Bayesian Analysis of Normal Model with Stan and Inla

[1]Firdoos yousuf,[2]Athar Ali Khan
[1]Research Scholar, [2]Professor
[1] Department of Statistics & Operations Research,
[1] Aligarh Muslim University, Aligarh, Uttar Pradesh, India.

Abstract : The Normal model is very common and the most widely used model in the field of Statistics. In this paper, an endeavour has been made to fit the Bayesian inference procedures for the Normal model. The implementation is made through R, Stan and INLA software packages. INLA is based on approximating the posterior marginal distributions of the model parameters by means of different Laplace approximations. Codes that have been created in R and Stan to implement the Bayesian approach using both optimization and simulation tools. Stan is a high-level language written in a C++ library for Bayesian modelling. Real data sets are used for illustrations. Moreover, parallel simulation tools are also implemented and furthermore actualized with a broad utilization of rstan. Its comparison has been made with the implementation using R-INLA software package.

Keywords **- Bayesian Inference, Posterior, Simulation, RStan, R-INLA, R, HMC.**

I. INTRODUCTION

Bayesian modelling is a part of statistical modelling in which the parameter itself considered as a random variable and data are constant, and in this modelling, we use prior information along with the observed data to inform our views about the parameter (Gelman et al., 2014). In a frequentist approach, the parameters θ considered as constant terms, and the aim is to study the distribution of the data given θ, through the likelihood of the sample. The three input constituents of Bayesian modelling are the likelihood function, prior distribution and posterior distribution. In this paper, an endeavour has been made to outline how the Bayesian approach proceeds to fit Normal model for econometric data using Stan. The equipment and methods used in this paper are in the Bayesian environment, which is implemented using rstan package. Bayesian inference is based on the Bayes rule which provides a sensitive technique for updating our beliefs in the light of new information.The Bayes rule states that posterior distribution is the combination of prior and data information. The prior distribution is significant in Bayesian inference since it influences the posterior. When no information is available, we need to specify a prior which will not influence the posterior distribution. Such priors are called weakly-informative, such as Normal, Gamma and half-Cauchy prior. The posterior distribution contains all the information required for Bayesian inference and the objective is to compute the numeric summaries of it via integration. 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 an extensive class of problems, this is the easiest way to get reliable results (Gelman et al, 2014). Gibbs sampling, Hamiltonian Monte Carlo (HMC) and Metropolis-Hastings algorithm are the MCMC techniques which provide difficult computational tasks quite feasible. HMC is a dominant sampling algorithm employed by several probabilistic programming languages. Probabilistic programming languages Stan and PyMC rely on HMC and its variants to carry out posterior inferences automatically and have made HMC a benchmark tool for applied Bayesian modelling (Stan Development Team, 2018). To make computation easier, software such as R, Stan (full Bayesian inference using the No-U-Turn sampler (NUTS), a modification of Hamiltonian Monte Carlo (HMC)) are used. Bayesian analysis of proposal appropriation has been made with the following objectives:

1. To define a Bayesian model, that is, specification of likelihood and prior distribution.
2. To write down the R code for approximating posterior densities with Stan.
3. To illustrate numeric as well as graphic summaries of the posterior densities.

A feasible alternative to MCMC methods able to reduce the computational costs of Bayesian inference is the Integrated Nested Laplace Approximation (INLA) algorithm. The INLA algorithm, proposed by Rue et al. (2009), is a deterministic algorithm for Bayesian inference. INLA specially designed for latent Gaussian models, an extensive and flexible class of models ranging from (generalised) linear mixed to spatial and spatiotemporal models. Compared to MCMC, it provides accurate results in shorter computing time.

1.1 The Gaussion or Normal Model

The Gaussian, also known as the Normal model, is a widely used model for the distribution of continuous variables and as a limiting case of the binomial distribution and applied it to problems arising in the game of chance. It is very common in the field of statistics. Whenever you measure things like people's height, weight, salary, the graph of the results is very often a Normal curve. Most of the distributions occurring in practice, e.g., Binomial, Poisson, Hypergeometric distributions, etc., can be approximated by the Normal distribution. It has two parameters, usually denoted by μ and σ^2, which are its mean and variance. The Normal or Gaussian probability density function (pdf) may be expressed as

$$f(y;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}, \quad -\infty < y < \infty$$

Where y express the experimental measurements of the random variable y, μ is its mean, and σ^2 is the scale parameter, or variance. The probability function, symbolize as $f(.)$, tells us that the data y is generated on the basis of the parameters μ and σ.

