

ISSN: 0976-3031

Available Online at <http://www.recentscientific.com>

CODEN: IJRSFP (USA)

International Journal of Recent Scientific Research
Vol. 8, Issue, 7, pp. 18692-18699, July, 2017

**International Journal of
Recent Scientific
Research**

DOI: 10.24327/IJRSR

Research Article

BAYESIAN ANALYSIS OF MARSHALL AND OLKIN FAMILY OF DISTRIBUTIONS

Najrullah Khan., Tanwir Akhtar and Athar Ali Khan

Department of Statistics and Operations Research Aligarh Muslim University,
Aligarh-202 002. India

DOI: <http://dx.doi.org/10.24327/ijrsr.2017.0807.0551>

ARTICLE INFO

Article History:

Received 17th April, 2017
Received in revised form 21th
May, 2017
Accepted 28th June, 2017
Published online 28th July, 2017

Key Words:

Bayesian Inference, Posterior, Laplace
Approximation, LaplacesDemon, JAGS,
Simulation, R.

ABSTRACT

In this paper an attempt has been made to fit Marshall and Olkin (1997) generalized family of distributions with special case of exponential distribution. Bayesian approach has been adopted to fit this model as survival model. Codes have been developed in R and JAGS to implement the Bayesian approach using both optimization and simulation tools. Right censoring case is illustrated for a real data and comparison of Bayesian simulation and optimization tools have been made.

Copyright © Najrullah Khan., Tanwir Akhtar and Athar Ali Khan, 2017, this is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

The field of survival analysis has experienced tremendous growth during the latter half of the 20th century. Survival analysis is a statistical method for data analysis where the outcome variable of interest is the time to the occurrence of an event. Hence, it is also referred to as time to event analysis. For example: time to recurrence of a disease, time until cardiovascular death after some treatment intervention, time to death of patient with certain disease, etc. Survival analysis techniques are used in a variety of disciplines including human and veterinary medicine, epidemiology, engineering and economy. Our main focus is on the application of survival analysis to data arising from medical research. Marshall and Olkin (A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families, *Biometrika* 84 (1997) 641-652) introduced a method for adding a new parameter into a family of distributions. In this paper an attempt has been made to fit Marshall and Olkin generalized family of distributions with special case of exponential distribution which play important role in reliability theory and survival analysis. Bayesian approach has been adopted to fit this model as survival model. Right censoring case is illustrated for a real data set. The tools and techniques used in this paper are in Bayesian environment, which are

implemented in Laplaces Demon package using Markov chain Monte Carlo (MCMC) tools (Statisticat LLC 2015). Goal of this package is to provide a complete and self contained Bayesian environment from within R, and approximate the posterior density using both analytic and simulation method like LaplaceApproximation and Markov chain Monte Carlo (MCMC) algorithms. The function LaplaceApproximation of LaplacesDemon approximates the posterior results analytically and then after convergence it gives simulated results using sampling importance resampling (SIR) method. In LaplaceApproximation function there is an argument called "Method". The method used to approximate posterior density is the Trust Region(TR) algorithm of Nosedal and Wright(1999). The TR algorithm attempts to reach its objective in the fewest number of iterations, is therefore very efficient, as well as safe. The efficiency of TR is attractive when model evaluations are expensive. Another important function of this package is LaplacesDemon, which maximizes the logarithm of the unnormalized joint posterior density using one of the MCMC algorithm. In LaplacesDemon the algorithm used for simulation is Independent Metropolis. Independent Metropolis(IM) algorithm proposed by Hastings (1970) and popularized by Tierney (1994). JAGS ("Just Another Gibbs Sampler") can be run directly from R using R2jags package. Package R2jags is also used for simulation from posterior density. The JAGS

*Corresponding author: **Najrullah Khan**

Department of Statistics and Operations Research Aligarh Muslim University, Aligarh-202 002. India

function takes data and starting values as input. It automatically writes a jags script, calls the model, and saves the simulations for easy access in R. Real survival data set is used to illustrate the implementation in R and JAGS. Thus, Bayesian analysis of Marshall-Olkin Exponential distribution 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 and JAGS code for approximating posterior densities with Laplace Approximation and simulation tools.
- To illustrate numeric as well as graphics summaries of posterior densities.

The Marshall-Olkin Exponential Distribution

Marshall Olkin distribution due to Marshall and Olkin (1997) is given by (S. Nadarajah and R. Rocha, Newdistns (2016)) the pdf

$$f(t) = \frac{bg(t)}{[b + (1 - b)G(t)]^2} \tag{1}$$

where g(t) and G(t) are corresponding pdf and cdf of t, respectively. Now, its generalized family of distributions with special case of exponential distribution when $g(t) \sim exp(\theta)$. This density is abbreviated as MOExp(b, θ).

