



ISSN: 0976-3031

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

CODEN: IJRSFP (USA)

International Journal of Recent Scientific Research
Vol. 8, Issue, 11, pp. 21837-21841, November, 2017

**International Journal of
Recent Scientific
Research**

DOI: 10.24327/IJRSR

Research Article

BAYESIAN ANALYSIS OF POISSON RELIABILITY MODEL WITH R AND JAGS

Tanwir Akhtar^{1*} and Athar Ali Khan²

¹Department of Public Health, College of Health Sciences, Saudi Electronic University, Jeddah-23442, SA

²Department of Statistics and Operations Research, Aligarh Muslim University, Aligarh-202002, India

DOI: <http://dx.doi.org/10.24327/ijrsr.2017.0811.1151>

ARTICLE INFO

Article History:

Received 15th August, 2017
Received in revised form 25th
September, 2017
Accepted 23rd October, 2017
Published online 28th November, 2017

Key Words:

Bayesian Analysis; Poisson Model;
Reliability Analysis; Simulation;
Posterior; R.

ABSTRACT

In many practical situations, reliability data are generated in the form of counts, also called failure counts, which record the number of times that a component fails in a specified period of time. Such data may arise because of limitation of the data capture system or the way the data are reported. For example, a system may keep track of the monthly number of failures and repair them. In this paper, an attempt has been made to model such type of data using Poisson distribution in Bayesian paradigm using two different prior distributions. Moreover, for the purpose of Bayesian modelling, two important techniques, that is, approximate *EM* algorithm and MCMC method are implemented using R and JAGS software packages, respectively. Approximate *EM* algorithm is implemented using R functions for approximating the posterior densities of model parameters. Whereas, JAGS is used to approximate the posterior parameters using Gibbs sampling and the Metropolis algorithm. R and JAGS code are developed and provided. A real data set is used for the purpose of illustrations.

Copyright © 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

In many practical situations, reliability data are generated in the form of counts, also called failure counts, which record the number of times that a component fails in a specified period of time. Such data may arise because of limitation of the data capture system or the way the data are reported. For example, a system may keep track of the monthly number of failures and repair them (Hamada *et al.*, 2008). Experimenters want to assess such types of reliability data. For modelling such type of data the Poisson model is used, and is appropriate when the probability of events occurring in disjoint time interval is independent and when the probability of events occurring in short time of interval is small (Hamada *et al.*, 2008). In this paper an attempt has been made to model failure count data in Bayesian paradigm using different prior distributions like Student-*t* prior and weakly informative conjugate prior distributions. In many cases it has been seen that non-Bayesian analysis of failure count data is not an easy task, whereas it can be implemented in principle in Bayesian paradigm, provided simulation tools are used (Akhtar and Khan, 2014a; Akhtar and Khan, 2014b, Khan *et al.*, 2016). Consequently, for the purpose of Bayesian modelling, two important techniques, that is, approximate *EM* algorithm and MCMC method are implemented using R (R Core Team, 2017) and JAGS (Plummer, 2003) software packages, respectively, from within

R. Approximate *EM* algorithm is implemented using arm package of R for approximating the posterior densities of model parameters. Whereas, JAGS approximates the posterior parameters using Gibbs sampling (Geman and Geman, 1984; Gelfand and Smith, 1990; Casella and George, 1992) and the Metropolis algorithm (Metropolis *et al.*, 1953) to generate a Markov chain by sampling from full conditional distributions. A real data set is used in subsequent analysis for the purpose of illustrations. Thus, the Bayesian analysis of Poisson reliability model has been made with the objectives: first to define a Bayesian model, that is, specification of likelihood and prior distribution, second to write down R and JAGS code for approximating the posterior densities using approximate *EM* algorithm and simulation tools, and finally to illustrate numeric as well as graphic summaries of the posterior densities.

The Poisson Reliability Model

The Model and Likelihood: Assume that we have a set of discrete reliability data generated in the form of counts y_i which follows Poisson distribution with mean θ . Algebraically,

$$y_i \sim \text{Poisson}(\theta t) = \frac{(\theta t)^{y_i} e^{-\theta t}}{y_i!}, i = 0, 1, 2, \dots, n, \theta > 0 \quad (1)$$

where y_i ($i = 0, 1, 2, \dots, n$) is the observed number of failures, θ is the mean number of failure per unit time, and t is the length of the specified time period. Note that equal mean and variance

*Corresponding author: Tanwir Akhtar

Department of Public Health, College of Health Sciences, Saudi Electronic University, Jeddah-23442, SA

(here θt) is the most limiting characteristic of the Poisson distribution (Hamada *et al.*, 2008).

