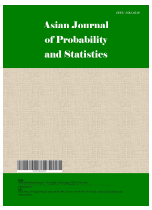

Asian Journal of Probability and Statistics

6(3): 25-46, 2020; Article no.AJPAS.54061
ISSN: 2582-0230

On Model Comparison: Application of Savage-Dickey Density Ratio to Bayes Factor

Olawale B. Akanbi[1*], Olusanya E. Olubusoye[1] and Samuel A. Babatunde[1]

[1]Department of Statistics, University of Ibadan, Nigeria.

Authors' contributions

This work was carried out in collaboration among all authors. Author OBA wrote the first draft of the manuscript, designed the study and managed the part of the literature searches. Author OEO modified the first draft and wrote the protocol. Author SAB performed the analyses of the study and managed the other part of the literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJPAS/2020/v6i330162
Editor(s):
(1) Dr. Jiteng Jia, School of Mathematics and Statistics, Xidian University, China.
(2) Dr. S. M. Aqil Burney, Department of Computer Science, University of Karachi, Pakistan.
(3) Dr. Manuel Alberto M. Ferreira, Retired Professor, Department of Mathematics, ISTA-School of Technology and Architecture, Lisbon University, Portugal.
Reviewers:
(1) Francisco Bulnes, Tecnológico de Estudios Superiores de Chalco, Mexico.
(2) Sie Long Kek, Universiti Tun Hussein Onn Malaysia, Malaysia.
(3) Josua Mwanyekange, University of Namibia, Namibia.
Complete Peer review History: http://www.sdiarticle4.com/review-history/54061

Original Research Article

Received: 01 December 2019
Accepted: 03 February 2020
Published: 07 February 2020

Abstract

Bayes factor is a major Bayesian tool for model comparison especially when the model priors are the same. In this paper, the Savage-Dickey Density Ratio (SDDR) is used to derive the Bayes factor to select a model from two competing models under consideration in a normal linear regression with an independent normal-gamma prior. The Gibbs sampling technique for the joint posterior distribution with equal prior precision for both the unrestricted and restricted models is used to obtain the model estimates. The result shows that the Bayes factor gave more support to the unrestricted model against the restricted and was consistent irrespective of changes in sample size.

Keywords: Nested model; prior precision; Savage-Dickey density ratio; Gibbs sampling.

**Corresponding author: E-mail: muhdbashola@yahoo.com;*

1 Introduction

Model comparison is the process by which models are compared to one another and the best model is selected using a particular model criterion. This is an effective Bayesian inference which normally requires choosing of the best model for the specific situation under investigation [1]. Usually in Bayesian paradigm, models are compared using Bayes factor, [2,3,4]. Bayes factors are notoriously difficult to compute, and the Bayes factor is only defined when the marginal density of y (dependent variable) under each model is proper. However, Bayes factors are easy to approximate with the Laplace-Metropolis Estimator, [5]. Although, initially the p-value of 0.05 was used for model comparison in Bayesian model selection but later showed that it cannot give much evidence against the null hypothesis [6,7]. Just of recent, [8] and [9] proposed computationally convenient default priors with desirable theoretical properties for the Bayes factor. Also, [10] pointed out the important of priors in Bayes factor when additional information is available for a model selection process. The Bayes factor uses the ratio of marginal likelihoods of the reduced model to the unreduced model. The SDDR is another way of writing Bayes factor for comparing nested models, using non-informative prior (part of the limitations of the SDDR. A simple method for approximating the Bayes factor that generalizes a method which in turn is attributed to the idea to Savage but did not compute for the likelihood, see [11,12,13,14,15]. The use of posterior simulation to compute Bayes factor which is applicable whenever the null hypothesis is a nested hypothesis was considered in the work of several authors [16,17]. The generalized version of the SDDR for representation of the Bayes factor of nested statistical models, the new version taking the form of a random Nikodym derivative and thus showed its applicability to a wider family of probability spaces than the original is shown in [18].

In this paper, the likelihood function is a multivariate normal distribution, with an independent normal-gamma prior, giving rise to the posterior conditional density of the parameters given the data and the precision belonging to the multivariate normal and the precision given the data and the parameters for the Gamma distribution. The normal linear regression model with independent normal-gamma prior is a very complex and complicated Bayesian econometrics model since it does not give a familiar posterior distributional form. Unlike the normal linear regression model with a conjugate normal-gamma prior which can be solved analytically. Independent normal-gamma prior can only be solved through a posterior simulation technique such as the Gibbs sampler technique. Thus, the posterior distribution does not take a familiar distributional form given rise to a situation whereby the marginal likelihood and the predictive densities do not exist, which makes it difficult for model comparison except through the use of Bayes factor which incorporates a Bayesian tool called the Savage-Dickey Density Ratio technique for the model comparison, which is the main focus for this study.

2 Methodology

The normal linear regression model used comprised of k independent explanatory variables, with additive error component which is normally distributed.

The model:

$$Y = X\beta + \varepsilon \tag{1}$$

Where, Y is (n x 1) vector of the response variable, X is (k x k) matrix of the explanatory variables. β is the ((k+1) x 1) vector of the regression parameters. ε is the error component which is normally independently and identically distributed.

Suppose further, (1) has a normal linear model, y with mean $X\beta$ and error precision $h=\frac{1}{\sigma^2}$, the expression for the likelihood density denoted by $P(y/\beta,h)$ is given as:

$$P(y|\beta, h) = \frac{h^{\frac{N}{2}}}{(2\pi)^{\frac{N}{2}}} \exp[-\frac{h}{2}(Y - X\beta)'(Y - X\beta)] \tag{2}$$

The independent prior cannot be joined together because of the theorem of independence, [19]. Therefore, $p(\beta, h) = p(\beta) \cdot p(h)$, where, $p(\beta) \sim$ *Normal distribution* & $p(h) \sim$ *Gamma distribution*

$$P(\beta) = \frac{1}{(2\pi)^{\frac{N}{2}}} |V|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left(\beta - \underline{\beta}\right)' \underline{V}^{-1} \left(\beta - \underline{\beta}\right)\right] \tag{3}$$

and

$$P(h) = C_G^{-1} h^{\frac{v-2}{2}} \exp\left(\frac{-h\underline{v}}{2\underline{s}^{-2}}\right) \tag{4}$$

Where, C_G^{-1} is an integrating constant, It is deduced that: $E(\beta) = \underline{\beta}$ is the prior mean of β and $Var(\beta) = \underline{V}$ is the prior covariance matrix of β with the mean of h, as $\underline{s}^{-2}$ and $\underline{v}$ degree of freedom. $P(h) = C_G^{-1} h^{\frac{v-2}{2}} \exp\left(\frac{-h\underline{v}}{2\underline{s}^{-2}}\right)$

Considering the Bayes Theorem (Joyce, James (2003)), then the posterior for (β, h) given the data is

$$P(\beta, h|y) = P(\beta|y, h) \,.\, P(h|y, \beta) \tag{5}$$

But note that (5), $P(\beta, h|y) \neq P(\beta|y, h) \,.\, P(h|y, \beta)$

Combining (2), (3) & (4), equation (5) becomes

$$P(\beta, h|y) = \frac{h^{\frac{N}{2}}}{(2\pi)^{\frac{N}{2}}} \frac{1}{(2\pi)^{\frac{k}{2}}} |V|^{-\frac{1}{2}} \exp -\frac{1}{2}\left[h(Y - X\beta)'(Y - X\beta) + \left(\beta - \underline{\beta}\right)' \underline{V}^{-1} \left(\beta - \underline{\beta}\right)\right] \left[C_G^{-1} h^{\frac{v-2}{2}} \exp\left(\frac{-h\underline{v}}{2\underline{s}^{-2}}\right)\right] \tag{6}$$

The term in the bracket of the first exponential can be further simplified as:

$$\begin{aligned} h\left(Y - X\beta\right)' (Y - X\beta) + (\beta - \underline{\beta})' \underline{V}^{-1} (\beta - \underline{\beta}) &= hY'Y - 2h\beta' X'Y + h\beta' X'X\beta + \beta' \underline{V}^{-1}\beta - 2\beta' \underline{V}^{-1}\underline{\beta} + \underline{\beta}' \underline{V}^{-1}\underline{\beta} \\ &= hY'Y + \underline{\beta}' \underline{V}^{-1}\underline{\beta} + \beta'(\underline{V}^{-1} + hX'X)\beta - 2\beta'(hX'Y + \underline{V}^{-1}\underline{\beta}) \end{aligned} \tag{7}$$

Let $\overline{V} = (\underline{V}^{-1} + hX'X)^{-1}$, *then*, $\overline{V}^{-1} = (\underline{V}^{-1} + hX'X)$ & $\overline{\beta} = \overline{V}(hX'Y + \underline{V}^{-1}\underline{\beta})$

Hence, the new parameters are substituted into the (7) above:

$$h\left(Y-X\beta\right)'(Y-X\beta)+(\beta-\underline{\beta})'\underline{V}^{-1}(\beta-\underline{\beta})=hY'Y+\underline{\beta}'\underline{V}^{-1}\underline{\beta}+\beta'\overline{V}^{-1}\beta-2\beta'\overline{V}^{-1}\overline{\beta}$$