The pdf, cdf, survival function and hazard function of MOExp(b, θ) are given below

$$f(t) = \frac{b/\theta exp(-t/\theta)}{[b + (1 - b)(1 - exp(-t/\theta))]^2} \tag{2}$$

$$F(t) = \frac{1 - exp(-t/\theta)}{[b + (1 - b)(1 - exp(-t/\theta))]} \tag{3}$$

$$S(t) = 1 - F(t) = \frac{b exp(-t/\theta)}{[b + (1 - b)(1 - exp(-t/\theta))]} \tag{4}$$

$$h(t) = \frac{f(t)}{S(t)} \tag{5}$$

Functions for Marshall-Olkin Exponential distribution in R require(Newdistns)

1. R code for probability density function is

```
dmoexp<-function(x, b,theta,...){
d1<-dmog(x,"exp",beta=b,rate=1/theta) return(d1)
}
```
2. R code for cumulative density function is

```
pmoexp<-function(x, b,theta,...){
p1<-pmog(x,"exp",beta=b,rate=1/theta) return(p1)
}
```
3. R code for random generation function is

```
rmoexp<-function(n,b,theta,...){
```

- ```
r1<-rmog(n,"exp",beta=b,rate=1/theta) return(r1)
}
```
4. R code for survival function is  

```
smoexp<-function(x, b,theta,...){
s1<-pmog(x,"exp",beta=b,rate=1/theta,lower.tail=F)
return(s1)
}
```

**The Prior Distributions**

In Bayesian paradigm, it is needed to specify prior information regarding the value of the parameter of interest or information that is available before analyzing the experimental data by using a probability distribution function. This probability distribution function is called the prior probability distribution, or simply the *prior*, since it reflects information about parameter prior to observing experimental data. Here some prior distributions are discussed according to their uses in subsequent Bayesian survival models.

**Weakly Informative Priors**

Weakly Informative Prior (WIP) distribution uses prior information for regularization and stabilization, providing enough prior information to prevent results that contradict our knowledge or problems such as an algorithmic failure to explore the state-space. Another goal is for WIPs to use less prior information than is actually available. A WIP should provide some of the benefit of prior information while avoiding some of the risk from using information that doesn't exist (Statisticat LLC, 2015). A popular WIP for a centered and scaled predictor (Gelman, 2008) may be

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

where  $\theta$  is normally-distributed according to a mean of 0 and a variance of 10,000, which is equivalent to a precision of  $1.0 \times 10^{-4}$ . In this case, the density for  $\theta$  is nearly flat (see Figure 1). Prior distributions that are not completely flat provide enough information for the numerical approximation algorithm to continue to explore the target density, the posterior distribution.

**The half-Cauchy prior distribution**

The probability density function of half-Cauchy distribution with scale parameter  $\alpha$  is given by

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

The mean and variance of the half-Cauchy distribution do not exist, but its mode is equal to 0. The half-Cauchy distribution with scale  $\alpha = 25$  is a recommended, default, noninformative prior distribution for a scale parameter. At this scale  $\alpha = 25$ , the density of half-Cauchy is nearly flat but not completely (see Figure 1), prior distributions that are not completely flat provide enough information for the numerical approximation algorithm to continue to explore the target density, the posterior distribution. The inverse-gamma is often used as a noninformative prior distribution for scale parameter, however, this model creates problem for scale parameters near zero. Gelman and Hill (2007) recommend that, the uniform, or if

more information is necessary the half-Cauchy is a better choice. Thus, in this paper, the half-Cauchy distribution with scale parameter  $\alpha = 25$  is used as a noninformative prior distribution (Khan, Akhtar and Khan, 2016).

$$I \approx \int f(\hat{\theta}) \exp \left( -n \left[ h(\hat{\theta}) + (\theta - \hat{\theta})h'(\hat{\theta}) + \frac{(\theta - \hat{\theta})^2}{2} h''(\hat{\theta}) \right] \right) d\theta.$$

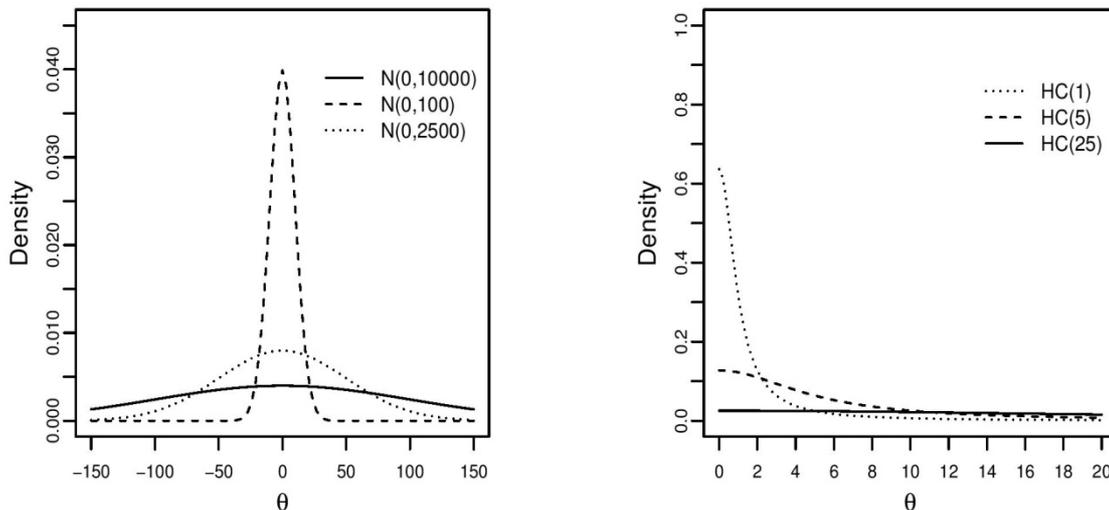


Figure 1 Weakly informative priors for the parameters.