A statistical model, whether it is based on a classical or Bayesian tactic, is framed to determine the values of parameters based on the given data. This is just the overturn of how a probability density function (pdf) is understood. The function that reverses the affiliation of what is to be generated or understood in a pdf is called a likelihood function. The likelihood function determines which parameter values build the data being modelled most likely – hence the name likelihood. It is like to the pdf except that the left-hand side of the equation now appears as (Hilbe et al., 2017)

$$L(\mu,\sigma^2;y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}},$$

Signifying that the mean and variance parameters are to be determined on the basis of the data. The log-likelihood function L for the Normal model may be expressed as follows:

$$L(\mu,\sigma^2;y) = \sum_{i=0}^{N}\left[-\frac{(y_i-\mu_i)^2}{2\sigma^2} - \frac{1}{2}\ln\left(2\pi\sigma^2\right)\right]$$

In the event of a Normal linear model with a single predictor, the response variable y is normally distributed and the association between the response variable y and the explanatory variable x usually takes the form

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon, \quad \varepsilon : N(0,\sigma^2)$$

1.2 Bayesian Inference

Gelman et al., (2014) crack applied Bayesian modelling 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 modelled 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.

Usually, this cycle will be repeated until a sufficient fit is achieved in the third step. Stan computerizes 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:

The parameters θ related to the distribution of the data, are considered as random variables. Their distribution is called the prior distribution and is denoted by $p(\theta)$.

Likelihood:

Likelihood function for variables are related in full probability model and is denoted by $p(y \mid \theta)$

Posterior distribution:

It is the joint posterior distribution that expresses uncertainty about parameter θ after considering about the prior and the data, as in equation

$$p(\theta \mid y) = p(y \mid \theta)p(\theta)$$

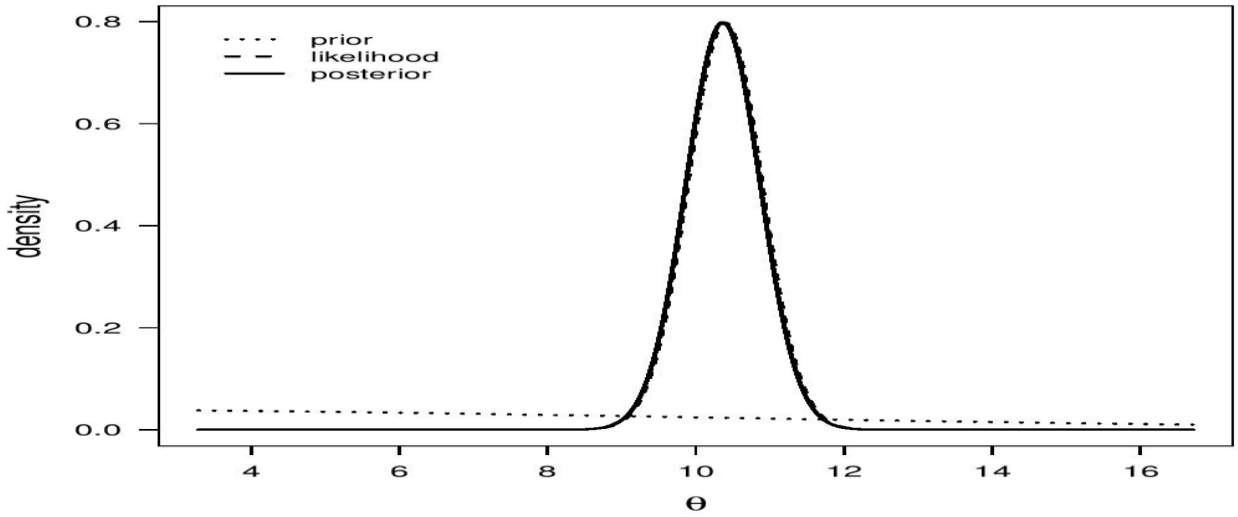

Figure 1: Prior distribution, likelihood, and posterior distribution of the mean θ.

1.3 The Prior Distributions

In the Bayesian paradigm, it is critical to indicate prior information with the value of the specified parameter or information which are obtained before analyzing the experimental data by using a probability distribution function which is called the prior probability distribution (or the prior). In this paper, we use two types of priors which are half-Cauchy prior and Normal prior. The simplest of all priors is a conjugate prior which makes posterior calculations easy. Also, a conjugate prior distribution for an unknown parameter leads to a posterior distribution for which there is a simple formula for posterior means and variances. Akhtar and Khan (2014) use the half-Cauchy distribution with scale parameter $\alpha = 25$ is used as a non-informative prior distribution.