For the Poisson model, the parameter of interest is the Poisson rate θ , which we wish to estimate. Therefore, the likelihood as a function of θ given the observed failure count y is given by,

$$p(y|\theta) = \prod_{i=1}^n \frac{e^{-\theta t_i} (\theta t_i)^{y_i}}{y_i!} = \frac{e^{-\theta \sum_{i=0}^n t_i} \prod_{i=0}^n (\theta t_i)^{y_i}}{y_i!} \quad (2)$$

or,

$$p(y|\theta) \sim e^{-\theta \sum_{i=0}^n t_i} \theta^{\sum_{i=0}^n y_i} \quad (3)$$

ignoring factors that do not depend on θ .

Bayesian Inference: In order to complete the Bayesian model, one must specify a prior distribution. Usually the target posterior distribution is not analytically tractable. In the past, intractability was avoided by using the conjugate prior distributions, which have the nice property of resulting to posteriors of the same distributional family (Ntzoufras, 2009). Furthermore, an advantage of conjugate prior is that it makes posterior calculation easy. Thus, to obtain posterior density for the Poisson model, let us consider a gamma prior for mean θ with parameter α and β having density function,

$$\theta \sim \text{Gamma}(\alpha, \beta)$$

$$p(\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta\theta} \propto \theta^{\alpha-1} e^{-\beta\theta}.$$

Then the resulting posterior distribution of θ by applying the Bayes' theorem is given as

$$p(\theta|y) \propto p(\theta)p(y|\theta) \propto \theta^{\alpha-1} e^{-\beta\theta} \times \theta^{\sum_{i=0}^n y_i} e^{-\theta \sum_{i=0}^n t_i}$$

$$\propto \theta^{\alpha+\sum_{i=0}^n y_i-1} e^{-\theta(\beta+\sum_{i=0}^n t_i)},$$

which implies that

$$\theta|y \sim \text{Gamma}\left(\alpha + \sum_{i=0}^n y_i, \beta + \sum_{i=0}^n t_i\right)$$

where $y = (y_1, y_2, y_3, \dots, y_n)$, and β can be interpreted as a prior sample size in contrast with the data sample size $\sum_{i=0}^n t_i$ where α is the prior number of failures in contrast with the observed number of failures $\sum_{i=0}^n y_i$. The posterior mean and variance are, respectively,

$$E(\theta|y) = \frac{\alpha + \sum_{i=0}^n y_i}{\beta + \sum_{i=0}^n t_i} = \frac{\alpha + n\bar{y}}{\beta + \sum_{i=0}^n t_i}$$

$$V(\theta|y) = \frac{\alpha + \sum_{i=0}^n y_i}{(\beta + \sum_{i=0}^n t_i)^2} = \frac{\alpha + n\bar{y}}{(\beta + \sum_{i=0}^n t_i)^2}$$

where \bar{y} is the sample mean (maximum likelihood estimator).

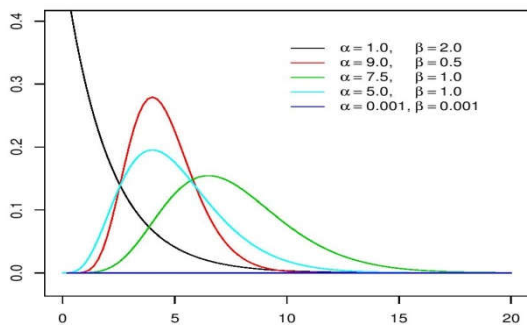


Figure 1 It is evident from the above plot that for shape $\alpha = 0.001$ and rate $\beta = 0.001$ the gamma distribution becomes almost uniform.

The usually selected low informative conjugate prior is a gamma distribution with low and equal prior parameters such as $\alpha = \beta = 10^{-3}$. This prior is convenient since its mean is equal to one while the variance is given by $1/\alpha$, which becomes large, expressing prior ignorance, for low value of α (Ntzoufras, 2009). This fact is evident from the Figure 1.