**The Laplace Approximation**

Many simple Bayesian analyses based on noninformative prior distribution give similar results to standard non-Bayesian approaches, for example, the posterior  $t$ -interval for the normal mean with unknown variance. The extent to which a noninformative prior distribution can be justified as an objective assumption depends on the amount of information available in the data; in the simple cases as the sample size  $n$  increases, the influence of the prior distribution on posterior inference decreases. These ideas, sometimes referred to as asymptotic approximation theory because they refer to properties that hold in the limit as  $n$  becomes large. Thus, a remarkable method of asymptotic approximation is the Laplace approximation which accurately approximates the unimodal posterior moments and marginal posterior densities in many cases. In this section we introduce a brief description of Laplace approximation method.

Suppose  $-h(\theta)$  is a smooth, bounded unimodal function, with a maximum at  $\hat{\theta}$ , and  $\theta$  is a scalar. By Laplace's method (e.g., Tierney and Kadane, 1986), the integral

$$I = \int f(\theta) \exp[-nh(\theta)] d\theta$$

can be approximated by

$$\hat{I} = f(\hat{\theta}) \sqrt{\frac{2\pi}{n}} \sigma \exp[-nh(\hat{\theta})],$$

where

$$\sigma = \left[ \frac{\partial^2 h}{\partial \theta^2} \Big|_{\hat{\theta}} \right]^{-1/2}.$$

As presented in Mosteller and Wallace (1964), Laplace's method is to expand about  $\hat{\theta}$  to obtain:

Recalling that  $h'(\hat{\theta}) = 0$ , we have

$$\begin{aligned} I &\approx \int f(\hat{\theta}) \exp \left[ -n \left( h(\hat{\theta}) + \frac{(\theta - \hat{\theta})^2}{2} h''(\hat{\theta}) \right) \right] d\theta \\ &= f(\hat{\theta}) \exp[-nh(\hat{\theta})] \int \exp \left( \frac{-n(\theta - \hat{\theta})^2}{2\sigma^2} \right) d\theta \\ &= f(\hat{\theta}) \sqrt{\frac{2\pi}{n}} \sigma \exp[-nh(\hat{\theta})]. \end{aligned}$$

Intuitively, if  $\exp[-nh(\theta)]$  is very peaked about  $\hat{\theta}$ , then the integral can be well approximated by the behavior of the integrand near  $\hat{\theta}$ . More formally, it can be shown that

$$I = \hat{I} \left[ 1 + O\left(\frac{1}{n}\right) \right].$$

To calculate moments of posterior distributions, we need to evaluate expressions such as:

$$E[g(\theta)] = \frac{\int g(\theta) \exp[-nh(\theta)] d\theta}{\int \exp[-nh(\theta)] d\theta}, \tag{6}$$

where  $\exp[-nh(\theta)] = L(\theta | y)p(\theta)$  (e.g., Tanner, 1996).

**Fitting with Laplace Approximation**

The LaplaceApproximation is a family of asymptotic techniques used to approximate the integrals. It is accurately approximates unimodal posterior moments and marginal posterior distributions in many cases. This function deterministically maximizes the logarithm of unnormalized joint posterior density with one of several optimization

algorithms. The goal of LaplaceApproximation is to estimate the posterior mode and variance of each parameter. The function and arguments are as follows

LaplaceApproximation(Model, parm, Data, Interval=1.0E-6, Iterations=100, Method="SPG",Samples=1000, Cov Est="Hessian",sir=TRUE,Stop.Tolerance=1.0E-5,CPUs=1,Type="PSOCK")

First argument Model is used as a user-defined function. The user-defined function is where the model is specified. LaplaceApproximation passes two arguments to the model function, parm and Data. The parm argument requires a vector of initial values equal in length to the number of parameters. Data argument accepts a list of data. By default method is Method=SPG. Several methods of optimization are available in LaplaceApproximation but we have found that trust region is better than others.

**Bayesian Analysis of Marshall-Olkin Exponential Model**

**The Model**

The pdf of MOExp(b, θ) is given by

$$f(y) = \frac{b/\theta \exp(-y/\theta)}{[b + (1-b)(1 - \exp(-y/\theta))]^2} \tag{7}$$

The survival function is given by

$$S(y) = 1 - F(y) = \frac{b \exp(-y/\theta)}{[b + (1-b)(1 - \exp(-y/\theta))]} \tag{8}$$

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

$$L = \prod_{i=1}^n Pr(y_i, \delta_i) \tag{9}$$

$$= \prod_{i=1}^n [f(y_i)]^{\delta_i} [S(y_i)]^{1-\delta_i} \tag{10}$$

where δ<sub>i</sub> 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

$$L = \prod_{i=1}^n \left[ \frac{b/\theta \exp(-y_i/\theta)}{[b + (1-b)(1 - \exp(-y_i/\theta))]^2} \right]^{\delta_i} \left[ \frac{b \exp(-y_i/\theta)}{[b + (1-b)(1 - \exp(-y_i/\theta))]} \right]^{1-\delta_i}$$