First, the probability density function of half-Cauchy distribution with scale parameter α is given by

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

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, weakly informative prior distribution for a scale parameter. At this scale, $\alpha = 25$, the density of half-Cauchy is nearly at but not completely (see Figure 2), prior distributions that are not totally flat afford 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 non-informative prior distribution for scale parameter, though, this model creates a problem for scale parameters near zero, Gelman and Hill (2007) recommend that, the uniform, or if more information is essential the half-Cauchy is a superior choice.

Next, the Normal (or Gaussian), each parameters is assigned a weak information Gaussian prior probability distribution. In this paper, we use the parameters β_j independently in the Normal distribution with mean=0 and standard deviation=1000, that is, $\beta_j : N(0,1000)$, for this, we obtain a flat prior. **Figure 2**, we see that the large variance indicates a lot of uncertainty about each parameter and hence, a weak informative distribution.

Figure 2: Prior distributions.

II. DATA SET: CPS1985

The application considered here is the estimation of a wage equation in the semilogarithmic form based on data taken from Berndt (1991). They represent a random subsample of cross-section data originating from the May 1985 Current Population Survey (CPS) by the US Census Bureau. This data is available in AER package in R software. It is in the form of a data frame containing 534 observations on 11 variables. After loading the data set CPS1985 from the package AER, we first rename it for convenience:

```
data("CPS1985", package = "AER")
cps <- CPS1985
```

For cps, a wage equation is estimated with log(wage) as the dependent variable and education and experience (both in number of years) as regressors. For experience, a quadratic term is included as well (Kleiber and Zeileis, 2008).

Our model of interest is

$$log(wage) = \beta_1 + \beta_2 experience + \beta_3 experience^2 + \beta_4 education + \varepsilon$$

This model can be fitted in R using the function `lm`

```
M1<- lm(log(wage) ~ experience + I(experience^2) + education, data = cps)
```