Introducing $-\overline{\beta}'\overline{V}^{-1}\overline{\beta}$ and $+\overline{\beta}'\overline{V}^{-1}\overline{\beta}$ terms into the equation above;

$$=hY'Y+\underline{\beta}'\underline{V}^{-1}\underline{\beta}+\beta'\overline{V}^{-1}\beta-2\beta'\overline{V}^{-1}\overline{\beta}-\overline{\beta}'\overline{V}^{-1}\overline{\beta}+\overline{\beta}'\overline{V}^{-1}\overline{\beta}$$

$$=hY'Y+\underline{\beta}'\underline{V}^{-1}\underline{\beta}-\overline{\beta}'\overline{V}^{-1}\overline{\beta}+\beta'\overline{V}^{-1}\beta-2\beta'\overline{V}^{-1}\overline{\beta}+\overline{\beta}'\overline{V}^{-1}\overline{\beta}$$

Let, $Q=hY'Y+\underline{\beta}'\underline{V}^{-1}\underline{\beta}-\overline{\beta}'\overline{V}^{-1}\overline{\beta}$ and $(\beta-\overline{\beta})'\overline{V}^{-1}(\beta-\overline{\beta})=\beta'\overline{V}^{-1}\beta-2\beta'\overline{V}^{-1}\overline{\beta}+\overline{\beta}'\overline{V}^{-1}\overline{\beta}$

$$\Rightarrow h(Y-X\beta)'(Y-X\beta)+(\beta-\underline{\beta})'\underline{V}^{-1}(\beta-\underline{\beta})=(\beta-\overline{\beta})'\overline{V}^{-1}(\beta-\overline{\beta})+Q \tag{8}$$

Substituting (8) into (6) and excluding the constants, it becomes the posterior for (*β, h*) given the data;

$$P(\beta,h\,|\,y)\propto\exp\left[-\frac{1}{2}\{(\beta-\overline{\beta})'\overline{V}^{-1}(\beta-\overline{\beta})\}\right]\cdot\exp[-\frac{1}{2}Q]h^{\frac{N+\underline{v}-2}{2}}\exp\left(\frac{-h\underline{v}}{2\underline{s}^{-2}}\right) \tag{9}$$

By ignoring the terms that do not involve *β*, then the posterior conditional probability of *β/y,h* is

$$P(\beta\,|\,y,h)\propto\exp\left[-\frac{1}{2}\{(\beta-\overline{\beta})'\overline{V}^{-1}(\beta-\overline{\beta})\}\right] \tag{10}$$

Which implies that $\beta|\,y,\,h\sim N(\beta,V)$, Multivariate Normal density where, $\bar{V}=(\underline{V}^{-1}+hX'X)^{-1}$ and $\bar{\beta}=\bar{V}\,(hX'Y+\underline{V}^{-1}\beta)$

Similarly, by treating equation (9) as a function of *h* ignoring terms that do not involve *h*,

The conditional posterior probability of *h/y,β* is

$$P(h\,|\,y,\beta)\propto h^{\frac{N+\underline{v}-2}{2}}\exp[-\frac{h}{2}\{(Y-X\beta)'(Y-X\beta)+\underline{v}\underline{s}^{2}\} \tag{11}$$

This also implies that $h|y,\,\beta\sim G(\bar{s}^{-2},\bar{v})$, a Gamma distribution

where, $\bar{v}=N+\underline{v}$ and $\bar{s}^{2}=\dfrac{(Y-X\beta)'(Y-X\beta)+\underline{v}\underline{s}^{2}}{\bar{v}}$

A Generalization of the SDDR to derive the Bayes factor for comparing the models:

Bayes factor is a summary of the evidence provided by the data in favor of one scientific theory represented by a statistical model, [5]. Thus, the Bayes factor comparing two models M_1 to M_2 is the ratio of marginal likelihood:

$$BF_{12}=\frac{p(y\,|\,M_1)}{p(y\,|\,M_2)} \tag{12}$$

Posterior odds ratio:

$$PO_{12} = BF_{12} \times \frac{p(M_1)}{p(M_2)}. \tag{13}$$

In this study, it is difficult to calculate the marginal likelihood $p(y/M_i), i = 1, 2$ directly. Hence, Bayes factor using SDDR to compare nested models only be available for certain type of priors and when certain conditions are satisfied (Appendix).

Let the likelihood (unrestricted model, M_2) and the joint prior be given as: $P(y|\beta, h, M_2)$ and $P(h|\beta = \beta_0, M_2)$ respectively, and the restricted model, M_1 has $\beta = \beta_0$ where, β_0 is a vector of constants, h is unrestricted in each model with likelihood and prior as; $P(y|h, M_1)$ and $P(h|M_1)$ respectively.

2.1 The SDDR theorem

Dickey [12] proposed that if and only if the prior of the unreduced model is equal to the prior of the reduced model, i.e. $P(h|\beta = \beta_0, M_2) = P(h|M_1)$ then the Bayes factor for comparing model 1 to model 2 is defined by:

$$BF_{12} = \frac{P(\beta = \beta_0 | y, M_2)}{P(\beta = \beta_0 | M_2)} \tag{14}$$

Proof: Recall, from posterior odds ratio we have;

$$PO_{12} = \frac{P(M_1|y)}{P(M_2|y)} = \frac{P(y|M_1) \cdot P(M_1)}{P(y|M_2) \cdot P(M_2)}$$

If $P(M_1) = P(M_2) = \frac{1}{2}$ (equal weight attached), then,

$$PO_{12} = \frac{P(M_1|y)}{P(M_2|y)} = BF_{12} \tag{15}$$

but in a situation where the posterior does not have a known form (analytic form), the marginal likelihood for y will not exist, hence; the approach of [20] can be used to prove the generalization of the SDDR; by definition;

$$BF_{12} = \frac{\int f(y|\beta = \beta_0, h, M_1) \cdot P(h|M_1) dh}{\int f(y|\beta, h) \cdot P(\beta, h, M_2) d\beta} \tag{16}$$

using a specific version of $P(h/\beta = \beta_0, M_2)$

$$BF_{12} = \frac{\int P(h|\beta = \beta_0, M_2) f(y|\beta = \beta_0, h) dh P(\beta = \beta_0, M_2)}{\int P(\beta, h, M_2) f(y|\beta, h) dh d\beta P(\beta = \beta_0, M_2)} \tag{17}$$

using a specific version of $P(\beta = \beta_0, M_2)$

$$BF_{12} = \frac{P(\beta = \beta_0 | y, M_2)}{P(\beta = \beta_0 | M_2)}$$

$$BF_{12} = \frac{\int P(h | M_1) \cdot f(y | \beta = \beta_0, h) dh}{m_2(y)} \tag{18}$$

The denominator is easily calculated, since the marginal for β is Normal, then the denominator is

$$P(\beta = \beta_0 | M_2) = \frac{1}{(2\pi)^{\frac{N}{2}}} |\underline{V}|^{-\frac{1}{2}} \exp[-\frac{1}{2}(\beta_0 - \underline{\beta})' \underline{V}^{-1} (\beta_0 - \underline{\beta})] \tag{19}$$

The Numerator is quite difficult to obtain but the Gibbs sampler will provide output, $\beta^{(s)}$ and $h^{(s)}$ for $s = S_0 + 1...S$, then, simply averaging $P(\beta = \beta_0 | y, h^{(s)}, M_2)$ across the draws $h^{(s)}$ will yield an estimate of $P(\beta = \beta_0 | y, M_2)$.

Hence;

$$P(\beta = \beta_0 | y, M_2) \simeq \frac{1}{S_1} \sum_{s=S_0+1}^{S} P(\beta = \beta_0 | y, h^{(s)}, M_2) \tag{20}$$

$$P(\beta = \beta_0 | y, M_2) = E[g(\beta, h) | y] \tag{21}$$

Table 1 shows the generalized standard for accepting or rejecting a model in Bayes factor model comparison. It depicts the Bayes factor values for the evidence hypothesis against the null hypothesis which spans from 1 to 100.

Table 1. The generalized standard for accepting or rejecting a model in Bayes factor model comparison

BF_{21}	*Log*10 *BF*21	EVIDENCE AGAINST H_0
1 to 3.2	0 to $\frac{1}{2}$	Not worth more than a bare mention
3.2 to 10	$\frac{1}{2}$ to 1	Substantially against
10 to 100	1 to 2	Strongly against
> 100	> 2	Decisive

Note: BF_{21}= Bayes factor for model 2 (Evidence hypothesis) against model 1(Null hypothesis)

3 Analysis and Discussion of Results

3.1 Application of Bayes factor for the nested model comparison

The data used for this study were artificially generated using Monte Carlo simulation technique, the four explanatory variables are drawn independently from a uniform [0,1] distribution. Values were fixed for the regression coefficients; $\beta_0 = 0.0$, $\beta_1 = 10.0$, $\beta_2 = 30.5$, $\beta_3 = 15.0$ and $\beta_4 = 6.5$, the error component drawn from a standard normal distribution, incorporated into the linear model to obtain the response variable, $y_j = X\beta + \varepsilon_j$, using three different sample sizes, 15, 200 & 500. The Savage-Dickey Density Ratio involves the comparison of nested model (model 1): the possible subsets of the original/ true model and model 2 (unrestricted).

Competing model: Model 1

Case 1: $y_j = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon_j$
Case 2: $y_j = \beta_0 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon_j$,
Case 3: $y_j = \beta_0 + \beta_1 X_1 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon_j$,
Case 4: $y_j = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_4 X_4 + \varepsilon_j$
Case 5: $y_j = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon_j$

True model: Model 2, $y_j = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon_j$

Then, H_0 : Model 1 (Restricted model) vs H_1: Model 2 (Unrestricted model)

Tables 2, 3 & 4 show estimates of the Prior, and the Posterior of the Gibbs sampler with 10,000 draws after the burn-ins of 1000 varying sample sizes with the prior and posterior respective standard deviations in parenthesis, the Gewekes Convergence Diagnostics (GCD), Numerical Standard Error (NSE), 95% Highest Posterior Density Interval (HPDI) and the Bayes factor (BF_{21}) for comparing model 2 to 1 using the normal linear regression model with Independent Normal-Gamma prior.