Thus, the joint posterior density is given by

$$p(\beta, b | y, X) \propto L(y, X | \beta, b) \times p(\beta) \times p(b) \propto \prod_{i=1}^n \left[ \frac{b/e^{X\beta} \exp(-y_i/e^{X\beta})}{[b + (1-b)(1 - \exp(-y_i/e^{X\beta}))]} \right]^{\delta_i}$$

$$\left[ \frac{b \exp(-y_i/e^{X\beta})}{[b + (1-b)(1 - \exp(-y_i/e^{X\beta}))]} \right]^{1-\delta_i} \times \prod_{j=1}^J \frac{1}{\sqrt{2\pi \times 10^3}} \exp\left(-\frac{1}{2} \frac{\beta_j^2}{10^3}\right) \times \frac{2 \times 25}{\pi(b^2 + 25^2)}$$

which is not in closed form. Consequently, the marginal posterior densities of parameters β<sub>j</sub> and b are also not in closed form. These marginal densities are the basis of Bayesian inference, and therefore one needs to use numerical integration or MCMC methods. However, due to the availability of computer software package like R and LaplacesDemon, the required model can easily be fitted in Bayesian paradigm using Laplace approximation as well as MCMC techniques.

**Data set: Survival of multiple myeloma patients**

Let us introduce a lifetime data set due to Krall *et al.*(1975) and discussed by Collet (2003), so that all the concepts and computations will be discussed around that data. Multiple myeloma is a malignant disease characterized by the accumulation of abnormal plasma cells, a type of white blood cell, in the bone marrow. The proliferation of the abnormal plasma cells within the bone causes pain and the destruction of bone tissue. Patients with multiple myeloma also experience anaemia, haemorrhages, recurrent infections and weakness. Unless treated, the condition is invariably fatal. The aim of a study carried out at the Medical Centre of the University of West Virginia, USA, was to examine the association between the values of certain covariates and the survival time of patients. In the study, the primary response variable was the time, in months, from diagnosis until death from multiple myeloma. A part of this data is considered for the study. The data contains the following variables: (Table 1 is header part of data)

- Age: age of the patient
- Sex: sex of the patient (0 = male, 1 = female)
- Bun: blood urea nitrogen
- Ca: serum calcium
- Hb: serum haemoglobin
- PC: percentage of plasma cells
- BJ: Bence-Jones protein (0 = absent, 1 = present)

**Table 1** Survival times of patients in a study on multiple myeloma.

| Survival time | Status | Age | Sex | Bun | Ca | Hb   | PC  | BJ |
|---------------|--------|-----|-----|-----|----|------|-----|----|
| 13            | 1      | 66  | 0   | 25  | 10 | 14.6 | 18  | 1  |
| 52            | 0      | 66  | 0   | 13  | 11 | 12.0 | 100 | 0  |
| 6             | 1      | 53  | 1   | 15  | 13 | 11.4 | 33  | 1  |
| 40            | 1      | 69  | 0   | 10  | 10 | 10.2 | 30  | 1  |
| 10            | 1      | 65  | 0   | 20  | 10 | 13.2 | 66  | 0  |
| 7             | 0      | 57  | 1   | 12  | 8  | 9.9  | 45  | 0  |

**Implementation using Laplaces Demon**

Bayesian modeling of MOExp (b, θ) distribution in LaplacesDemon package includes the creation of data, model specification, initial values and fitting model using LaplaceApproximation and LaplacesDemon functions.

### Creation of data for LaplacesDemon

The data of Krall *et al.* (1975) is used for Bayesian modeling of MOExp( $b, \theta$ ) distribution. The data set dat.txt contains seven regressor variables namely, Age, Sex, Bun, Ca, Hb, PC and BJ. The design matrix X is extracted from survreg object M2 and is named as X, naming of predictors, naming of parameters, information regarding censoring and response variable.

```
require(LaplacesDemon)
censor<-c(1,0,1,1,1,0,1,0,1,1,1,1,1,1,0,1,0,1,0,0,1,1,1,1,0,1,1,1,
1,1,1,1,1,1,1,0,1,0,1,1,1,1,0,1,1,0)
dat=read.table("dat.txt",header=TRUE)
require(survival)
M2<-
survreg(Surv(time,Status)~Agec60+Sext+Bunc+Cac+Hbc+Pce
llsc+Protein,data=dat)
X<-model.matrix(M2)
J<-ncol(X)
N=nrow(X)
y<-dat$time
mon.names<-c("LP","b")
parm.names<-as.parm.names(list(beta=rep(0,J),log.b=0))
MyData<-list(J=J,N=N,X=X,mon.names=mon.names,
parm.names=parm.names,y=y,censor=censor)
```

In this case, there are two parameters beta and b which must be specified in vector parm.names. The logposterior LP is included as monitored variables in vector mon.names. The number of observations is specified by N, that is, 48. Censoring is taken into account, where 0 stands for censored and 1 for uncensored values. Finally, all these things are combined with object name MyData which returns the data in a list.

### Model Specification

The function LaplaceApproximation can fit any Bayesian model for which likelihood and prior are specified(e.g., Khan, Akhtar and Khan, 2016).