Implementation in Bayesian Paradigm

In order to implement the Bayesian analysis of Poisson model a super computer failure count data is taken from Hamada *et al.* (2008). Using these data different aspects of Bayesian analysis is discussed. For the purpose of fitting the model two different functions bayesglm and jags of R and JAGS, respectively, are used to estimate the parameter of interest, θ .

Supercomputer Failure Count Data

The supercomputer failure count data discussed about the monthly number of failures of shared memory processors (SMPs) components of the Los Alamos National Laboratory Blue Mountain Supercomputer. The supercomputer consists of 47 identical SMPs and Table 1 represents monthly number of failures for the first month of operation. The supercomputer engineers expect that there should be no more than 10 failures for each component in the first month of operation.

Table 1 Monthly number of failures for 47 supercomputer components (Hamada *et al.*, 2008).

1	5	1	4	2	3	1	3	6	4	4	4	2	3	2	2
4	5	5	2	5	3	2	2	3	1	1	2	5	1	4	1
1	1	2	1	3	2	5	3	5	2	5	1	1	5	2	

Analysis Using R

In the beginning, a Bayesian analysis of Poisson model using the bayesglm function from arm (Gelman and Su, 2014) package of R is conducted. This function is a Bayesian alteration of classical generalized linear model that uses an approximate EM algorithm to update the β s at each step using an augmented regression to represent the prior information (Gelman *et al.*, 2008). This function uses Student-*t* prior distribution for the coefficients. Since, in our case, model involves intercept only, therefore, the prior distribution for the intercept term is *t*-distribution with 1 degree of freedom (i.e., Cauchy distribution), and prior mean is 0 whereas prior scale is 10. The main arguments of the bayesglm function are:

```
bayesglm (formula, family = gaussian, data, ...)
```

The first argument formula requires a symbolic description of the model to be fitted. For example, if a model involves intercept only, the formula of the model will be of the form $y \sim 1$. The argument family requires a description of the error distribution and link function to be used in the model. The default error distribution is gaussian. The data argument needs an optional data frame, list, or environment containing the variables in the model. For more details see Gelman and Su (2014).

Data Creation

Firstly, for Bayesian fitting of Poisson model for supercomputer data with the function bayesglm, the data are created in R format as follows,
`n <- 47`

```
y<-c(1,5,1,4,2,3,1,3,6,4,4,4,2,3,2,2,4,5,
5,2,5,3,2,2,3,1,1,2,5,1,4,1,1,1,2,1,3,2,5,
3,5,2,5,1,1,5,2)
```

where n represents the total number of SMPs, and y represents the monthly number of failures for the first month of operations.

Model Specification

The distribution assumed for count data is a Poisson, which applies when counted things are distributed independently and randomly. The Poisson model has a single parameter, the expected count θ , that is often called intensity (Kery, 2010) and, in case of supercomputer data, represents monthly failure rate or mean number of failures. In contrast to the normal, the Poisson variance is not a free parameter but is equal to the mean, θ .

Thus, for supercomputer failure count data, where the response (y_1, y_2, \dots, y_n) is the monthly number of failures recorded for the SMPs, the model for these responses is

$$y_i \sim \text{Poisson}(\theta), \quad i = 0, 1, 2, \dots, 47.$$

Here, count response y_i for i th SMP is distributed as a Poisson random variable with mean number of failure θ , for the first month of operation (i.e., $t_i = 1$ month). Moreover, for Bayesian analysis of Poisson model the function `bayesglm` uses default *log-link* function, that is,

$$\log(\theta) = \text{Linear predictor} = \text{Intercept}$$

In this case, intercept is the only term in linear predictor. Therefore, after fitting the model, to get the results in original metric the exponential function `exp` is used to transform the results.

Model Fitting

Finally, to fit the above defined model, the function `bayesglm` is called to perform the analysis and put its results into an object called `Fit`. The `display` function of `arm` package is used to print the summary of results. This generic function with its `detail=TRUE` argument gives a clean printout of the fitted object, focusing on the most pertinent pieces of information including *p-value* or *z-value*.

```
Fit<-bayesglm(y~1, family=poisson
(link="log"))
display(Fit, detail=TRUE)
```

