International Journal of Computational and Theoretical Statistics
ISSN (2210-1519)
Int. J. Comp. Theo. Stat. 2, No. 1 (May-2015)

Bayesian Analysis in Industrial Applications using Markov Chain Monte Carlo Simulations

Pranesh Kumar[1] and Hemantha S B Herath[2]

[1] *Department of Mathematics, University of Northern British Columbia, 3333 University Way, Prince George, BC, Canada*
[2] *Department of Accounting, Faculty of Business, Brock University,500 Glenridge Avenue, St. Catharines, ON, Canada*

Received Dec. 12, 2014, Revised Apr. 6, 2015, Accepted Apr. 9, 2015, Published May 1, 2015

Abstract: Hierarchical modeling is often used a tool which, as an interdisciplinary effort, combines the estimation technique and data mining techniques to model reliability systems. The reliability of the model is measured in terms of how much sufficiently accurate model is over the entire input range and the level of confidence in predictions. WinBUGS is Windows based software which provides researchers, especially in production process engineering, with a very useful data analytical tool. WinBUGS has ability to fit complex statistical models which express interdependence among several response variables based on Bayesian methods of inference and Markov Chain Monte Carlo (MCMC) simulation. In this paper, we present a short description of WinBUGS and discuss implementation of WinBUGS programs by analyzing real data sets from two industrial applications. First application undertakes the analysis of the behavior of the overhead-costs with the number of machine-hours operated and the number of production-runs in a production process. In the second illustration, we analyze the relative importance of between fluxes variability versus sampling variation in a weld experiment which considers welding fluxes with differing chemical compositions.

Keywords: Regression Model, Gibbs Sampling, Bayesian Updating, WinBUGS.

1. INTRODUCTION

A major objective in modeling reliability systems is to capture the end-to-end behavior of large systems in a complex physical environment with the maximum level of accuracy. Hierarchical modeling methods are often used as a possible tool to modeling such systems reliably with multiple degrees of abstraction. The reliability of the model is generally measured in terms of how much sufficiently accurate model is over the entire input interval and the desired level of confidence one can attach to the predictions. For example, in predicting fuel efficiency of cars on different roads, where car fuel efficiency can be assumed to be a function of car and road parameters, sparse regression cubes can lead to the best categorization of cars for purposes of building fuel prediction models in each category. Categorization might be by car-class, make, model, manufacturer, year, or other attributes, however, these categories have a hierarchical structure. One may also aggregate these over years or over car-models to generate prediction models for larger categories. Such generalizations help when there is not enough data on each type of car to build a reliable model for that type alone. It may be noted though when the samples used to build a regression model are sparse, model may over fit the samples and may result in poor predictions. Reliability of a regression model is measured in terms of the predictive power of the model (Weiseberg 1980; Breima et al. 1984). For an interesting work in data mining and machine learning research that build such generalized hierarchies, we refer to the regression trees in which a tree of regression models is used. Regression Cubes are analytical processing frameworks that do the same by exploring all different ways of generalizing data and using mostly categorical attributes to divide the input spaces into smaller subspaces (Chan et al. 2006; Ahmadi et al. 2011). Hierarchical structures, like regression trees or regression cubes, tend to construct a large number of models to represent all different ways of dividing the input space into subspaces. A reliability measure for a particular regression model over an input range indicates the ability of the fitted model to predict the output. The measure also provides an indication of accuracy of prediction by calculating a confidence interval. Thus, the framework needs to make sure that the hierarchy of regression models used for prediction are indeed reliable. This mechanism is an extension to widely used forward selection techniques (Kaplan and Atkinson 1989; Sheu and O'Curry 1998).