To use this method one must specify a model

$$y \sim MOExp(b, \theta)$$

Since, b and  $\theta$  both are positive, hence, logarithm link function is used to spread them on the whole real line, that is

$$\log \theta = X\beta$$

$$\theta = \exp(X\beta)$$

where, X is model matrix and  $\beta$  is the vector of regression coefficients. Prior probabilities are specified for  $\beta$  and b are

$$\beta_j \sim N(0,1000), \quad j = 1, \dots, J$$

$$b \sim HC(25).$$

The large variance indicates a lot of uncertainty about each  $\beta$  and is hence a weak informative prior distribution. Similarly, half-Cauchy is weakly informative prior for b(Statisticat LLC 2015).

```
Model<-function(parm,Data) {
beta<-parm[1:Data$J]
b<-exp(parm[Data$J+1])
beta.prior<-sum(dnormv(beta,0,1000,log=TRUE))
```

```
b.prior<-dhalfcauchy(b,25,log=TRUE)
mu<-tcrossprod(beta,Data$X)
theta<-exp(mu)
lf<-log(b/theta)-(y/theta)-2*log(b+(1-b)*(1-exp(-y/theta)))
ls<-log(b)-(y/theta)-log(b+(1-b)*(1-exp(-y/theta)))
LL<-censor*lf+(1-censor)*ls
LL<-sum(LL)
LP<-LL+beta.prior+b.prior
Modelout<-list(LP=LP,Dev=-2*LL,Monitor=c(LP,b),
yhat=rmoexp(length(y),b,theta),parm=parm) return(Modelout)
}
```

### Initial Values

To start the optimization, the function LaplaceApproximation requires a vector of initial values for the parameters. Each initial value is a starting point for the estimation of a parameter. So all the beta parameters have been set equal to zero and the remaining parameter, log.b (since,  $b > 0$ ,  $\log(b)$  will spread on whole real line, and optimization becomes unconstrained, which is requirement of the algorithm), has been set equal to  $\log(1)$ , which is zero, LaplaceApproximation will optimize initial values using trust region (TR) algorithm. The order of the elements of the vector of initial values must match the order of the parameters. Thus, define a vector of initial values. However, instead of taking this default guess we have taken regression coefficients obtain from fitting survreg function to represent a fitted parametric survival regression model. This empirical guess converges faster. Initial.Values<-c(coef(M2),log(2))

### Laplace Approximation

To fit the above specified model, an object Fit1 has been created as a result of using LaplaceApproximation function. So the summary of results are printed by the function print.

```
Fit1<-LaplaceApproximation(Model,Initial.Values,
Data=MyData,Iterations=1000,Method="TR")
print(Fit1)
```

### Summarizing Output

The function LaplaceApproximation approximates the posterior density of the fitted model, and posterior summaries can be seen in the following tables. Table 2 represents the analytic results using LaplaceApproximation method while Table 3 represents the simulated results using sampling importance resampling(SIR) method.

**Table 2** Approximated posterior summary using LaplaceApproximation function with posterior mode, posterior sd and their quantiles

| Parameter | Mode    | SD     | LB      | UB      |
|-----------|---------|--------|---------|---------|
| Intercept | 2.7584  | 0.3651 | 2.0181  | 3.4788  |
| Age       | 0.0056  | 0.0217 | -0.0379 | 0.0493  |
| Sex       | -0.2446 | 0.3038 | -0.8523 | 0.3631  |
| Bun       | -0.0189 | 0.0037 | -0.0265 | -0.0114 |
| Ca        | 0.0007  | 0.0525 | -0.1043 | 0.1058  |
| Hb        | -0.0260 | 0.0181 | -0.0624 | 0.0102  |
| Pcells    | 0.0001  | 0.0050 | -0.0101 | 0.0103  |
| Protein   | 0.5967  | 0.3353 | -0.0740 | 1.2675  |
| log.b     | 0.7367  | 0.5711 | -0.4055 | 1.8789  |

**Table 3** Simulated posterior summary using sampling importance resampling with posterior mode, posterior sd and their quantiles.

| Parameter | Mode    | SD     | MCSE   | ESS     | LB        | Median  | UB      |
|-----------|---------|--------|--------|---------|-----------|---------|---------|
| Intercept | 3.0275  | 0.3530 | 0.0112 | 1000.00 | 2.3925    | 2.9889  | 3.6949  |
| Age       | 0.0048  | 0.0243 | 0.0007 | 1000.00 | -0.0395   | 0.0041  | 0.0476  |
| Sex       | -0.1940 | 0.3325 | 0.0105 | 1000.00 | -0.8712   | -0.2232 | 0.5045  |
| Bun       | -0.0176 | 0.0042 | 0.0001 | 1000.00 | -0.0253   | -0.0178 | -0.0084 |
| Ca        | 0.0043  | 0.0857 | 0.0027 | 1000.00 | -0.1656   | 0.0142  | 0.1531  |
| Hb        | -0.0268 | 0.0179 | 0.0005 | 1000.00 | -0.0626   | -0.0262 | 0.0079  |
| Pcells    | 0.0002  | 0.0051 | 0.0001 | 1000.00 | -0.0106   | 0.0004  | 0.0092  |
| Protein   | 0.6440  | 0.3708 | 0.0117 | 1000.00 | -0.0191   | 0.6566  | 1.4582  |
| log.b     | 0.2613  | 0.5158 | 0.0163 | 1000.00 | -0.8889   | 0.2947  | 1.1746  |
| Deviance  | 312.68  | 3.4483 | 0.1090 | 1000.00 | 307.2626  | 312.21  | 320.91  |
| LP        | -195.00 | 1.7243 | 0.0545 | 1000.00 | -199.1114 | -194.76 | -192.29 |
| b         | 1.4730  | 0.7413 | 0.0234 | 1000.00 | 0.4111    | 1.3428  | 3.2369  |