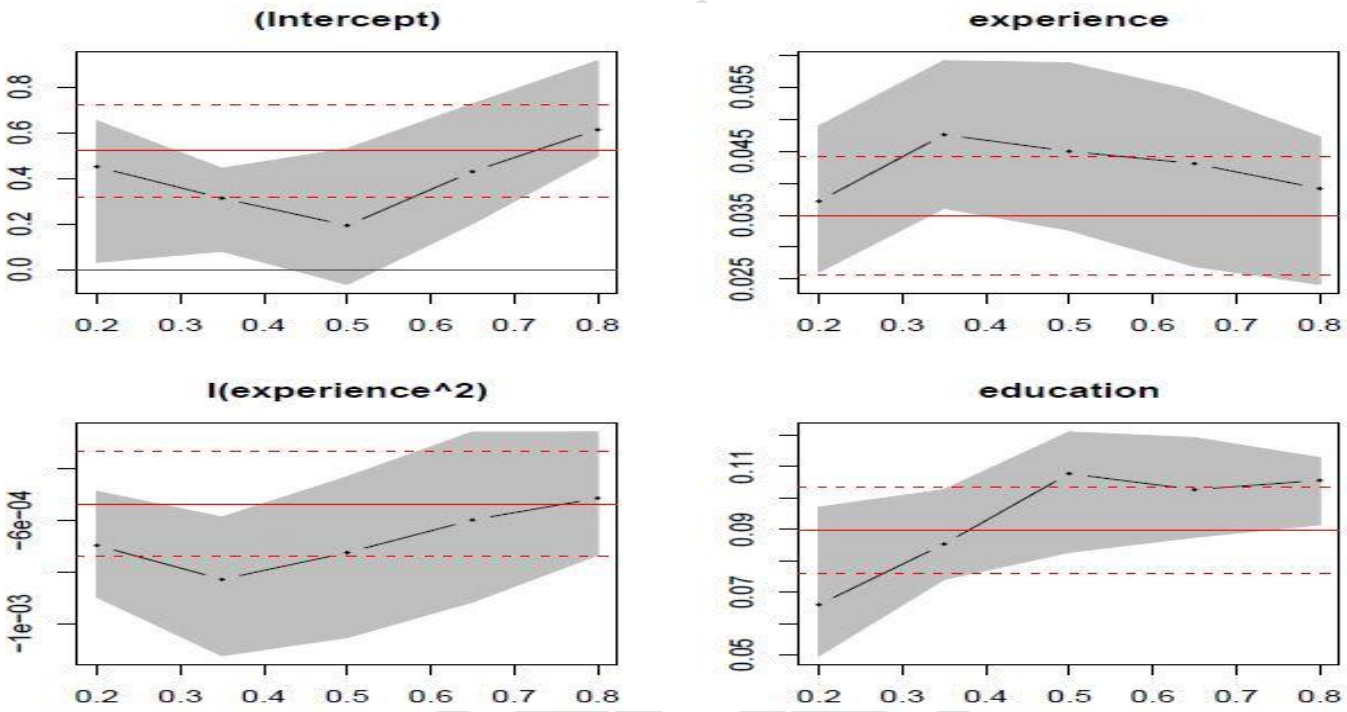

Figure 3: Coefficients of quantile regression for varying quantiles, with confidence bands (gray) and least-squares estimate (red).

2.1 Bayesian Fitting with Stan

Stan is a high-level language written in a C++ library for Bayesian modelling and (Carpenter et al., 2017) is a new Bayesian software program for the inference that primarily uses the No-U-Turn sampler (NUTS), which is a type of Hamiltonian Monte Carlo simulation (Hoffman and Gelman, 2014) and optimization-based point estimation. Stan works particularly well for hierarchical models (Betancourt and Girolami, 2015). In more difficult settings, though, HMC to be faster and more reliable than basic Markov chain simulation, Gibbs sampler and the Metropolis algorithm since they explore the posterior parameter space more efficiently. Accordingly, Stan is significantly more efficient than the traditional Bayesian software programs. Though, the main function in the rstan package is the stan, which calls the Stan software program to estimate a specified statistical model. Stan can be used via the R interface rstan. Stan is automatically installed when the R package rstan is installed. Stan requires that a C++ compiler is installed on the computer. Therefore, before downloading rstan we have to install a C++ compiler (e.g., Rtools). A comprehensive manual introduces the Stan language (Stan Development Team, 2018).

A Stan program defines a statistical model through a conditional probability function $p(\theta \mid y, x)$, where θ is a sequence of modelled unknown values (e.g., model parameters, latent variables, missing data, future predictions), y is a sequence of modelled known values, and x is a sequence of unmodeled predictors and constants (e.g., sizes, hyperparameters).

Stan programs consist of variable type declarations and statements. Variable types include constrained and unconstrained integer, scalar, vector, and matrix types, as well as (multidimensional) arrays of other types. Variables are declared in blocks corresponding to the variable's use: data, transformed data, parameters, transformed parameters, model, and generated quantities. The transformed data, transformed parameters, and generated quantities blocks contain statements defining the variables declared in their blocks.

Creation of data

```
X=model.matrix(M1)
M=ncol(X)
N=nrow(X)
y=log(CPS1985$wage)
```

In this case X is a model matrix which can be extracted by the function model.matrix() from an lm() fitted object. The response variable y is a vector of length N and M denotes the number of columns of the model matrix.

Model Specification

Now we will examine the posterior estimates of the parameters when the Normal model is fitted to the above mentioned information (data).

```
library(rstan)
stan_code="
data{
int<lower=0> N; // number of observations
vector[N] y; // observed times
int<lower=0> M; // number of covariate
matrix[N,M] X; // matrix of covariates with N rows and M columns
}
parameters{
vector[M] beta; // coefficients in the linear predictor (including intercept)
 real<lower=0> sigma;
}
 model {
vector[N] mu;
mu=X*beta;
//priors
beta~normal(0,5);
sigma~cauchy(0,5);
//likelihood
y~normal(mu,sigma); // likelihood function
}"
dat=list(N=N,M=M,X=X,y=y)
             //regression coefficient with log(y) as a guess to initialize
                        beta1=solve(crossprod(X),crossprod(X,y))
//convert matrix to a vector
beta1=c(beta1)
```

The model block contains the model specification. Unlike BUGS, Stan functions can handle vectors. Therefore, we do not have to loop through all observations. Here, we use a Cauchy distribution as a prior distribution for sigma. This distribution can have negative values, but because we defined the lower limit of sigma to be 0 in the parameters block, the prior distribution actually used in the model is a truncated Cauchy distribution (truncated at zero).

Now we run Stan with 3 chains for 5000 iterations and display the results numerically and graphically. The defaults are 5000 for iter and warmup is set to iter/2, which gives you 2500 warmup samples and 2500 real samples to use for inference. We use the defaults to make sure that the chain is get started good.

```
M2=stan(model_code=stan_code,data=dat,iter=5000, chains=3)
print(M2,digits=3)
```

Summarizing Output:

The function stan approximates the posterior density of the fitted model, and posterior summaries can be seen in Table 1, contains summaries for all chains merged and individual chains, respectively. According to 95% credible intervals, beta1, beta2, beta4 and sigma are found to be statistically significant. Hence they are appropriate variables for modelling cps data. Also table 1 provides the summaries of quantiles, means, standard deviations (sd), n_eff, and Rhat. For each parameter, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat=1). For the summary of all chains merged, Monte Carlo standard errors(se_mean) are also reported.