Table 2 shows that the true value is not far from the posterior estimates, most parameters show support for model 2 (unrestricted model), GCD values are less than 1.96, which shows support for convergence, minimal NSE. HPDI also showed support for model 2 except for the intercept's credible interval.

Table 2. The estimates when N = 15

When N=15	True value	Prior mean (SD)	Posterior mean (SD)	NSE	GCD	HPDI [95% C.I]	BF_{21} ($\beta_j = 0$)
β_0	0	0 (29.3)	0.5992 (1.1236)	0.0118	0.7882	-1.5850, 2.7879	2.60×10^{-9}
β_1	10	17 (50.4)	10.3063 (0.9805)	0.0103	-0.8112	8.3253, 12.1453	5.75×10^{-9}
β_2	30.5	13.5 (900.6)	32.2139 (0.00904)	0.8578	0.3075	30.5617, 33.953	1.62×10^{-7}
β_3	15	1.4 (600)	15.3105 (0.946)	0.01	0.4146	13.4032, 17.096	6.90×10^{-8}
β_4	6.5	10 (50)	6.1763 (1.0992)	0.0116	-1.013	3.9830, 8.2495	4.90×10^{-9}

Table 3 shows that the true value is not far from the posterior estimates, most parameters show support for model 2 (unrestricted model), GCD values are less than 1.96, which shows support for convergence, while the NSE decreased.

Table 3. The estimates when N = 200

When N=200	True value	Prior (SD)	Posterior (SD)	NSE	GCD	HPDI [95% C.I]	BF_{21} ($\beta_j = 0$)
β_0	0	0 (29.3)	0.0883 (0.2756)	0.0029	0.7284	-0.4437, 0.6236	2.72×10^{-9}
β_1	10	17 (50.4)	9.319 (0.239)	0.0025	0.1035	8.8734, 9.8009	5.91×10^{-9}
β_2	30.5	13.5 (900.6)	28.893 (0.2425)	0.0026	-0.861	28.3957, 29.353	1.65×10^{-7}
β_3	15	1.4 (600)	13.7415 (0.2471)	0.0026	-0.468	13.2528, 14.213	7.24×10^{-8}
β_4	6.5	10 (50)	6.0782 (0.2408)	0.0025	-0.8492	5.6090, 6.5475	5.21×10^{-9}

Table 4 shows that the true value is not far from the posterior estimates, most parameters show support for model 2 (unrestricted model), GCD values are less than 1.96, which shows support for convergence, while the NSE decreased.

Table 4. The estimates when N = 500

When N=500	True value	Prior (SD)	Posterior (SD)	NSE	GCD	HPDI [95% C.I]	BF_{21} ($\beta_j = 0$)
β_0	0	0 (29.3)	-0.1695 (0.1699)	0.0018	1.16	-0.5049, 0.1522	2.52×10^{-9}
β_1	10	17 (50.4)	10.3802 (0.1557)	0.0016	0.4884	10.0687, 10.682	5.44×10^{-9}
β_2	30.5	13.5 (900.6)	30.859 (0.1497)	0.0016	-0.0408	30.5672, 31.150	1.52×10^{-7}
β_3	15	1.4 (600)	15.1576 (0.1584)	0.0017	-0.5979	14.8337, 15.455	6.79×10^{-8}
β_4	6.5	10 (50)	6.6617 (0.1498)	0.0016	-1.951	6.3519, 6.9380	4.81×10^{-9}

3.2 Convergence investigations for the Gibbs sampler

Figures below show the graphical structures of the Gibbs Sampler, before and after a burn-ins of 1000 replications, from an MCMC draw of 10000 in order to be sure of convergence, using sample sizes of 15 & 500. The Figures labeled "*sampled.betaj, j* = 0,1,2,3,4" represent the draws before burn-in while "$final.betaj, \quad j = 0,1,2,3,4$" are for draws after burn-ins, alongside Histogram and Normal Q-Q plots are provided.

Fig. 1 shows the draws before and after burn ins, histogram & normal Q-Q plot of β_0 when the sample size is 15. Fig. 1(b) presents the fluntuation graph after the burn-ins.

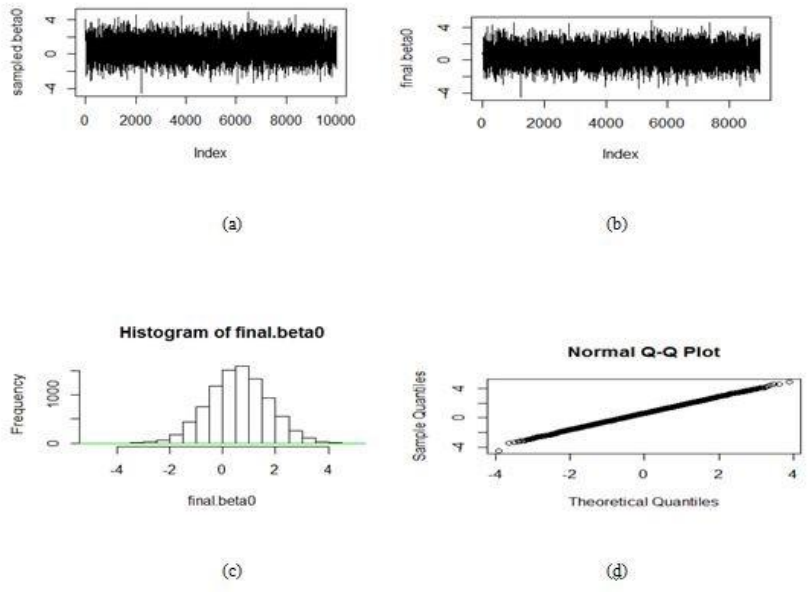

Fig. 1. Convergence diagnosis for β_0 at N=15

Fig. 2 shows the draws before and after burn-ins, histogram & normal Q-Q plot of β_1 when the sample size is 15. Fig. 2(b) presents the fluntuation graph after the burn-ins.

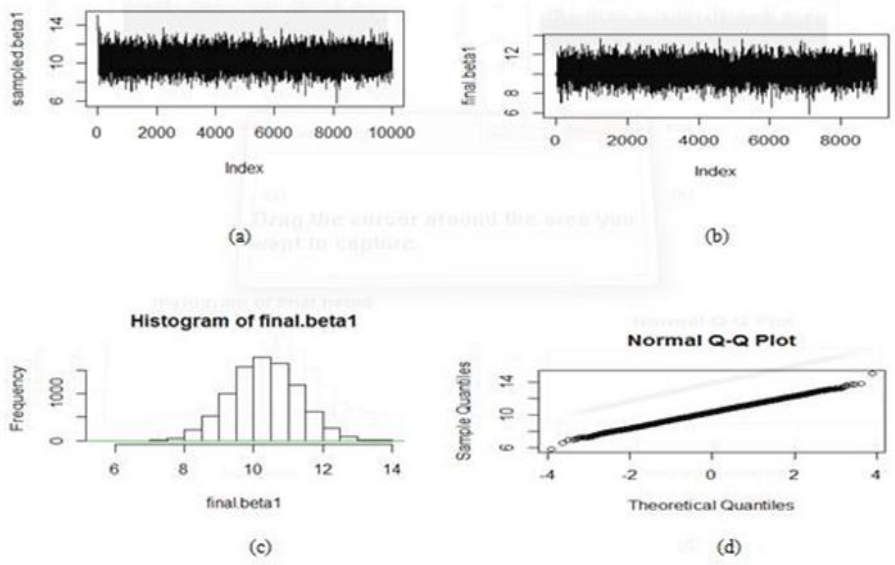

Fig. 2. Convergence diagnosis for β_1 at N=15

Fig. 3 shows the draws before and after burn-ins, histogram & normal Q-Q plot of β_2 when the sample size is 15. Fig. 3(b) presents the fluntuation graph after the burn-ins.

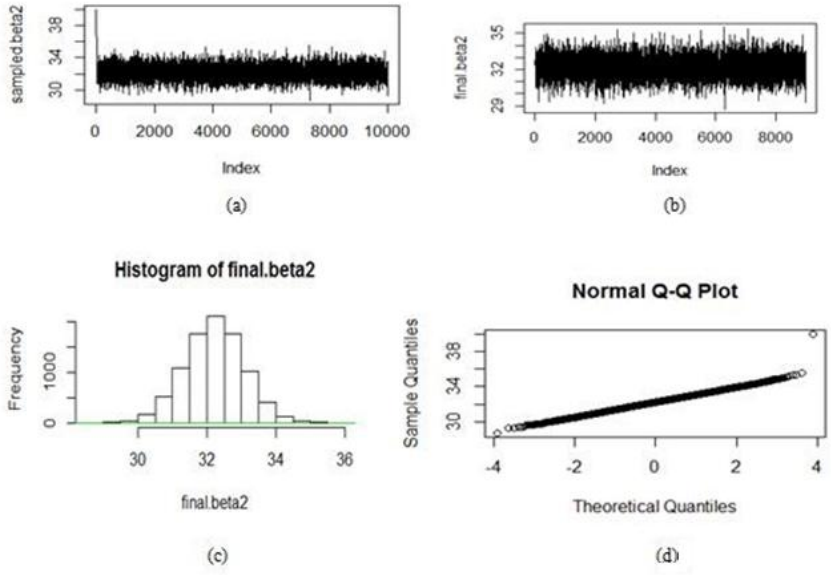

Fig. 3. Convergence diagnosis for β_2 at N=15

Fig. 4 shows the draws before and after burn-ins, histogram & normal Q-Q plot of β_3 when the sample size is 15. Fig. 4(b) presents the fluntuation graph after the burn-ins.