**Fitting with Laplaces Demon**

The LaplacesDemon function is the main function of LaplacesDemon package. This function maximizes the logarithm of the unnormalized joint posterior density with MCMC and provides samples of the marginal posterior distributions, deviance, and other monitored variables. The LaplacesDemon function for this model, simulates the data from posterior density with Independent Metropolis (IM) algorithm. The main arguments of the LaplacesDemon can be seen by using the function args as:

```
LaplacesDemon(Model, Data, Initial.Values, Covar= NULL,
Iterations= 10000, Status= 1000, Thinning= 100, Algorithm=
"RWM", Specs= NULL,...)
```

The arguments Model and Data specify the model to be implemented and list of data, which are specified in the previous section, respectively. The argument Iterations accepts integers larger than 10, and determines the number of iterations that Laplace’s Demon will update the parameters while searching for target distributions.

The function LaplacesDemon is used to analyze the same data, that is, multiple myeloma patients survival data.

```
Initial.Value<-as.initial.values(Fit1)
```

```
Fit2<- LaplacesDemon(Model, Data=MyData, Initial.Values,
Covar=Fit1$Covar, Iterations=80000, Status=0, Thinning=1,
Algorithm="IM",
Specs=list(mu=Fit1$Summary1[1:length(Initial.Values),1]))
print(Fit2)
```

**Summarizing Output**

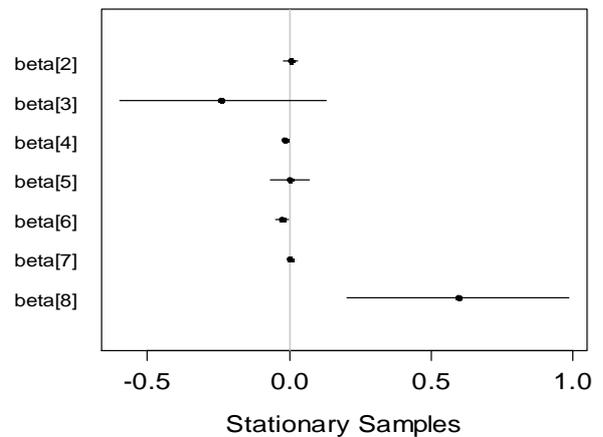
The function LaplacesDemon for this regression model, simulates the data from the posterior density with Independent Metropolis algorithm, and summaries of results are reported in the following table.

**Fitting with JAGS**

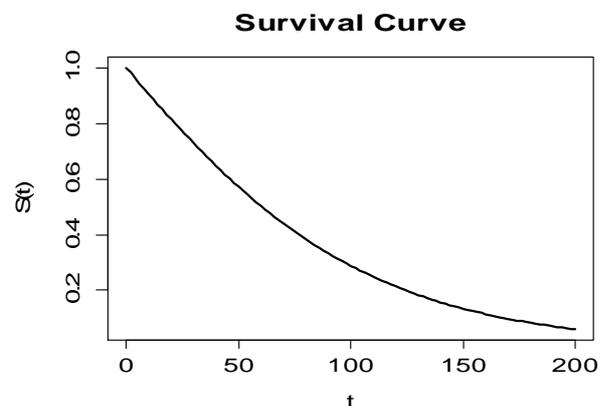
Let us consider the Bayesian analysis of the same data, that is, multiple myeloma data with JAGS using its interface of R, that is, R2jags package of R. R2jags is designed for inference on Bayesian models using Markov chain Monte Carlo (MCMC) simulation. Running a model refers to generating samples from the posterior distribution of the model parameters. It is designed to work directly with JAGS from R. The jags function takes data and starting values as input. It automatically writes a jags script, calls the model, and saves the simulations for easy access in R.

**Table 4** Simulated posterior summary using Laplaces Demon function with posterior mean, posterior sd and their quantiles

| Parameter | Mean    | SD     | MCSE   | ESS     | LB        | Median  | UB      |
|-----------|---------|--------|--------|---------|-----------|---------|---------|
| Intercept | 2.8012  | 0.2159 | 0.0046 | 4265.87 | 2.3872    | 2.7990  | 3.2325  |
| Age       | 0.0055  | 0.0129 | 0.0028 | 4156.24 | -0.0201   | 0.0055  | 0.0308  |
| Sex       | -0.2384 | 0.1827 | 0.0035 | 4912.03 | -0.5976   | -0.2408 | 0.1298  |
| Bun       | -0.0188 | 0.0022 | 0.000  | 4157.74 | -0.0232   | -0.0188 | -0.0143 |
| Ca        | 0.0017  | 0.0361 | 0.0007 | 4341.61 | -0.0679   | 0.0011  | 0.0719  |
| Hb        | -0.0260 | 0.0106 | 0.0002 | 4760.89 | -0.0475   | -0.0259 | -0.0048 |
| Pcells    | 0.0001  | 0.0030 | 0.0000 | 4447.73 | -0.0057   | 0.0001  | 0.0061  |
| Protein   | 0.5927  | 0.1991 | 0.0039 | 4151.29 | 0.2025    | 0.5952  | 0.9866  |
| log.b     | 0.6502  | 0.3425 | 0.0074 | 4264.05 | -0.0108   | 0.6488  | 1.3065  |
| Deviance  | 307.31  | 1.4402 | 0.0405 | 2338.82 | 305.1928  | 307.09  | 310.76  |
| LP        | -192.32 | 0.7213 | 0.0201 | 2341.73 | -194.0470 | -192.21 | -191.26 |
| b         | 2.0315  | 0.7122 | 0.0153 | 4201.41 | 0.9892    | 1.9133  | 3.6933  |