Table 1: Summary of the simulated results using stan 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

Parameter	Mean	Se_mean	SD	2.5%	50%	97.5%	n_eff	Rhat
beta[1]	0.520	0.002	0.124	0.281	0.519	0.767	2805	1
beta[2]	0.035	0.000	0.006	0.024	0.035	0.046	4036	1
beta[3]	-0.001	0.000	0.000	-0.001	-0.001	0.000	4513	1
beta[4]	0.090	0.000	0.008	0.073	0.090	0.106	3017	1
sigma	0.463	0.000	0.014	0.437	0.463	0.492	4378	1
lp	144.126	0.030	1.569	140.245	144.446	146.204	2664	1

The assortment of appropriate regressor variables can also be done by using a caterpillar plot. Caterpillar plots are well-liked plots in Bayesian inference for summarizing the quantiles of posterior samples. **Figure 4**, we see 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 coloured areas are used to specify which range the value of the split Rhat statistic is in. This may be used to create a caterpillar plot of posterior samples. In MCMC estimation, it is essential to thoroughly assess convergence as it in **"Fig. 5"** the rstan contains a specialized function to visualise the model output and assess convergence. Generally, Bayesian specialist like to verify whether a model estimation converges well or not, we want to plot a distribution (histogram) of each estimated parameter. The smooth way is using library coda.

```
stan_plot(M2,pars = c("beta"))+ggtitle("Quantile summary plot of posterior samples")
stan_ac(M2,"beta")
traceplot(M2,"beta")
```

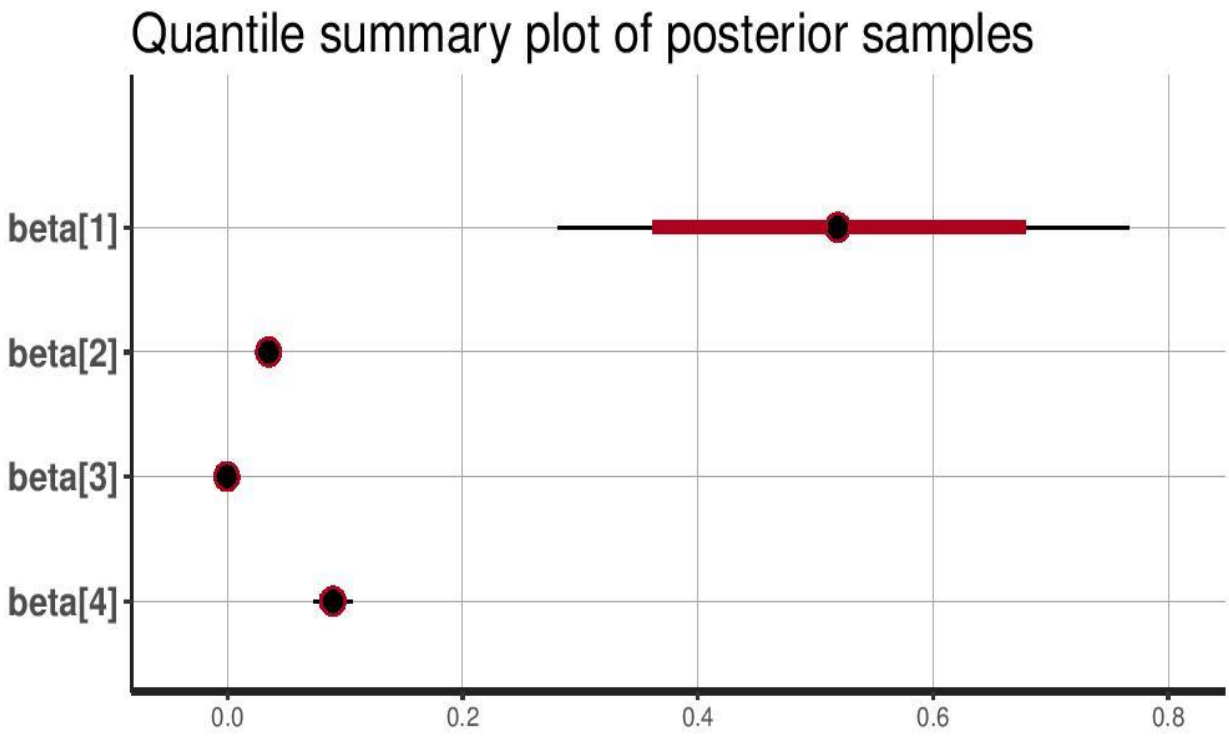

Figure 4: Caterpillar plot for Normal model.