In order to get the posterior simulations of the model parameter `theta`, a generic function of `arm` package called `sim` is used. To get the clear picture of parameter estimates, 1000 (`n.sim=1000`) independent draws are made. Moreover, to convert the results in original metric the function `exp` is used, and its output is reported and discussed in the next section.

```
set.seed(123)
## Simulation
Fit.sim<-sim(Fit,n.sim=1000)
coef.Fit.sim<-coef(Fit.sim)
exp(coef.Fit.sim)
## Estimate
apply(exp(coef.Fit.sim),2,mean)
## Standard Deviation
apply(exp(coef.Fit.sim),2,sd)
```

```
## Quantiles
quantile(exp(coef.Fit.sim),prob=c(0.025,0.
25,0.5,0.75,0.975))
```

Summarizing Output

Posterior summaries and densities after being fitted the model with `bayesglm` are provided in Table 2. From this table, it is observed that, the estimate of `theta` is 2.82 with standard deviation 0.24, and that the 95% credible interval is [2.37, 3.35], which is statistically significant. The plot for the posterior density of `theta` can be seen in Figure 1.

Table 2 From this output it is evident that posterior mean and median are very close, which is an indication of symmetric posterior density

Parameter	Mean	SD	2.5%	25%	Median	75%	97.5%
Theta	2.82	0.24	2.37	2.65	2.80	2.97	3.35

Analysis Using JAGS

Let us consider the Bayesian analysis of the same failure count data with JAGS using its interface of R that is `R2jags` (Su and Yajima, 2014) package of R, which includes the posterior simulation and convergence diagnostic of the model. For modelling of these data in JAGS, one must specify the model to run, and to load data which is created in a separate file and the initial values of the model parameters for a specified number of Markov chains (Akhtar and Khan, 2017; Khan *et al.*, 2017). The `R2jags` package makes use of this feature and provides convenient functions to call JAGS directly from within R. Furthermore, it is possible to work with the results after importing them back into R again, for example, to create a posterior predictive simulations, or more generally, graphical displays of data and posterior simulations.

Data Creation

The first part of modelling in JAGS is data definition, which must be defined in a list containing the name of each vector. In this case, the monthly number of failures are the response variable y_i ($i = 0, 1, 2, \dots, 47$). These data can be created in R format as:

```
n <- 47
y <-
c(1,5,1,4,2,3,1,3,6,4,4,4,2,3,2,2,4,5,5,2,
5,3,2,2,3,1,1,2,5,1,4,1,1,1,2,1,3,2,5,3,5,
2,5,1,1,5,2)
jags.data <- list("n","y")
```

Here n is the total number of supercomputer components (SMPs), y is the response values of monthly number of failures, and these are combined in a list with object `jags.data`.

Model Specification

For modelling these data, the monthly number of failures y_i is assumed to follow a Poisson distribution

$$y_i \sim \text{Poisson}(\theta), \quad i = 0, 1, 2, \dots, 47,$$

where θ is the individual monthly failure rate for the SMPs. In this case, the monthly failure rate θ is the unknown model parameter which we want to estimate. It is expected that there should be no more than 10 failures for each component. To represent this, the prior information for the parameter θ is

assumed to be a conjugate gamma prior distribution with a mean of 5, that is,

$$\theta \sim \text{Gamma}(5, 1),$$

It is observed that, the probability that θ , exceeds 10 for Gamma(5, 1) prior is 0.03, which is very small (Hamada *et al.*, 2008).

Thus, the specification of above defined model in JAGS language must be put in a separate file which is then read by JAGS. When working in R, this is most conveniently done using the `cat` function of R which behaves pretty much like paste with the exception that the results is not a character object but directly written to a file we specify (Akhtar and Khan, 2017; Khan *et al.*, 2017). Here is the JAGS code specifying the model using `cat` function to put in the file `model2.jags`:

```
cat("model{
for(i in 1:n){
y[i]~dpois(theta)
}
theta~dgamma(5,1)
}", file="model2.jags.txt")
```

This model allows each observed response y to follow a Poisson distribution with mean θ , which is drawn from a gamma distribution with shape 5 and rate 1, that is, mean value 5.