**Figure 2** Caterpillar plot for Marshall Olkin Exponential distribution



**Figure 3** Survival curve of MOExp(b,  $\theta$ ) Model.

**Creation of data**

The data set dat.txt contains seven regressor variables namely, Age, Sex, Bun, Ca, Hb, PC and BJ. The design matrix X is extracted from survreg object M2 and is named as X.

```

sensor<-
c(1,0,1,1,1,0,1,0,1,1,1,1,1,1,0,1,0,1,0,0,1,1,1,1,0,1,1,1,1,1,1,1,
1,1,1,0,1,0,1,1,1,1,0,1,1,0)
dat=read.table("dat.txt",header=TRUE)
dat$sensor<-sensor
require(survival)
M2<-
survreg(Surv(time,Status)~Agec60+Sext+Bunc+Cac+Hbc+Pce
llsc+Protein,data=dat)
X<-model.matrix(M2)
J<-ncol(X)
n<-length(y)
y<-dat$time
zeros<- rep(0,n)
C<-10000
data<-list(n=n,y=y,X=X,zeros=zeros,sensor=sensor,C=C)

```

**Model definition**

The model definition consists of a series of relations inside a block delimited by curly brackets { and } and preceded by the keyword model. Here is the JAGS code specifying the model, using cat function to put it in the file model.txt. The specification of this model in JAGS using Poisson zero trick method is

```

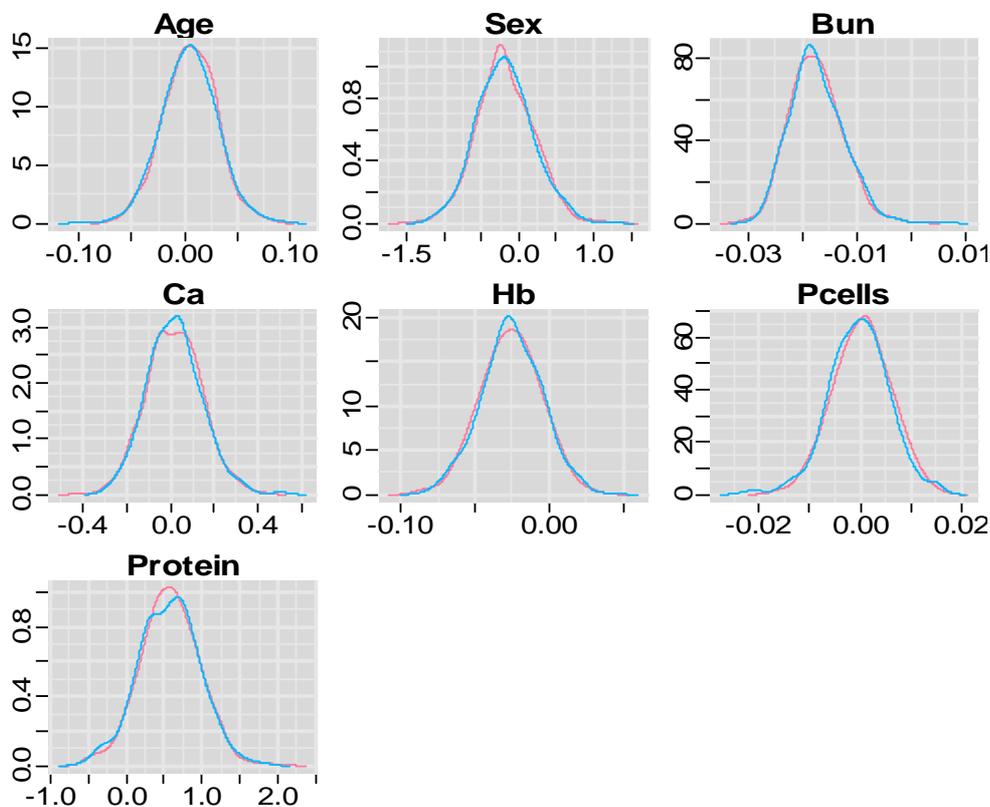
cat("model{
for(i in 1:n){
zeros[i]~dpois(phi[i])
phi[i]<- -l[i]+C
l[i]<-sensor[i]*(log(b/theta[i])-y[i]/theta[i]-2*log(b+(1-b)*(1-
exp(-y[i]/theta[i]))))+
(1-sensor[i]*(log(b-y[i]/theta[i]-log(b+(1-b)*(1-exp(-
y[i]/theta[i])))))
log(theta[i])<-inprod(X[i,],beta[i]) }
##Deviance dev<-sum(l[i])
Deviance<- -2*dev
Priors b~dunif(0,100)
beta[1]~dnorm(0,0.001)