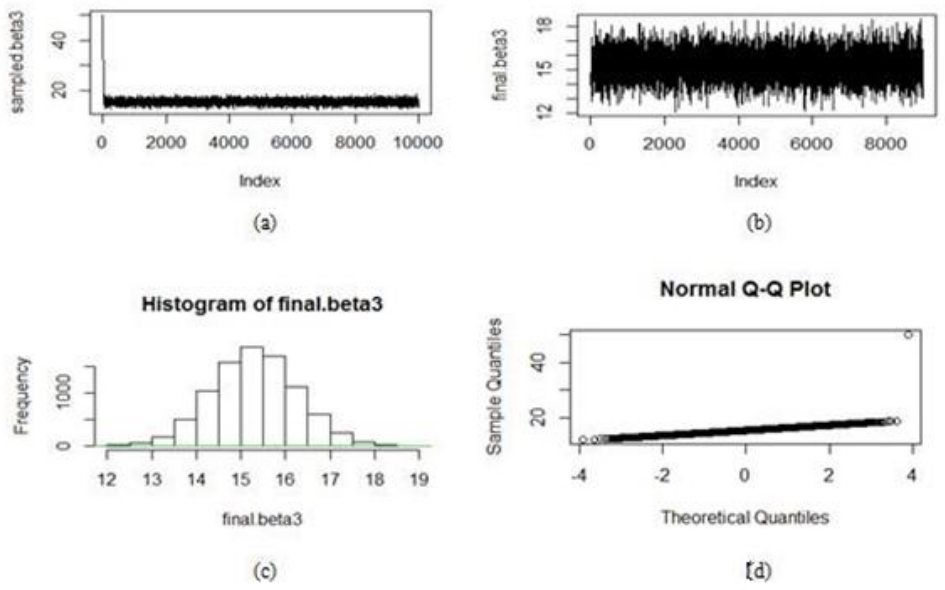

Fig. 4. Convergence diagnosis for β_3 at N=15

Fig. 5 shows the draws before and after burn-ins, histogram & normal Q-Q plot of β_4 when the sample size is 15. Fig. 5(b) presents the fluntuation graph after the burn-ins.

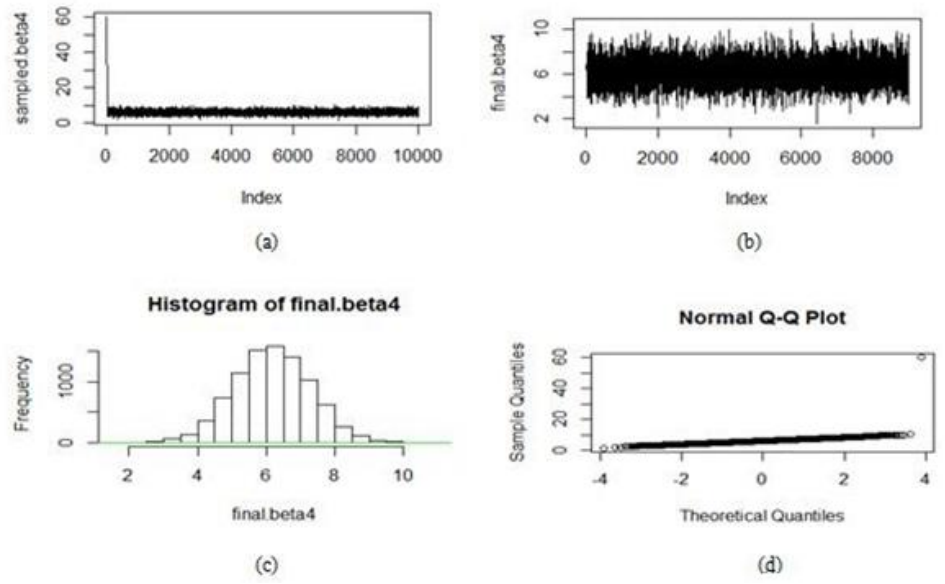

Fig. 5. Convergence diagnosis for β_4 at N=15

Fig. 6 shows the draws before and after burn-ins, histogram & normal Q-Q plot of β_0 when the sample size is 500. Fig. 6(b) presents the fluntuation graph after the burn-ins.

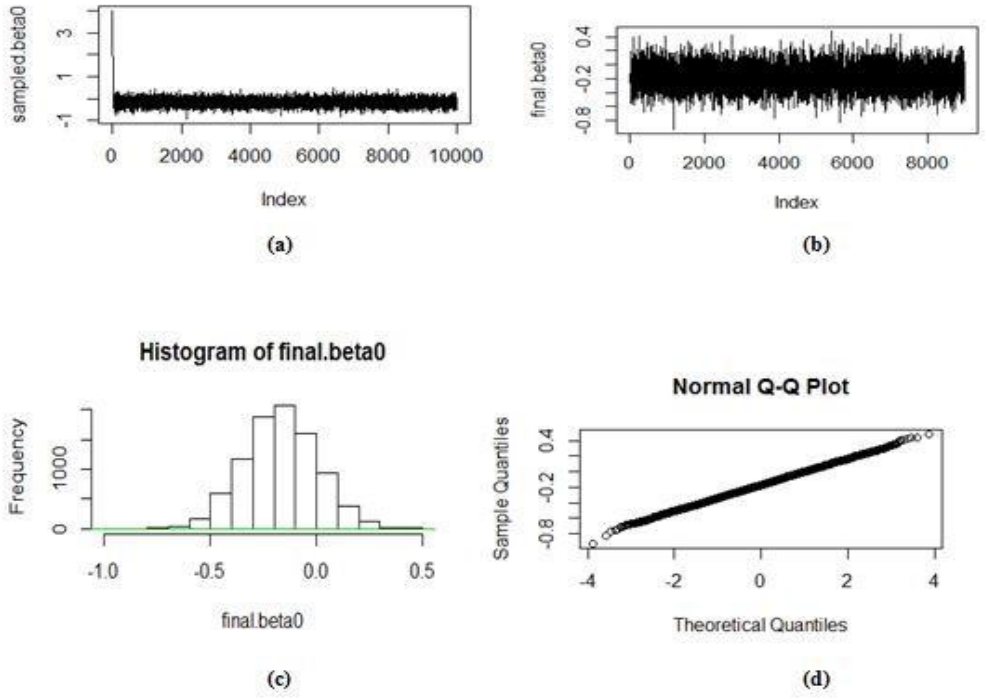

Fig. 6. Convergence diagnosis for β_0 at N = 500

Fig. 7 shows the draws before and after burn-ins, histogram & normal Q-Q plot of β_1 when the sample size is 500. Fig. 7(b) presents the fluntuation graph after the burn-ins.

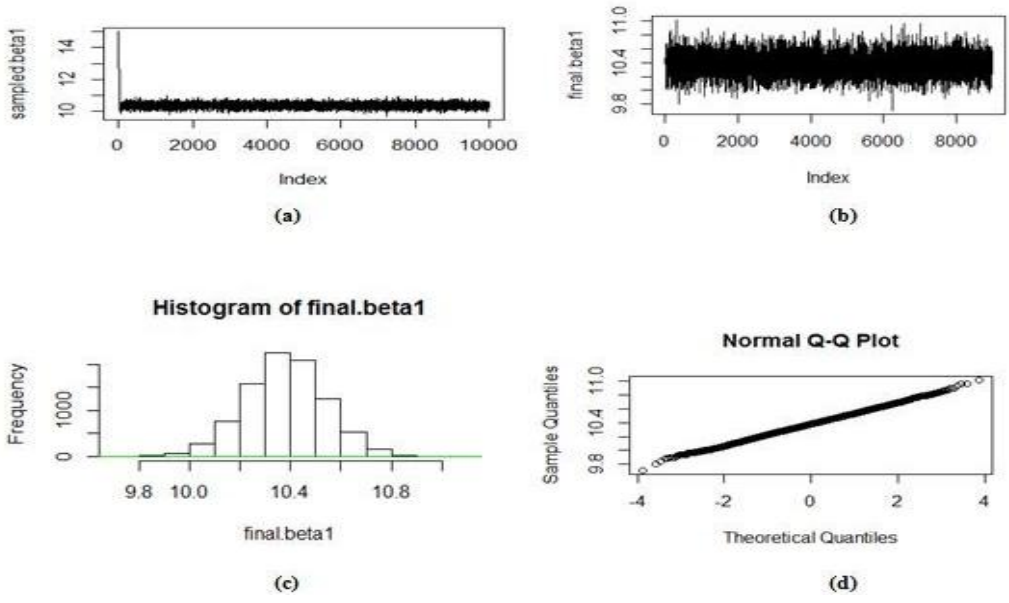

Fig. 7. Convergence diagnosis for β_1 at N = 500

Fig. 8 shows the draws before and after burn-ins, histogram & normal Q-Q plot of β_2 when the sample size is 500. Fig. 8(b) presents the fluntuation graph after the burn-ins.