Initial Values

The starting values used to initialize the chain are simply called the *initial values*. Usually it is necessary to specify a starting value for the chains to start the MCMC simulation. In most cases, JAGS will however be able to generate the initial values itself. In order to be able to monitor convergence, we will normally run several chains for each parameter. The starting value for the chains is a named list, names are the parameters used in the model. Each element of the list is itself a list of starting values for the JAGS model, or a function creating (possible random) initial values. In this case, there is only one parameter, called θ , in the model:

```
inits <- function(){list(theta =
rlnorm(1))}
```

Model Fitting

Using the objects `jags.data`, `inits`, and `model2.jags`, we can now compile the model and run MCMC simulation to get estimate for θ by calling JAGS. For MCMC simulation 3 chains (`n.chain=3`) each with 5000 iterations (`n.iter=5000`) has been run, and parameter θ is monitored. Its results are assigned with object `Fit`, and can conveniently be printed by `print(Fit)`, which prints details summary of the results and are summarized in the next section.

```
Fit <- jags(jags.data, inits,
parameter=c("theta"), n.iter=5000,
n.chain=3, model.file="model2.jags.txt",)
print(Fit)
```

Summarizing Output

The output of the R function `jags` is a list which includes several components, most notable are the summary of the inference and convergence, and a list containing the simulation

draws of all the saved parameters. In this case, the `jags` call is assigned to the R object `Fit`, and so typing `print(Fit)` from the R console will display the summary of the fitted model shown below. The `print` method displays information on the mean, standard deviation, 95% credible interval (CI) estimates, the effective sample size, and potential scale reduction factor \hat{R} of the Brook-Gelman-Rubin (BGR) statistics (Gelman and Rubin, 1992; Brooks and Gelman, 1998). The BGR statistics is an analysis of variance (ANOVA)-type diagnostic that compares within- and among-chain variance (Kery, 2010). Values around 1 indicate good convergence, with 1.1 considered as acceptable limit (Gelman and Hill, 2007).

Table 3 Posterior summary of JAGS simulation after being fitted to the Poisson model for the supercomputer failure count data.

Parameter	Mean	SD	2.5%	Median	97.5%	Rhat	n.eff
theta	2.86	0.249	2.39	2.85	3.36	1.001	3800
deviance	171.61	1.507	170.57	171.04	175.77	1.001	3800
pD		1.1		DIC		172.7	

The first five columns of numbers of Table 3 give inference for model parameter. In this case, the posterior estimate for mean θ is 2.86 with standard deviation 0.249 and 95% credible interval is [2.39, 3.36], which is statistically significant. The median estimate is 2.85 and very close to mean estimate. At the bottom, `pD` shows the estimated effective number of parameters in the model, and `DIC`, the deviance information criterion, an estimate of predictive error. Finally, consider the right most columns of the output, where `Rhat` gives information about convergence of the algorithm. At convergence, the number at this column should be equal to 1. In this case, `Rhat` is very close to 1, indication of well mixing of the three chains and thus good convergence. The final column, `n.eff` is the effective sample size of the simulations.

Additionally, to see the complete picture of the results, a plot can be generated by `plot(Fit)`, and the resulting plot is depicted in Figure 1. In this plot, the left column shows a quick summary of inference and convergence, that is, `Rhat`. The right column shows inference for the model parameter θ . Finally, the posterior density plot for parameter θ is depicted in Figure 2.

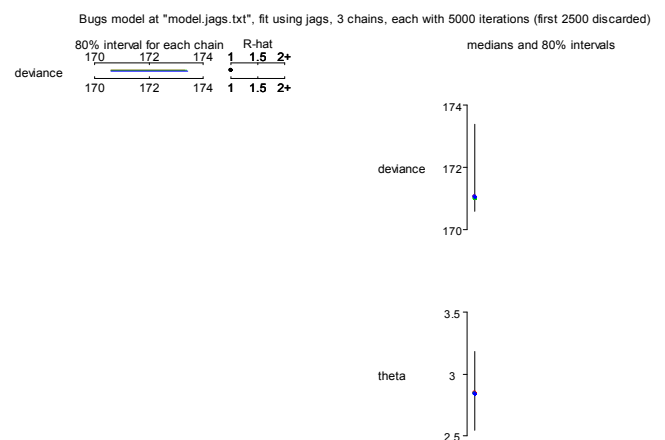


Figure 2 Graphical summary plot of JAGS for the Poisson model, fit to the supercomputer failure count data. `R-hat` is near to one for parameter θ , indicating good convergence.

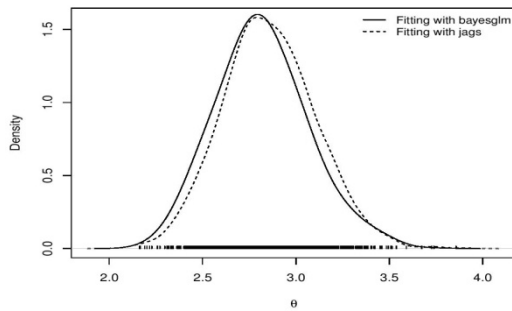


Figure 3 Posterior density plot of the model parameter theta. Results from different methods have different styles. The closeness of the two approaches look evident in the graphics.