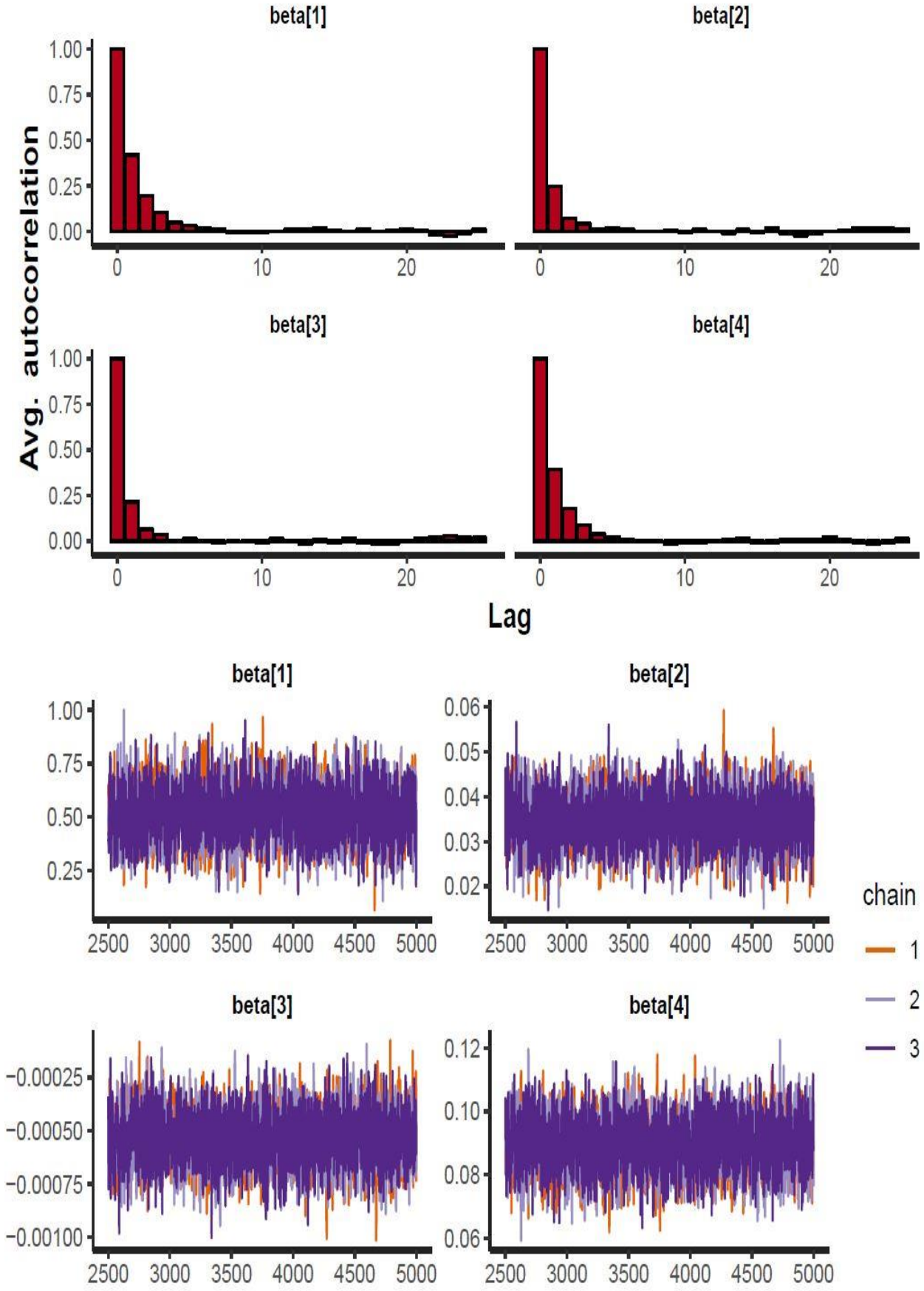

Figure 5: Checking model convergence using rstan, through inspection of the traceplots or the autocorrelation plot.