```

**Table 5** Summary of JAGS simulations after being fitted to the MOExp(b,  $\theta$ ) model for the myeloma patients data

| Parameter | mu.vect | sd.vect | 2.5%    | 25%     | 50%    | 75%    | 97.5%  | Rhat | n.eff |
|-----------|---------|---------|---------|---------|--------|--------|--------|------|-------|
| Intercept | 3.130   | 0.525   | 2.290   | 2.760   | 3.071  | 3.430  | 4.323  | 1.00 | 2000  |
| Age       | 0.005   | 0.026   | -0.046  | -0.012  | 0.005  | 0.022  | 0.057  | 1.00 | 2000  |
| Sex       | -0.185  | 0.380   | -0.924  | -0.443  | -0.204 | 0.062  | 0.591  | 1.00 | 2000  |
| Bun       | -0.017  | 0.005   | -0.026  | -0.021  | -0.018 | -0.014 | -0.007 | 1.00 | 2000  |
| Ca        | 0.017   | 0.126   | -0.226  | -0.0700 | 0.018  | 0.098  | 0.283  | 1.00 | 2000  |
| Hb        | -0.025  | 0.020   | -0.0660 | -0.039  | -0.025 | 0.011  | 0.014  | 1.00 | 2000  |
| Pcells    | 0.001   | 0.006   | -0.012  | -0.004  | 0.001  | 0.004  | 0.011  | 1.00 | 350   |
| Protein   | 0.573   | 0.393   | -0.255  | 0.311   | 0.580  | 0.832  | 1.310  | 1.00 | 2000  |
| Deviance  | 314     | 4.533   | 307     | 311     | 314    | 317    | 325    | 1.00 | 870   |
| b         | 1.66    | 1.217   | 0.265   | 0.868   | 1.370  | 2.140  | 4.901  | 1.01 | 1700  |

denplot(Fit.jags,parms=2:8)



**Figure 4** Posterior densities plots of JAGS for the MOExp(b,  $\theta$ ) model, fit to the myeloma patients data

```
beta[7]~dnorm(0,0.001)
for(j in c(2,3,4,5,6,8)){
beta[j]~dnorm(0.0,.0001)
}
}", file="model.txt")
```

JAGS model is completed, the rest of the work happens in R. In the code below, we define those data elements that are reference in the JAGS model and initial values for the parameters. The JAGS model file, data objects and initial values are passed to the Fit.jags function.

```
inits<-
list(list(b=3,beta=coef(M2)*1.5),list(b=2.1,beta=coef(M2)*3))
Now, the above specified model is fitted with function Jags, an
output object is Fit.jags.
set.seed(12)
Fit.jags<-jags(data=data, inits=inits,
param=c("b","beta","Deviance"), n.chains=2,
n.iter=10000,progress.bar=NULL, model.file="model.txt")
print(Fit.jags)
```

## CONCLUSION

This paper shows that it is possible to implement Bayesian approach to survival data. For this, three different tools i.e; asymptotic analytic tools using LaplaceApproximation and simulation using R2jags and LaplacesDemon have been proposed. After viewing the results obtained by these different tools in different sections, it is clear that simulation tools provide better results in terms of standard error as compared to that obtained by asymptotic approximation.

## References

1. Collet, D. (2003). *Modelling Survival Data in Medical Research*, second edition. Chapman & Hall, London.
2. Gelman, A. and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, New York.

3. Gelman, A. (2008). Scaling Regression Inputs by Dividing by Two Standard Deviations. *Statistics in Medicine*, 27:2865-2873.
4. Ibrahim, J. G., Chen, M.H. and Sinha, D. (2001). *Bayesian Survival Analysis*.
5. Khan, N., Akhtar, M. T. and Khan, A. A. (2016). A Bayesian Approach to Survival Analysis of Inverse Gaussian Model with Laplace Approximation. *International Journal of Statistics and Applications*, 6(6): 391-398.
6. Marshall, A.W., Olkin, I. (1997). A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families, *Biometrika*, vol. 84, 641-652.
7. Mosteller, F. and Wallace, D. L. (1964). *Inference and Disputed Authorship: The Federalist*. Reading: Addison-Wesley.
8. Nadarajah, S. and Rocha, R. Newdistns(2016): An R Package for New Families of Distributions, *Journal of Statistical Software*, 69(10), 1-32, doi:10.18637/jss.v069.i10
9. Nocedal, J., and Wright, S. J. (1999). *Numerical Optimization*. Springer-Verlag.
10. R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, URL <http://www.R-project.org>.
11. Statisticat LLC (2015). *LaplacesDemon: Complete environment for Bayesian inference*. R package version 3.2.0, <http://www.bayesian-inference.com/software>.
12. Tanner, M. A. (1996). *Tools for Statistical Inference*. Springer-Verlag, New York.
13. Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81, 82-86.

### How to cite this article:

Najrullah Khan., Tanwir Akhtar and Athar Ali Khan.2017, Bayesian Analysis of Marshall and Olkin Family of Distributions. *Int J Recent Sci Res.* 8(7), pp. 18692-18699. DOI: <http://dx.doi.org/10.24327/ijrsr.2017.0807.0551>

\*\*\*\*\*