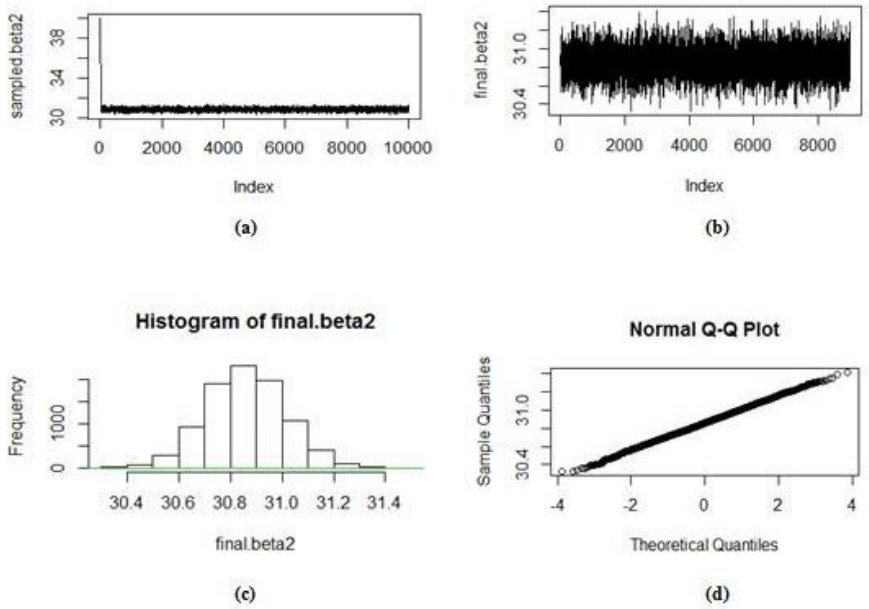

Fig. 8. Convergence diagnosis for β_2 at N = 500

Fig. 9 shows the draws before and after burn-ins, histogram & normal Q-Q plot of β_3 when the sample size is 500. Fig. 9(b) presents the fluntuation graph after the burn-ins.

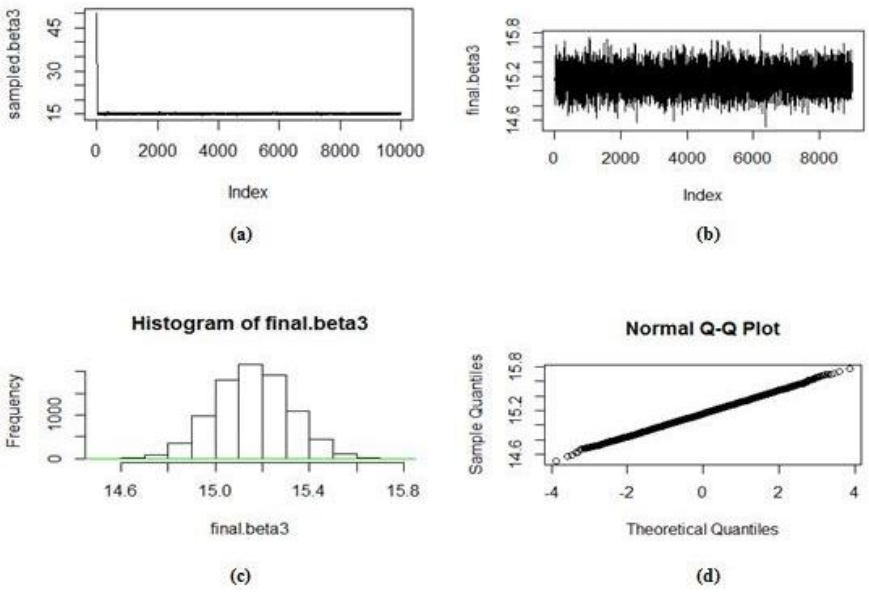

Fig. 9. Convergence diagnosis for β_3 at N = 500

Fig. 10 shows the draws before and after burn-ins, histogram & normal Q-Q plot of β_4 when the sample size is 500. Fig. 10(b) presents the fluntuation graph after the burn-ins.

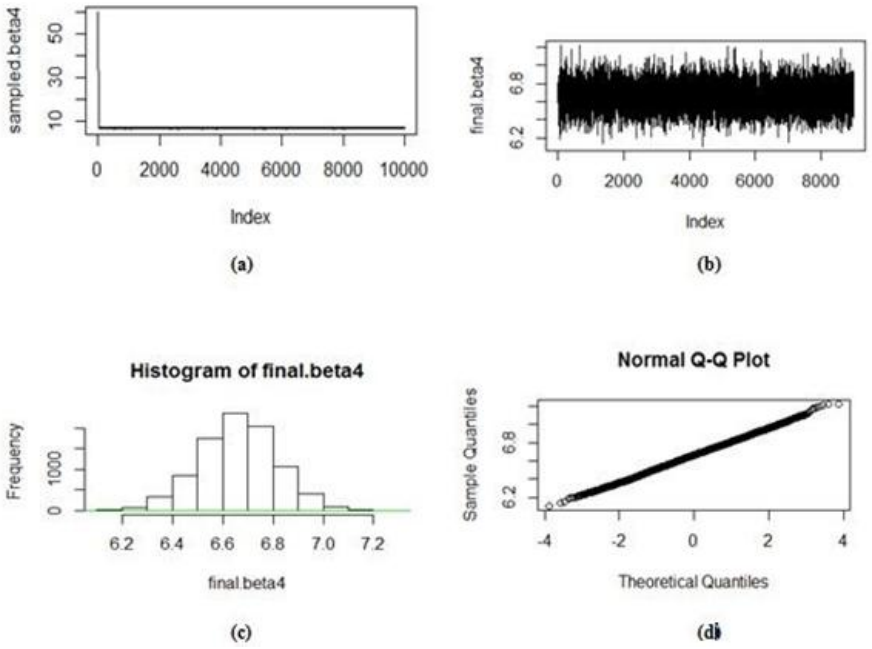

Fig. 10. Convergence diagnosis for β_4 at N = 500

4 Summary and Conclusion

This paper gave the possible cases of model restrictions that can occur in a Bayesian model comparison, particularly using the SDDR to compute the Bayes factors under various sample sizes of 15, 200 & 500. It obtained the posterior estimates of the normal linear regression model with an independent normal - gamma prior and the SDDR tool as the Bayes factor for model comparison in a situation where the form of the posterior densities are not of closed form or analytical. The tables above showed that the Bayes factor gave more support for the unrestricted model, M_2 because log of the BF_{21} values of all parameters except β_0 falls under the evidence against model 1 (restricted model). The Geweke's Convergence Diagnostics for all the parameters are less than 1.96 under the various sample sizes, which showed that the Gibbs sampler converged at all levels of parameters. Also, the least Numerical Standard Error is recorded at the stage of using the highest sample size, 500. Furthermore, when the 95% Highest Posterior Density Interval (HPDI) does not include zero, then it shows evidence against Model 1 (the restricted model) or otherwise. Therefore, HPDI only showed support for the restricted model, M_1 at β_0, while the other regression parameters showed support to the unrestricted model, M_2. Therefore, the Bayes factor derived for nested models remained consistent irrespective of changes in samples sizes, the NSE decreased as sample sizes increased and the Gibbs sampler converged accordingly.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Anderson CJ. The analysis of multivariate, longitudinal categorical data by Log-Multilinear models. Ph.D Thesis, University of Illinois at Urbana-Champaign; 1993.

[2] Jeffreys H. Theory of probability (3rd Edition). New York: Oxford University Press; 1961. Joyce, James. Zalta, Edward N, (Ed.), "Bayes' Theorem", The Stanford Encyclopedia of Philosophy (Spring 2019 Ed.), Metaphysics Research Lab, Stanford University; 2003.
(Retrieved 2020-01-17)

[3] Schonbrodt FD, Wagenmakers EJ. Bayes factor design analysis: Planning for compelling evidence. 2018;25(1):128-142.

[4] Gronau QF, Wagenmakers EJ, Heck DW, Matzke D. A simple method for comparing complex models: Bayesian model comparison for hierarchical multinomial processing tree models using Warp-III bridge sampling. Psychometrika. 2019a;84(1):261–284.

[5] Kass RE, Raftery AE. Bayes factors. Journal of the American Statistical Association. 1995;90:773–795.

[6] Berger JO, Mortera J. Interpreting the stars in precise hypothesis testing. International Statistical Review / Revue Internationale de Statistique. 1991;59(3):337-353.

[7] Edwards W, Lindman H, Savage LJ. Bayesian statistical inference for psychological research. Psychological Review. 1963;70:193-242.

[8] Liang F, Paulo R, Molina G, Clyde MA, Berger JO. Mixtures of g-priors for Bayesian variable selection. Journal of the American Statistical Association. 2008;103:410-423.

[9] Gronau QF, Heathcote A, Matzke D. Computing Bayes factors for evidence-accumulation models using Warp-III bridge sampling. Behavior Research Methods. 2019b;1–20.

[10] Wagenmakers EJ, Morey RD, Lee MD. Bayesian benefits for the pragmatic researcher. Current Directions in Psychological Science. 2016;25:169–176.

[11] Dickey J, Lientz B. The weighted likelihood ratio, sharp hypothesis about chances, the order of a Markov chain. The Annals of Mathematical Statistics. 1970;41(1):214-226.

[12] Dickey J. The weighted likelihood ratio linear hypotheses on normal location parameters. The Annals of Mathematical Statistics. 1971;42:204–223.