E-mail: pranesh.kumar@unbc.ca, hemantha.herath@brocku.ca

The open-source software WinBUGS [©1996-2008 BUGS] is a Windows based computer program for the Bayesian analysis of statistical models using Markov Chain Monte Carlo (MCMC) numerical methods. WinBUGS stands for

Windows Bayesian analysis software Using Gibbs Sampling. In 1989, the MRC Biostatistics Unit, Cambridge initiated this project and developed WinBUGS software jointly with the Imperial College School of Medicine at St. Mary's, London. The Bayesian analysis is advantageous in data analysis because Bayesian approach combines prior distributions (beliefs/experience) with the likelihood of (experimental/sample data) to drive the posterior/revised distributions (Gelman et al. 1995; Carlin and Louis 1996; Congdon 2001; Lunn et al. 2012). The MCMC numerical methods enhance the computations for most of the statistical models (Gilks et al. 1996).

We organize contents of our paper as follows. Following the brief introduction to what is meant by reliability measure in the context of regression models and WinBUGS in section 1, we present, in section 2, steps to install and implement WinBUGS. In section 3, we describe key aspects of WinBUGS application like monitoring parameter values, convergence and summaries of the sample values of the parameters of interest. We describe, in section 4, an industrial application which undertakes the BUGS analysis of the behavior of the overhead-costs with the number of machine-hours operated and the number of production-runs in a production process. In another illustration in section 5, we analyze an industrial experimental data which investigates the relative contributions of between fluxes variance and the sampling variance in a weld experiment in which welding fluxes with differing chemical compositions were prepared. We conclude paper in section 6 with some remarks. WinBUGS syntax for some commonly used probability distributions; data format and WinBUGS access from other software are presented in Appendix.

2. WINBUGS: INTRODUCTION

WinBUGS is an interactive Windows based BUGS program for Bayesian analysis of complex statistical models using Markov Chain Monte Carlo (MCMC) simulation. The basic idea behind the Gibbs sampling algorithm is to successively sample from the conditional distribution of each node given all others in the graph. These are known as the full conditional distributions. Under broad conditions this process results in samples from the joint posterior distribution of the unknown quantities. Given likelihood and prior distribution, WinBUGS samples model parameters from their posterior distributions. After sampling parameters from several iterations, parameter estimates are obtained. Empirical summary statistics and diagnostics are computed from these samples and are used to draw inference about the population constants. For an excellent reading of Bayesian inference methods, Markov Chain Monte Carlo methods and WinBUGS introduction, refer to Gelman et al. (1995), Gilks et al. (1996), Stringer et al. (2011), Lunn et al. (2012).

A. WinBUGS Installation

To install free WinBUGS software from http://www.mrc-bsu.cam.ac.uk/bugs, first download WinBUGS14.exe. Linux users can download the OpenBUGS program from http://mathstat.helsinki.fi/openbugs. The key for unrestricted use can be obtained by registration http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/register.shtml. The WinBUGS manuals and other users' resources are also available online.

B. To Fit a Model in WinBUGS

Step 1. Load the *program code*, *initial values* and *data*.

- Start the WinBUGS program by double click on the WinBUGS icon or WinBUGS14.exe file in the WinBUGS14 directory.
- From File menu on tool bar, select Open option to open program directory/file or New option to create programming code, data and initial values in one file or in separate files.
- Select Model menu on tool bar and highlight Specification option to open *Specification Tool* window (Fig. 1).

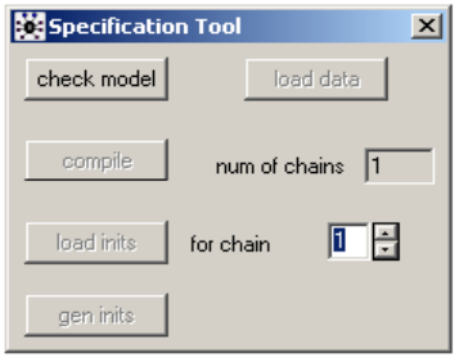


Figure 1. Specification tool window.