DISCUSSION AND CONCLUSION

Component reliability is the foundation of reliability assessment and refers to the reliability of single component. In this paper, Bayesian approach is used to model failure count component reliability data for intercept model with Poisson distribution. The log-link function is used to relate the model parameter and intercept term for linear Poisson model. Two different prior distributions, that is, Student- t and gamma prior distributions are used and implemented using R and JAGS software packages, respectively. For modelling of failure count data, the complete R and JAGS code are written and provided with detailed description. It is observed that the results obtained from approximate EM algorithm and simulation method using different software packages are very close to each other. The benefits of approximate EM algorithm and simulation method seem clear in the plot of posterior densities. Moreover, it is evident from the summaries of results that the Bayesian approach based on different priors is simpler to implement than the frequentist approach. Finally, the wealth of information provided in these numeric as well as graphic summaries is not possible in classical framework.

References

1. Akhtar, M. T. and Khan, A. A. (2014). Bayesian Analysis of Generalized Log-Burr Family with R. *SpringerPlus* 2014, 3:185.
2. Akhtar, M. T. and Khan, A. A. (2014). Log-logistic Distribution as a Reliability Model: A Bayesian Analysis. *American Journal of Mathematics and Statistics*, 4(3): 162-170.
3. Akhtar, M. T. and Khan, A. A. (2017). Bayesian Reliability Analysis of Binomial Model -- Application to Success/ Failure Data. *Journal of Modern Applied Statistical Methods* (accepted).
4. Brooks, S. P. and Gelman, A. (1998). General Method for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, 7(4): 434-455.
5. Casella, G. and George, E. (1992). Explaining the Gibbs Sampler. *The American Statistician*, 46: 167-174.
6. Gelfand, A. and Smith, A. (1990). Sampling-based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, 85: 398-409.
7. Gelman, A. and Rubin, D. B. (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4): 457-511.
8. Gelman, A. and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, New York.
9. Gelman, A., Jakulin, A., Pittau, M. G., and Su, Y. (2008). A Weekly Informative Default Prior Distribution for Logistic and Other Regression Models. *The Annals of Applied Statistics*, 2(4): 1360-1383.
10. Gelman, A. and Su, Y. (2014). *arm: Data Analysis Using Regression and Multilevel/Hierarchical Models*. R package version 1.7-07, URL <http://CRAN.R-project/package=arm>.
11. Geman, S. And Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6: 721-741.
12. Hamada, M. S., Wilson, A. G., Reese, C. S., and Martz, H. F. (2008). *Bayesian Reliability*. Springer, New York.
13. Kery, M. (2010). Introduction to WinBUGS for Ecologist: A Bayesian Approach to Regression, ANOVA, Mixed Models, and Related Analyses. Academic Press, USA.
14. Khan, N, Akhtar, M. T., and Khan, A. A. (2016). Bayesian Analysis of Marshal and Olkin Family of Distributions. *International Journal of Recent Scientific Research*, 8(7): 18692-18699.
15. Khan, N, Akhtar, M. T., and Khan, A. A. (2017). A Bayesian Approach to Survival Analysis of Inverse Gaussian Model with Laplace Approximation. *International Journal of Statistics and Applications*, 6 (6): 391-398.
16. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, A. (1953). Equation of State Calculations by Fast Computing Machine. *Journal of Chemical Physics*, 21: 1087-1092.
17. Ntzoufras, I. (2009). *Bayesian Modeling Using WinBUGS*. John Wiley & Sons, New York.
18. Plummer, M. (2003). JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling. *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003), March 2002, Vienna, Austria*. ISSN 1609-395X.
19. R Core Team (2017). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
20. Su, Y. And Yajima, M. (2014). *R2jags: A Package for Running JAGS from R*. R package version 0.04-01, URL <http://CRAN.R-project.org/package=R2jags>.

How to cite this article:

Tanwir Akhtar and Athar Ali Khan.2017, Bayesian Analysis of Poisson Reliability Model with R and JAGS. *Int J Recent Sci Res*. 8(11), pp. 21837-21841. DOI: <http://dx.doi.org/10.24327/ijrsr.2017.0811.1151>