[13] Dickey J. Approximate posterior distributions. Journal of the American Statistical Association. 1976;71:680–689.

[14] Gunel E, Dickey J. Bayes factors for independence in contingency tables. Biometrika. 1974;61(3):545–557.

[15] Sander Van der Linden, Chryst B. No need for Bayes factors: A fully Bayesian evidence synthesis. Frontiers in Appl. Math. Stat. 2017;3(12):1-3.

[16] Newton MA, Raftery AE. Approximate Bayesian inference by the weighted likelihood bootstrap. J. Roy. Statist. Soc. B. 1994;56:3–48.

[17] Gelfand A, Dey D. Bayesian model choice: Asymptotics and exact calculations. Journal of the Royal Statistical Society Series B. 1994;56:501–514.

[18] Ewan, Cameron. A generalized Savage-Dickey density ratio. Electronic Journal of Statistics. 2013;176:1311–1292.

[19] Feller W. An introduction to probability theory and its applications. Third Edition, John Wiley & Sons Inc. 1971;1.

[20] Verdinelli I, Wasserman L. Computing Bayes factors using a generalization of the Savage Dickey density ratio. Journal of the American Statistical Association. 1995;90:614-618.

Appendix

The 'R' Codes Used for this Study

The Algorithm below gives the set of data simulated using sample size of 15, the data analysis for the other Nine sample sizes can be obtained by just changing the sample sizes to the desired one using the R commands. The other part of it was used to obtain the Savage-Dickey density ratio (Bayes factor) for model comparison.

```
## THE NORMAL LINEAR MODEL WITH INDEPEDENT NORMAL-GAMMA PRIOR
#.....
# THE MODEL:
# y = XB+e, where, # y is Nx1 matrix of y's, # e is Nx1 vector, # B is Kx1 vector, # X is NxK matrix of x's
#.....
### THE LIKELIHOOD FUNCTION, P(y/B,h)
#.....
# y/B,h ~ N(XB,h^-1), where, h=1/sigma^2 (h is precision)
N=15 # N varies as; N1=15,N2=30,N3=45,N4=60,N5=90,N6=100,N7=200,N8=300,N9=400,N10=500 k=5
B0=0.0
B1=10.0
B2=30.5
B3=15.0 B4=6.5
set.seed(12345) #to call for x1 use the exact number(12345) x1=runif(N,0,1) x1 set.seed(12346) ##to call for
x2 use the exact number(12346) x2=runif(N,0,1) x2 set.seed(12347)#to call for x3 use the exact
number(12347) x3=runif(N,0,1) x3 set.seed(12348) ##to call for x4 use the exact number(12348)
x4=runif(N,0,1) x4 set.seed(2468) ##to call for e use the exact number(2468) e=rnorm(N,0,1) e
y=B0+(B1*x1)+(B2*x2)+(B3*x3)+(B4*x4)+e y=matrix(y,nrow=N,ncol=1,byrow=T) y
x=cbind(1,x1,x2,x3,x4) x
X=as.matrix(x) X v=N-k
xpx=t(X)%*%(X)      # x'x
xpxinv=solve(xpx)     # (x'x)^-1
xpy=t(X)%*%(y)      # x'y
B=(xpxinv)%*%(xpy)# B=(x'x)^-1*(x'y)
B
Y=(X)%*%(B)      # Y=XB
SSE=t((y-Y))%*%(y-Y)    # e'e=(y-XB)'*(y-XB) ssqr=SSE/v # ssqr=(y-XB)*(y-XB)/v ssqr h =
ssqrinv=(ssqr)^-1 h varE=matrix(c(h^-1,0,0,0,0,0,h^-1,0,0,0,0,0,h^-1,0,0,0,0,0,h^-1,0,0,0,0,0,h^-1), nrow=5,
ncol=5, byrow=T) varE
### sure;checking the ols estimates of Beta ols=lm(y~X) ols
#.....
### THE PRIOR (Independent Normal-Gamma),p(B,h)=p(B).p(h)
#..... # B/h ~ N(Bprior,Vpri)
Bpri=rbind(0,27,13.5,1.4,10.0)     # B of prior (5 x 1)vector
Bpri=as.matrix(Bpri, nrow=5,ncol=1)
Bpri
Vpri=matrix(c(29.30^2,0,0,0,0,0,50.4^2,0,0,0,0,0,900.60^2,0,0,0,0,0,600.0^2,0,0,0,0,0,50.0^2),nrow=5,ncol
=5,byrow=T)
Vpri
Vinvpri=solve(Vpri)
Vinvpri
# h ~ G(ssqrinvpri,vpri), where, ssqrinvpri=s^-2 #let sigma=1000, # ssqrinvpri=h=1/sigma^2=1/1000000^2
ssqrinvpri=1/(5000)^2 ssqrinvpri ssqrpri=1/(ssqrinvpri) ssqrpri vpri=5.46 # 1% of N # noninformative prior
## deduced that prior means & var-covs are:
```

```
## FOR B/h;
# E(B/h)=Bpri
Bprior.mean=Bpri
Bprior.mean
# Var(B/h)=Vpri #prior covariance matrix of B
Bpri.varcov=Vpri
Bpri.varcov
## FOR h;
# E(h)=ssqrinvpri & degree of freedom vpri h.mean=ssqrinvpri
h.mean hd.f=vpri hd.f
#.....
### THE POSTERIOR, p(B,h/y)
#..... # the posterior conditionals do not have an analytical form
# but we can only sample through a posterior simulation technique called the Gibbs Sampler
## from p(B,h/y)=P(y/B,h).p(B).p(h), it was deduced that
## p(B,h/y) (not equal to) p(B/y,h).p(h/y,B), then;
### B/y,h ~ N(Bpos,Vpos) #multivariate-normal distn
Vpos=solve(Vinvpri+(drop(h)*xpx)) # Vpos=(Vpri^-1+h(xpx))
Vpos
Bpos=Vpos%*%((Vinvpri%*%Bpri)+(drop(h)*xpy)) #Bpos=Vpos(Vpri^-1+h(xpy))
Bpos
### h/y,B ~ G(ssqrinvpos,vpos) #Gamma distn vpos=N+vpri vpos
ssqrpos=(SSE+(vpri*ssqrpri))/(vpos) ssqrpos ssqrinvpos=solve(ssqrpos) ssqrinvpos
#.....
#THE GIBBS SAMPLER
#.....
# Define the number of runs for the sampler and create storage vectors for the samples mcmc = 10000
sampled.beta0 = numeric(mcmc) sampled.beta1 = numeric(mcmc) sampled.beta2 = numeric(mcmc)
sampled.beta3 = numeric(mcmc) sampled.beta4 = numeric(mcmc)
sampled.h = numeric(mcmc)
Bpri=rbind(0,27,13.5,1.4,10.0)        # B of prior (5 x 1)vector
Bpri=as.matrix(Bpri, nrow=5,ncol=1)
Bpri
Vpri=matrix(c(29.30^2,0,0,0,0,0,50.4^2,0,0,0,0,0,900.60^2,0,0,0,0,0,600.0^2,0,0,0,0,0,50.0^2),nrow=5,ncol
=5,byrow=T)
Vpri
Vinvpri=solve(Vpri)
Vinvpri
# h ~ G(ssqrinvpri,vpri), where, ssqrinvpri=s^-2 #let sigma=1000, # ssqrinvpri=h=1/sigma^2=1/1000000^2
ssqrinvpri=1/(5000)^2 ssqrinvpri ssqrpri=1/(ssqrinvpri) ssqrpri vpri=5.46 # 1% of N # noninformative prior
# Initialize and run the loop current.beta = rbind(4,15,40,50,60) current.beta = as.matrix(current.beta)
current.h = 1 sampled.beta0[1] = current.beta[1,] sampled.beta1[1] = current.beta[2,] sampled.beta2[1] =
current.beta[3,] sampled.beta3[1] = current.beta[4,] sampled.beta4[1] = current.beta[5,] sampled.h[1] =
current.h
for(i in 2:mcmc){
  # Sample from the full conditional of B/h library(MASS) current.h = 1
  Bposj = Vpos%*%((Vinvpri%*%Bpri)+(drop(current.h)*xpy)) #Bpos=Vpos(Vpri^-1+h(xpy))
  Vposj=solve(Vinvpri+(drop(current.h)*xpx)) # Vpos=(Vpri^-1+h(xpx)) current.beta = mvrnorm(1, Bposj,
  Vposj) current.beta
  current.beta = as.matrix(current.beta) current.beta
  # Sample from the full conditional of h vpos=N+vpri vpos
  SSe = t(y-(X%*%(current.beta)))%*%(y-(X%*%(current.beta))) ssqrpos = (SSe+(vpri*ssqrpri))/(vpos)
  ssqrpos ssqrinvpos = ssqrpos^-1 ssqrinvpos current.h = rgamma(1, ssqrinvpos, vpos)
  # Store the results
```

```