In the *Specification tool* window:

- Focus at the window containing the *model* code, *data* and *initial* values. Highlight the word *model* at the beginning of the program code and click *check model* box in the *Specification tool* window. A message should appear on the left bottom screen of the WinBUGS program window: *model is syntactically correct*.
- Open data (in a separate file or in the same file as the model code). Highlight the word *list* in the beginning of the data file and check *load data* in the *Specification tool* window. A message should appear on the left bottom screen of the WinBUGS program window: *data loaded*.
- If you desire to run more than one chain, specify the number in *nums of chains* box in the *Specification tool* window. The default is 1 chain.
- Click on the *compile* box in the *Specification tool* window to compile the model. A message should appear on the left bottom screen of the WinBUGS program window: *model compiled*.
- Open initial values (in a separate file or in the same file as the model code). Highlight the word *list* in the beginning of set of initial values and check *load inits* box in the *Specification tool* window. A message should appear on the left bottom screen of the WinBUGS program window: *this chain contains uninitialized variables*.
- If more than 1 chain to run is specified, load separate initial values for each chain by repeating the above steps for each file.
- Click *gen inits* box in the *Specification tool* window. A message should appear on the left bottom screen of the WinBUGS program: *initial values generated, model initialized.*
- Close the *Specification tool* window.

Step 2. Set *monitors* to store sampled values for parameters of interest.
From the File menu on tool bar, select Inference and click *samples* box. *Sample Monitor Tool* window opens (Fig. 2).

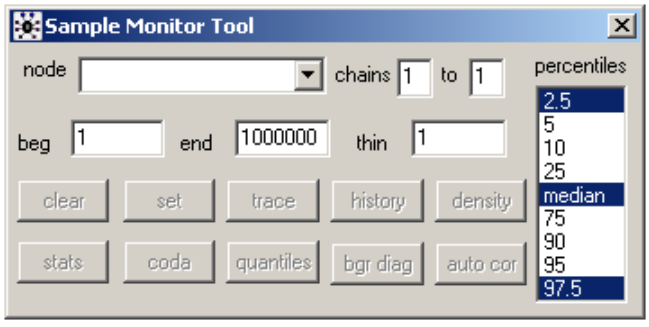


Figure 2. Sample monitor tool window.

In the *Sample Monitor Tool* window:

- In the *node* box, type parameter (say, theta) and click *set*.
- Repeat the same for other parameters of interest.

Step 3. Run the simulations.
Select Model menu on the toolbar and highlight *Update* option. *Update Tool* window opens (Fig. 3).

Figure 3. Update tool window.

In the *Update Tool* window:

- Type the number of updates (iterations of the simulation) in *updates* box (Default is 1000).
- Click once on *updates* box and program starts simulating values for each parameter in the model.
- The *iteration* box indicates number of updates currently being completed.
- The number of times this value is revised depends on the number set for *refresh* box (Default = 100 iterations). To report more frequently, set *refresh* to 10, 5, 1.
- When updates are finished, a message should appear on the left bottom screen of the WinBUGS program: *updates took**** *s.*

Step 4. View *graphical* and *numerical* summaries of samples for the parameters of interest for which we set the monitors.

- In the Inference window, select *sample monitor tool*. In the *sample monitor tool:*
- In *node* box, type the name of the parameter (or * for all parameters) for posterior inference.
- Several boxes about numerical summaries appear on *sample monitor tool* window: *trace, history, density, stats, coda, quantiles, bgr diag, auto cor*.

Step 5. Save files created during execution of WinBUGS program. Focus the window containing results/information of interest and select the Save As option from the File menu on the tool bar.
Step 6. To quit WinBUGS program, select Exit option from File menu on the tool bar.

3. MONITORING PARAMETER VALUES, CONVERGENCE AND SUMMARIES

The important decisions are required regarding checking convergence, number of iterations after convergence and posterior summaries of the model parameters. We need to set *monitors* for each parameter of interest and store the sampled values for those parameters. If not, WinBUGS will automatically discard the simulated values.

