simplest way is using library `coda`.

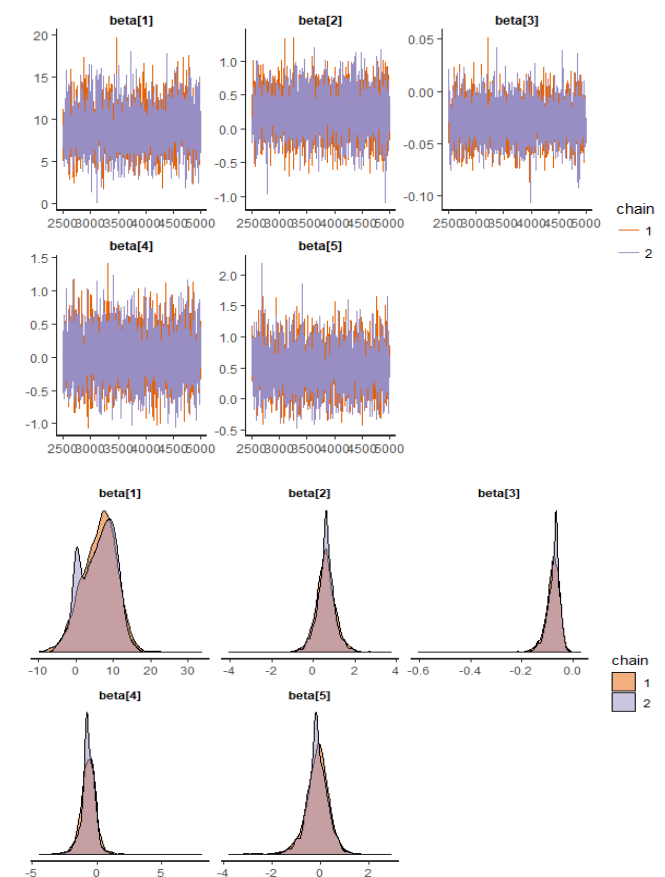


Figure 12. Checking model convergence using `rstan`, through inspection of the traceplots or the autocorrelation plot

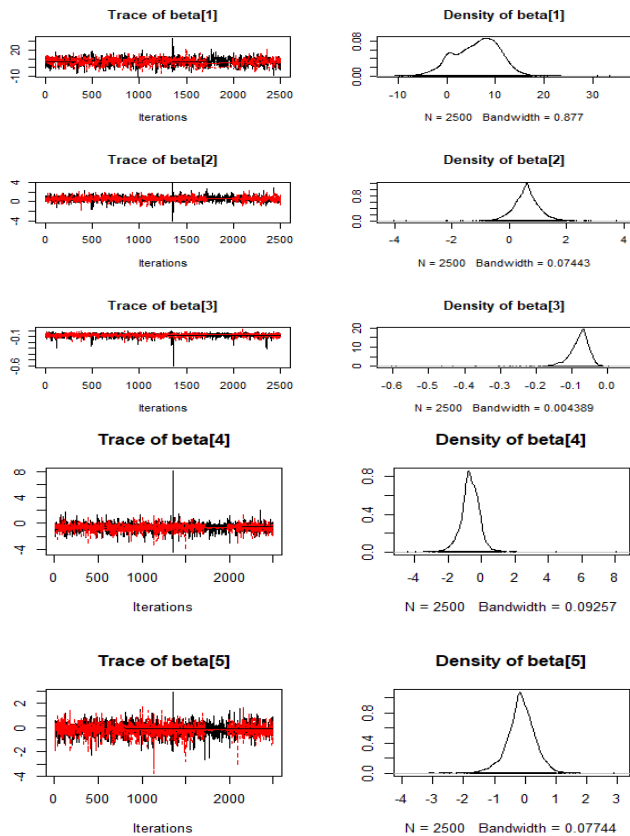


Figure 13. Checking model convergence using coda, through inspection of the simulated posterior density plots with trace plots of regressor variables obtained by HMC

8. Conclusions

It is a standout amongst the most critical issues in Bayes statistics how to make accurate Markov Chain Monte Carlo (MCMC) process. Here there are some MCMC methods, for example, the Metropolis method, the Gibbs sampler method, and Hamiltonian Monte Carlo method (HMC). Another problems is how we know which model best. Numerical calculation of widely applicable information criterion (WAIC, Watanabe 2010) and deviance, are help a lot for this tow problems depends on the accuracy of MCMC process and models. Here, therefore, Table 5 clearly demonstrates that Topp-Leone exponentiated exponential is the most proper model for the Stan as it has least estimation of deviance and WAIC, when contrasted with Topp-Leone exponential and Topp-Leone exponential extension.