   sampled.beta0[i] = current.beta[1,] sampled.beta1[i] = current.beta[2,] sampled.beta2[i] = current.beta[3,]
   sampled.beta3[i] = current.beta[4,] sampled.beta4[i] = current.beta[5,] sampled.h[i] = current.h
}
# Trace plots #before burnin plot(sampled.beta0, type = 'l') mean(sampled.beta0) qqnorm(sampled.beta0)
plot(sampled.beta1, type = 'l') mean(sampled.beta1) qqnorm(sampled.beta1) plot(sampled.beta2, type = 'l')
mean(sampled.beta2) qqnorm(sampled.beta2) plot(sampled.beta3, type = 'l') qqnorm(sampled.beta3)
mean(sampled.beta3) plot(sampled.beta4, type = 'l') mean(sampled.beta4) qqnorm(sampled.beta4)
plot(sampled.h, type = 'l') mean(sampled.h) qqnorm(sampled.h)
burnin=1000 final.beta0 = sampled.beta0[-(1:burnin)] final.beta1 = sampled.beta1[-(1:burnin)] final.beta2 =
sampled.beta2[-(1:burnin)] final.beta3 = sampled.beta3[-(1:burnin)] final.beta4 = sampled.beta4[-(1:burnin)]
final.h = sampled.h[-(1:burnin)]
#trace plot #after burnin plot(final.beta0, type = 'l') mean(final.beta0) qqnorm(final.beta0) plot(final.beta1,
type = 'l') mean(final.beta1) qqnorm(final.beta1) plot(final.beta2, type = 'l') mean(final.beta2)
qqnorm(final.beta2) plot(final.beta3, type = 'l') mean(final.beta3) qqnorm(final.beta3) plot(final.beta4, type
= 'l') mean(final.beta4) qqnorm(final.beta4) plot(final.h, type = 'l') mean(final.h)
# Histograms of the samples minus burnin hist(final.beta0, prob=TRUE, xlab="x-variable", ylim=c(-4,4),
main="normal curve over histogram") hist(final.beta0) abline(lsfit(1:10000, sampled.beta0,
intercept=FALSE), col=3) abline(a=NULL, b=NULL, h=NULL, v=NULL, reg=NULL, coef=NULL,
untf=FALSE,col = 'red', lwd = 3) hist(final.beta1) abline(lsfit(1:10000, sampled.beta1, intercept=FALSE),
col=3) hist(final.beta2) abline(lsfit(1:10000, sampled.beta2, intercept=FALSE), col=3) hist(final.beta3)
abline(lsfit(1:10000, sampled.beta3, intercept=FALSE), col=3) hist(final.beta4) abline(lsfit(1:10000,
sampled.beta4, intercept=FALSE), col=3) hist(final.h)
# curve showing posterior density
## curve(final.beta0, from=0, to=1,xlab="p",ylab="Density",lty=1,lwd=4)
abline(v = h, col = 'red', lwd = 3)
mean(final.beta0) mean(final.beta1) mean(final.beta2) mean(final.beta3) mean(final.beta4) mean(final.h)
#THE CREDIBLE INTERVALS OF B AND h beta0.interval = quantile(final.beta0, c(.025,.5, .975))
beta0.interval beta1.interval = quantile(final.beta0, c(.025,.5, .975)) beta1.interval beta2.interval =
quantile(final.beta0, c(.025,.5, .975)) beta2.interval beta3.interval = quantile(final.beta0, c(.025,0.5,.975))
beta3.interval beta4.interval = quantile(final.beta0, c(.025,.5, .975)) beta4.interval
h.interval = quantile(final.h, c(.025,.5, .975))
h.interval
## SUMMARY OF THE GIBBS SAMPLER ESTIMATES summary(sampled.beta0)
summary(sampled.beta1) summary(sampled.beta2) summary(sampled.beta3) summary(sampled.beta4)
summary(sampled.h)
### install packages 'coda' and 'boa'
### the MCMC Output analysis & Diagnostics with the coda library
## Install.packages("coda") library("coda") ## codamenu() help(package="coda")
## to obtain the summary of gibbs sampled,trace plots and density curve
## for final.beta0 b0.mcmc=mcmc(final.beta0) summary(b0.mcmc) plot(b0.mcmc, col="blue") title('b0',
xlab = 'mcmc', ylab = 'b0.mcmc') autocorr.plot(b0.mcmc, col="blue") effectiveSize(b0.mcmc) # watchout
for capital "S"
## for final.beta1 b1.mcmc=mcmc(final.beta1) summary(b1.mcmc) plot(b1.mcmc, col="pink") title('b1',
xlab = 'mcmc', ylab = 'b1.mcmc') autocorr.plot(b1.mcmc, col="pink") effectiveSize(b1.mcmc) # watchout
for capital "S" ## for final.beta2 b2.mcmc=mcmc(final.beta2) summary(b2.mcmc) plot(b2.mcmc,
col="purple") title('b2', xlab = 'mcmc', ylab = 'b2.mcmc') autocorr.plot(b2.mcmc, col="purple")
effectiveSize(b2.mcmc) # watchout for capital "S"
## for final.beta3 b3.mcmc=mcmc(final.beta3) summary(b3.mcmc) plot(b3.mcmc, col="blue") title('b3',
xlab = 'mcmc', ylab = 'b3.mcmc') autocorr.plot(b3.mcmc, col="blue") effectiveSize(b3.mcmc) # watchout
for capital "S"
## for final.beta4 b4.mcmc=mcmc(final.beta4) summary(b4.mcmc) plot(b4.mcmc, col="green") title('b4',
xlab = 'mcmc', ylab = 'b4.mcmc') autocorr.plot(b4.mcmc, col="green") effectiveSize(b4.mcmc) # watchout
for capital "S"
## for final.h
```

```
h.mcmc=mcmc(final.h) summary(h.mcmc) plot(h.mcmc, col="blue") title('h', xlab = 'mcmc', ylab =
'h.mcmc') autocorr.plot(h.mcmc, col="blue") effectiveSize(h.mcmc) # watchout for capital "S"
## GEWEKE'S CONVERGENCE DIAGNOSTICS (GEWEKE'S CD)
## frac1 is the fraction of d 1st-set of est of B after burnin i.e 1000/9000
## frac2 is the fraction of d last-set of est of B after burnin
## DECISION RULE: if CD<1.96, then convergence of d-#     --MCMC algorithm has occured for all the
parameters library(coda) geweke.diag(final.beta0, frac1=0.11, frac2=0.55) geweke.diag(final.beta1,
frac1=0.11, frac2=0.55) geweke.diag(final.beta2, frac1=0.11, frac2=0.55) geweke.diag(final.beta3,
frac1=0.11, frac2=0.55) geweke.diag(final.beta4, frac1=0.11, frac2=0.55) #geweke.diag(final.h, frac1=0.11,
frac2=0.55)
#### GEWEKE CONVERGENCE PLOTS ######
geweke.plot(mcmc(final.beta0), frac1 = 0.11, frac2 = 0.55, nbins = 20, pvalue = 0.05, auto.layout = TRUE)
geweke.plot(mcmc(final.beta1), frac1 = 0.11, frac2 = 0.55, nbins = 20, pvalue = 0.05, auto.layout = TRUE)
geweke.plot(mcmc(final.beta2), frac1 = 0.11, frac2 = 0.55, nbins = 20, pvalue = 0.05, auto.layout = TRUE)
geweke.plot(mcmc(final.beta3), frac1 = 0.11, frac2 = 0.55, nbins = 20, pvalue = 0.05, auto.layout = TRUE)
geweke.plot(mcmc(final.beta4), frac1 = 0.11, frac2 = 0.55, nbins = 20, pvalue = 0.05, auto.layout = TRUE)
#geweke.plot(mcmc(final.h), frac1 = 0.11, frac2 = 0.55, nbins = 20, pvalue = 0.05, auto.layout = TRUE)
## HIGHEST PROBABILITY DENSITY (HPD)
HPDinterval(mcmc(final.beta0), prob = 0.95, attr('probability'))
HPDinterval(mcmc(final.beta1), prob = 0.95, attr('probability'))
HPDinterval(mcmc(final.beta2), prob = 0.95, attr('probability'))
HPDinterval(mcmc(final.beta3), prob = 0.95, attr('probability'))
HPDinterval(mcmc(final.beta4), prob = 0.95, attr('probability'))
HPDinterval(mcmc(final.h), prob = 0.95, attr('probability')) ###.....
```