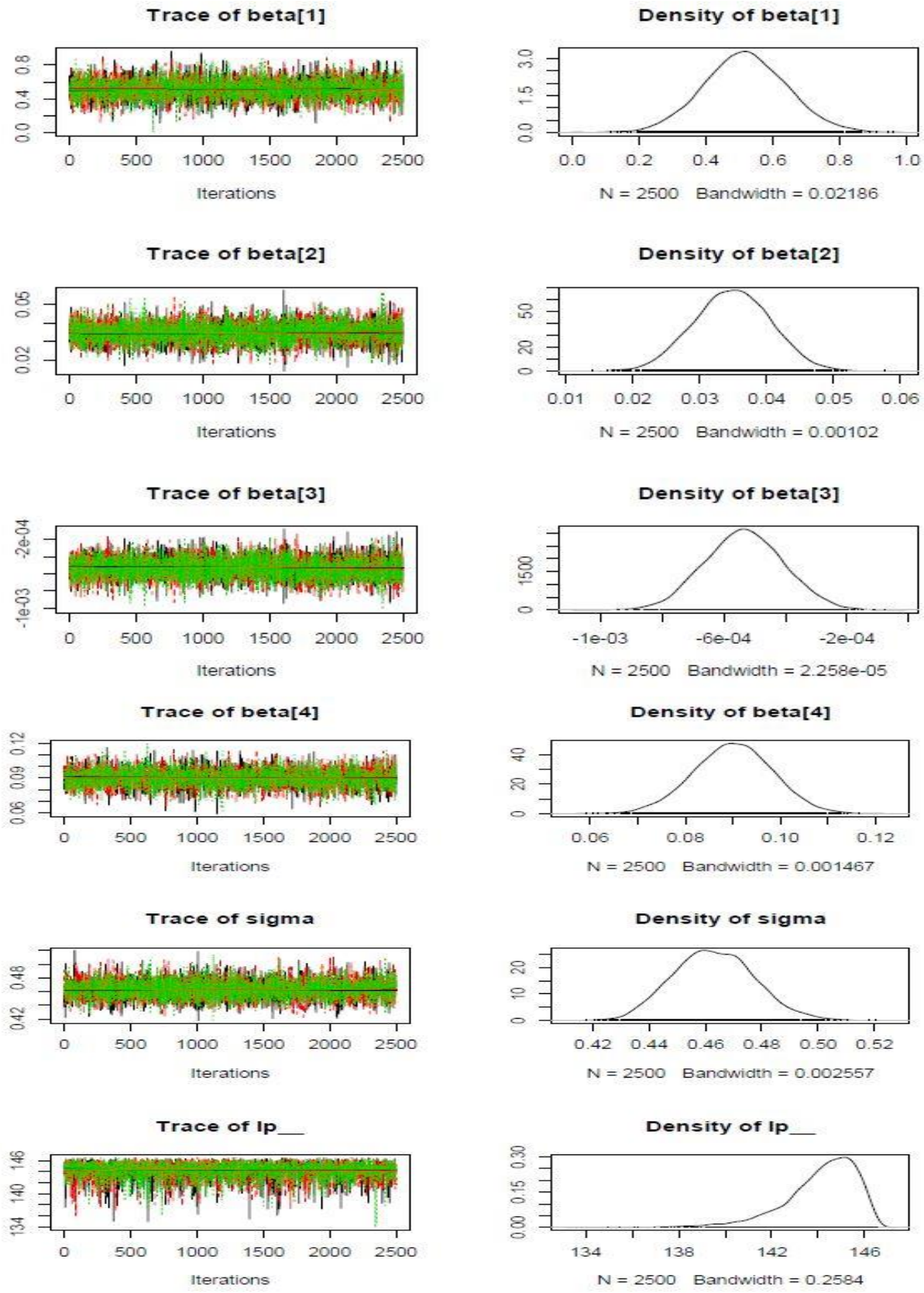

Figure 6: Checking model convergence using coda, through inspection of the simulated posterior density plots with trace plots of regressor variables obtained by HMC.

2.1 Bayesian Fitting with Inla

The Integrated Nested Laplace Approximation (INLA) is based on approximating the posterior marginal distributions of the model parameters by means of different Laplace approximations. The INLA methodology was first introduced by Rue et al. (2009), following by developments in Martins et al. (2013), and is most lately reviewed in Rue et al. (2017). It is a determinist methodology to approximate Bayesian inference for Latent Gaussian models (LGMs). The R-INLA package can be used for quick and reliable Bayesian inference in practical applications. The INLA method is an approximation that could be made more accurate at the expense of longer computation. The approximate posterior marginals obtained from the INLA method can then be used to compute summary statistics of interest, such as posterior means, variances and quantiles. As a by-product of the main computations,INLA also computes other quantities of interest like deviance information criterion (DIC), marginal likelihoods, etc., which are beneficial to compare and validate models.