Table 5. Model comparison of Topp-Leone exponential, Topp-Leone exponentiated exponential and Topp-Leone exponential extension models for the ovarian cancer patients data. It is evident from this table that Topp-Leone exponentiated exponential is much better than Topp-Leone exponential and Topp-Leone exponential Extension

Models	Stan Deviance	WAIC
Topp-Leone exponential	182.64	409.1
Topp-Leone exponentiated exponential	181.18	408.3
Topp-Leone exponential extension	183.30	411.6

REFERENCES

- [1] Akhtar, M. T. and Khan, A. A. (2014a). Bayesian analysis of generalized log-Burr family with R. Springer Plus, 3-185.
- [2] Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., & Riddell, A. (2017). Stan: A probabilistic programming language. Journal of Statistical Software, 76, 1-32.
- [3] Cordeiro, G. M., Ortega, E. M. and da Cunha, D. C. C. (2013). The exponentiated generalized class of distributions. Journal of Data Science, 11, 1-27.
- [4] Duane, A., Kennedy, A., Pendleton, B., and Roweth, D. (1987). Hybrid Monte Carlo. Physics Letters B, 195(2): 216-222. 24.
- [5] Edmunson, J.H., Fleming, T.R., Decker, D.G., Malkasian, G.D., Jorgenson, E.O., Jeffries, J.A., Webb, M.J. and Kvols, L.K. (1979). Different chemotherapeutic sensitivities and host factors affecting prognosis in advanced ovarian carcinoma versus minimal residual disease. Cancer Treatment Reports, 63, 241-7.
- [6] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014). Bayesian Data Analysis (3rd ed.), Chapman and Hall/CRC, New York.
- [7] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). Bayesian Data Analysis. Chapman & Hall/CRC Press, London, third edition.
- [8] Gelman, A. and Hill, J. (2007) Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press, New York.
- [9] Hoffman, Matthew D., and Andrew Gelman. (2012). "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." Journal of Machine Learning Research.
- [10] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, M., and Teller, E. (1953). Equations of state calculations by fast computing machines. Journal of Chemical Physics, 21: 1087-1092. 24, 25, 354.
- [11] Mohammed H AbuJarad and Athar Ali Khan. 2018, Exponential Model: A Bayesian Study with Stan. Int J Recent Sci Res. 9(8), pp. 28495-28506. DOI: <http://dx.doi.org/10.24327/ijrsr.2018.0908.2470>.
- [12] Nadarajah, S. and Kotz, S. (2003). Moments of some J-shaped distributions. Journal of Applied Statistics, 30, 311-317.
- [13] Rezaei, S., Sadr, B. B., Alizadeh, M., and Nadarajah, S. (2016). Topp-Leone Genetated Family of distribution: Properties and applications. Communications in Statistics-Theory and Methods, 46(6): 2893-2909.
- [14] Sangsanit, Y. and Bodhisuwan, W. (2016). The topp-leone generator of distributions: properties and inferences. Songklanakarin Journal of Science & Technology, 38(5).
- [15] Statisticat LLC (2015). Laplaces Demon: Complete Environment for Bayesian Inference. R package version 3.4.0, <http://www.bayesian-inference.com/software>.

- [16] Neal, R. M. (1994). An improved acceptance procedure for the hybrid monte carlo algorithm. *Journal of Computational Physics*, 111:194-203. 24.
- [17] Therneau, T.M. (1986). The COXREG Procedure. In SAS SUGI Supplemental Library User's Guide, version 5 edn, SAS Institute Inc., Cary, North Carolina.
- [18] Topp, C. W. and Leone, F. C. (1955). A family of j-shaped frequency function. *Journal of the American Statistical Association*, 50(269): 209-219.