THE BAYES FACTOR (SDDR)

```
#.....
## APPLICATION OF BAYES FACTOR (SDDR) FOR MODEL COMPARISON
## THE NORMAL LINEAR MODEL WITH INDEPEDENT NORMAL-GAMMA PRIOR
#.....
##BAYES FACTOR (SDDR) MODEL COMPARISON
#.....
## H0:M1(RESTRICTED) VS H1:M2 (UNRESTRICTED)
#.....
# let the unrestricted model be M2
# y = XB+e, where, # y is Nx1 matrix of y's, # e is Nx1 vector, # B is Kx1 vector, # X is NxK matrix of x's
# y=B0+(B1*x1)+(B2*x2)+(B3*x3)+(B4*x4)+e
#.....
# the restricted model be M1
#.....
# y0=(B1*x1)+(B2*x2)+(B3*x3)+(B4*x4)+e # H0:B0=0 VS H1: NOT H0
# y1=B0+(B2*x2))+(B3*x3)+(B4*x4)+e # H0:B1=0 VS H1: NOT H0 #
y2=B0+(B1*x1)+(B3*x3)+(B4*x4)+e # H0:B2=0      VS H1: NOT H0
# y3=B0+(B1*x1)+(B2*x2)+(B4*x4)+e # H0:B3=0   VS H1: NOT H0 #
y4=B0+(B1*x1)+(B2*x2)+(B3*x3)+e # H0:B4=0      VS H1: NOT H0 #.....
### THE LIKELIHOOD FUNCTION, P(y/B,h)
#.....
# y/B,h ~ N(XB,h^-1), where, h=1/sigma^2 (h is precision)
N=15 # N varies as; N1=15,N2=45,N3=60,N4=60,N5=90,N6=100,N7=200,N8=300,N9=400,N10=500 k=4
#B0=0.0
B1=10.0
B2=30.5
```

```
B3=15.0 B4=6.5 set.seed(12345) #to call for x1 use the exact number(12345) x1=runif(N,0,1) x1
set.seed(12346) ##to call for x2 use the exact number(12346) x2=runif(N,0,1) x2 set.seed(12347)#to call for
x3 use the exact number(12347) x3=runif(N,0,1) x3 set.seed(12348) ##to call for x4 use the exact
number(12348) x4=runif(N,0,1) x4 set.seed(2468) ##to call for e use the exact number(2468)
e=rnorm(N,0,1) e
#.....
# H0:B0=0 VS H1: NOT H0
y0=(B1*x1)+(B2*x2)+(B3*x3)+(B4*x4)+e # H0:B0=0 VS H1: NOT H0 #.....
y0=matrix(y0,nrow=N,ncol=1,byrow=T) y0
x=cbind(x1,x2,x3,x4) x
X=as.matrix(x) X v=N-k
xpx=t(X)%*%(X)       # x'x
xpxinv=solve(xpx)            # (x'x)^-1 xpy0=t(X)%*%(y0)        # x'y
B=(xpxinv)%*%(xpy0)# B=(x'x)^-1*(x'y)
B
Y=(X)%*%(B)          # Y=XB
SSE=t((y0-Y))%*%(y0-Y)# SSE=(y-XB)'*(y-XB) ssqr=SSE/v              # ssqr=(y-XB)*(y-XB)/v
h=ssqrinv=(ssqr)^-1 h varE=matrix(c(h^-1,0,0,0,0,h^-1,0,0,0,0,h^-1,0,0,0,0,h^-1), nrow=4, ncol=4,
byrow=T) varE
### sure;checking the ols estimates of Beta ols=lm(y0~X) ols
#.....
### THE PRIOR (Independent Normal-Gamma),p(B,h)=p(B).p(h)
#..... # B ~ N(Bprior,Vpri)
Bpri=rbind(27,13.5,1.4,10.0)     # B of prior (4 x 1)vector
Bpri=as.matrix(Bpri, nrow=4,ncol=1)
Bpri
Vpri=matrix(c(50.4^2,0,0,0,0,900.6^2,0,0,0,0,600.0^2,0,0,0,0,50.0^2),nrow=4,ncol=4,byrow=T) Vpri
Vinvpri=solve(Vpri)
Vinvpri
# h ~ G(ssqrinvpri,vpri), where, ssqrinvpri=s^-2 #let sigma=1000, # ssqrinvpri=h=1/sigma^2=1/5000^2
ssqrinvpri=1/(5000)^2 ssqrinvpri ssqrpri=1/(ssqrinvpri) ssqrpri vpri=5.46 # 1% of N # noninformative prior
## deduced that prior means & var-covs are:
## FOR B/h;
# E(B/h)=Bpri
Bprior.mean=Bpri
Bprior.mean
# Var(B/h)=Vpri #prior covariance matrix of B
Bpri.varcov=Vpri
Bpri.varcov
## FOR h;
# E(h)=ssqrinvpri & degree of freedom vpri h.mean=ssqrinvpri
h.mean hd.f=vpri hd.f
#.....
### THE POSTERIOR, p(B,h/y)
#..... # the posterior conditionals do not have an analytical form
# but we can only sample through a posterior simulation technique called the Gibbs Sampler
## from p(B,h/y)=P(y/B,h).p(B).p(h), it was deduced that
## p(B,h/y) (not equal to) p(B/y,h).p(h/y,B), then;
### B/y,h ~ N(Bpos,Vpos) #multivariate-normal distn
Vpos=solve(Vinvpri+(drop(h)*xpx)) # Vpos=(Vpri^-1+h(xpx))
Vpos
Bpos=Vpos%*%((Vinvpri%*%Bpri)+(drop(h)*xpy0)) #Bpos=Vpos(Vpri^-1+h(xpy))
Bpos
```

```
### h/y,B ~ G(ssqrinvpos,vpos) #Gamma distn vpos=N+vpri vpos ssqrpos=(SSE+(vpri*ssqrpri))/(vpos)
ssqrpos ssqrinvpos=solve(ssqrpos) ssqrinvpos
#.....................................................
#THE GIBBS SAMPLER
#.....................................................
# Define the number of runs for the sampler and create storage vectors for the samples mcmc = 10000
#sampled.beta0 = numeric(mcmc) sampled.beta1 = numeric(mcmc) sampled.beta2 = numeric(mcmc)
sampled.beta3 = numeric(mcmc) sampled.beta4 = numeric(mcmc)
sampled.h = numeric(mcmc)
# Set the prior parameters. We'll use an uninformative prior centered at 2 and an uninformative prior for phi.
Bpri=rbind(27,13.5,1.4,10.0)        # B of prior (4 x 1)vector
Bpri=as.matrix(Bpri, nrow=4,ncol=1)
Bpri
Vpri=matrix(c(50.4^2,0,0,0,0,900.6^2,0,0,0,0,600.0^2,0,0,0,0,50.0^2),nrow=4,ncol=4,byrow=T)
Vpri
Vinvpri=solve(Vpri)
Vinvpri
# h ~ G(ssqrinvpri,vpri), where, ssqrinvpri=s^-2 #let sigma=1000, # ssqrinvpri=h=1/sigma^2=1/1000000^2
ssqrinvpri=1/(5000)^2 ssqrinvpri ssqrpri=1/(ssqrinvpri) ssqrpri vpri=5.46 # 1% of N # noninformative prior
# Initialize and run the loop current.beta = rbind(15,40,50,60) current.beta = as.matrix(current.beta) current.h
= 1
#sampled.beta0[1] = current.beta[1,] sampled.beta1[1] = current.beta[1,] sampled.beta2[1] = current.beta[2,]
sampled.beta3[1] = current.beta[3,] sampled.beta4[1] = current.beta[4,]
sampled.h[1] = current.h
for(i in 2:mcmc){
   # Sample from the full conditional of B/h library(MASS) current.h = 1
   Bposj = Vpos%*%((Vinvpri%*%Bpri)+(drop(current.h)*xpy0)) #Bpos=Vpos(Vpri^-1+h(xpy))
   Vposj=solve(Vinvpri+(drop(current.h)*xpx)) # Vpos=(Vpri^-1+h(xpx)) current.beta = mvrnorm(1, Bposj,
   Vposj) current.beta current.beta=as.matrix(current.beta) current.beta
   # Sample from the full conditional of h vpos=N+vpri vpos
   SSe = t(y0-(X%*%(current.beta)))%*%(y0-(X%*%(current.beta))) ssqrpos = (SSe+(vpri*ssqrpri))/(vpos)
   ssqrpos ssqrinvpos = ssqrpos^-1 ssqrinvpos current.h = rgamma(1, ssqrinvpos, vpos)
   # Store the results
   #sampled.beta0[i] = current.beta[1,] sampled.beta1[i] = current.beta[1,] sampled.beta2[i] = current.beta[2,]
   sampled.beta3[i] = current.beta[3,] sampled.beta4[i] = current.beta[4,]
   sampled.h[i] = current.h
}
burnin=1000
#final.beta0 = sampled.beta0[-(1:burnin)] final.beta1 = sampled.beta1[-(1:burnin)] final.beta2 =
sampled.beta2[-(1:burnin)] final.beta3 = sampled.beta3[-(1:burnin)] final.beta4 = sampled.beta4[-(1:burnin)]
final.h = sampled.h[-(1:burnin)]
# to represent the BAYES FACTOR SUMMATION #final.beta0 = mean(final.beta0) final.beta1 =
mean(final.beta1) final.beta2 = mean(final.beta2) final.beta3 = mean(final.beta3) final.beta4 =
mean(final.beta4) mean(final.h)
#.....................................................
# THE BAYES FACTOR (SDDR-GENERALIZATION)
#.....................................................
# FOR DENOMENATOR
# P(Beta=B0/M2)=1/(2*3.142)^(k/2)*(det(Vpri))^(-0.5)*exp(-0.5*t((B-Bpri))%*%(solve(Vpri))%*%(B-
Bpri)) pD = 1/(2*3.142)^(k/2)*(det(Vpri))^(-0.5)*exp(-(0.5)*t(B-Bpri)%*%(solve(Vpri))%*%(B-Bpri)) pD
#.....................................................
#for B0=0
## NUMERATOR
# P(B=B0/y,M2) # from unreduced model
```

```
## p(B=B0/y,h^(s),M2)=1/(2pi)^k/2*(det(Vpos))^-0.5*exp[-1/2*(B0-Bpos)'*solve(Vpos)*(B0-Bpos)]
## (1/S1)*sum(p(B=B0/y,h^(s),M2)) = P(B=B0/y,M2)
## S1=S-S0, implies that S1=10000-1000(mcmc-burnin) S1 = 9000 p = 1/S1
## OR (1/S1)*(final.beta1+final.beta2+final.beta3+final.beta4)= P(B=B0/y,M2) pN =
p*(final.beta1+final.beta2+final.beta3+final.beta4) pN
## THE BAYES FACTOR (SDDR generalization)
## BF12 = P(B=B0/y,M2)/P(Beta=B0/M2)
BF12=pN/pD
BF12# THE BAYES FACTOR for Bj, j=1,2,...,k
## using jeffrey's suggestion
BF21=1/BF12
BF21
```

© 2020 Akanbi et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://www.sdiarticle4.com/review-history/54061