In this section, cps data is used to fit the model. The model which is defined in Section II by using the function lm() is reproduced in R-INLA through the command

```
formula<-log(wage) ~ experience + I(experience^2) + education
```

Finally, we run the INLA algorithm using the inla function as follows:

```
M3<-inla(formula,family="gaussian",data=cps)
```

The inla function returns an object, here named M3, of class inla. This medium is a list containing many objects which can explore with names (M3). For a general summary of the results using

```
round(M3$summary.fixed[,1:5],3)
```

Table 2: Summary of the simulated results using inla function

Coefficients	Mean	SD	0.025quant	0.5quant	0.975quant
(Intercept)	0.520	0.124	0.278	0.520	0.763
experience	0.035	0.006	0.024	0.035	0.046
I(experience^2)	-0.001	0.000	-0.001	-0.001	0.000
education	0.090	0.008	0.073	0.090	0.106

This summary includes some statistics about the computing times as well as the posterior mean, standard deviation, quartiles of the fixed effects ($\beta_1, \beta_2, \beta_3, \beta_4$) and of the hyperparameter (denoted by Precision for the Gaussian observations). The similar output can be obtained by means of these single commands to access the output (Blangiardo and Cameletti, 2015).

```
M3$summary.fixed
M3$summary.hyperpar
```

If we compare this model with the following standard regression model obtained running the R function lm (see Section II). We obtain the same results by using the command

```
summary(lm(formula,data=cps))
```

Table 3: Summary results using lm function

Estimate	Std. Error	t value	Pr(> |t|)
0.520	0.124	4.209	0.000
0.035	0.006	6.185	0.000
-0.001	0.000	-4.307	0.000
0.090	0.008	10.787	0.000

We essentially obtain the same results as we are assuming noninformative prior distributions on all the parameters and we are not specifying any hierarchical structure.

III. CONCLUSION

It is exceptional among the most critical issues in Bayesian 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 the Hamiltonian Monte Carlo method (HMC). In this article, the Bayesian approach is applied to wage equation in a semilogarithmic form based on cps data. The Normal mode is used as a Bayesian model to fit the data, and for the analysis. For this model, Stan and INLA give almost the same results. We also see that results for Stan and INLA are very similar both as point estimates and the distribution of posterior quantiles. It seems that Stan becomes useful only when your model cannot be coded in INLA.

REFERENCES

[1] Berndt, E. R. 1991. The practice of econometrics: classic and contemporary. Addison Wesley Publishing Company.
[2] Betancourt, M., and Girolami, M. 2015. Hamiltonian Monte Carlo for hierarchical models. Current trends in Bayesian methodology with applications, 79, 30.
[3] Blangiardo, M., and Cameletti, M. 2015. Spatial and spatio-temporal Bayesian models with R-INLA. John Wiley and Sons.
[4] Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., and Riddell, A. 2017. Stan: A probabilistic programming language. Journal of statistical software, 76(1).

[5] Gelman, A. and Hill, J. 2007. Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press, New York
[6] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. 2014. Bayesian Data Analysis. Chapman and Hall/CRC.
[7] Hoffman, M. D., and Gelman, A. 2014. The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. Journal of Machine Learning Research, 15(1):1593-1623.
[8] Hilbe, J. M., De Souza, R. S., and Ishida, E. E. 2017. Bayesian Models for Astrophysical Data: Using R, JAGS, Python, and Stan. Cambridge University Press.
[9] Akhtar, M. T., and Khan, A. A. 2014. Bayesian analysis of generalized log-Burr family with R. SpringerPlus, 3(1):185.
[10] Kleiber, C., and Zeileis, A. 2008. Applied econometrics with R. Springer Science and Business Media.
[11] Martins, T. G., Simpson, D., Lindgren, F., and Rue, H. 2013. Bayesian computing with INLA: new features. Computational Statistics and Data Analysis, 67:68-83.
[12] Rue, H., Martino, S., and Chopin, N. 2009. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. Journal of the royal statistical society: Series b (statistical methodology), 71(2):319-392.
[13] Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P., and Lindgren, F. K. 2017. Bayesian computing with INLA: a review. Annual Review of Statistics and Its Application, 4, 395-421.
[14] Stan Development Team 2018. RStan: the R interface to Stan. R package version 2.18.2, http://mc-stan.org/.

.