A. Monitors for Parameters

Two types of monitors can be set up in WinBUGS. A *sample* monitor tells WinBUGS to store every value it simulates for that parameter. We need to set sample monitors to view trace plots of the samples to check convergence and to derive posterior quantiles (for example, the posterior 95% Bayesian credible interval for that parameter).

- A *summary* monitor in WinBUGS stores the running mean and standard deviation for the parameter. The values saved contain less information than saving each individual sample in the simulation but require less storage. However, if we want to estimate Bayesian credible intervals, we need to set a *sample* monitor to store the necessary information for each parameter.
- To set a *summary* monitor: Select *Summary* from the Inference menu. Type the name of the parameter to be monitored in the box marked *node*. Click once on the box marked *set*. Repeat these steps for each parameter to be monitored.

B. Chain Convergence

Checking chain (simulation) convergence is very essential for the accuracy of the sample results. However, it is not easy to diagnose that a chain has converged. We need to select the number of chains, which means, sets of samples to generate. The default is 1 but we can run 2 or multiple chains to check the convergence of MCMC simulations.

1) Guidelines to Assess Convergence

- In practice, we may start using a single chain to check that the model compiles and runs and to obtain an estimate of the time taken per iteration.
- Once we are satisfied with the model, we can choose multiple chains (say, 2-5 chains) to obtain a final set of posterior estimates.
- Examine *trace* plots of the sample values versus iteration in the *Specification tool* window to look for evidence of when the simulation appears to have stabilized.
- To obtain live *trace* plots for a parameter:
  - Select Samples from the Inference menu.
  - Type the name of the parameter in the *node box* or select from the pull down list.
  - Click the *trace* box. An empty graphics *Dynamic trace* window appears on screen.
  - Repeat for each parameter if required.
  - Start running the simulations using the Update Tool.
  - *Trace* plots for these parameters appear *live* in the graphics windows.
- To obtain a *trace* plot showing the full *history* of the samples for any parameter set in the *Sample Monitor Tool* window during the *updates*:
  - Select Samples from the Inference menu.
  - Type the name of the parameter in the node box or select from the pull down list.
  - Repeat for each parameter if required.
  - Click the *history* box. A graphics *Time series* window showing the sample *trace* will appear.
- If we run more than one chain simultaneously, the *trace* and *history* plots will show each chain in a different color. Thus, we can be reasonably sure that convergence has been achieved if all the chains appear to be overlapping one another.

2) ***Number of Iterations after Convergence***

For assuring that chain (simulation) convergence has been achieved, we need to run the simulation for a further number of iterations to obtain samples that can be used for the posterior inference. More samples we have, more accurate will be our posterior estimates. One way to assess the accuracy of the posterior estimates is by calculating the Monte Carlo error for each parameter estimate. Monte Carlo error is an estimate of the difference between the mean of the sampled values which we are using as our estimate of the posterior mean for each parameter and the true posterior mean. As a rule of thumb, we should run simulations until the Monte Carlo error for each parameter of interest is less than 5% of the sample standard deviation. The Monte Carlo error (*MC error*) and sample standard deviation (*sd*) are reported in the *Node statistics* table which opens when we click *stats* box in the *Specification tool* window.

C. Node Summary Statistics of the Posterior Distributions

The posterior samples are summarized graphically by *kernel density plots* or numerically by *summary statistics* such as the mean, MC error, standard deviation and 2.5%, median and 97.5% quantiles of the sample.

To obtain node summaries of the monitored samples:

- Select Samples from the Inference menu.
- Type the parameter in the *node* box in the *Sample Monitor Tool* window or select from the pull down list or type * to select all monitored parameters.
- Type the iteration number that you want to start your summary from in the *beg* box in the *Sample Monitor Tool* window. This discards the pre-convergence *burn-in* samples.
- Click on the *stats* box. A table in *Node statistics* window reporting various summary statistics of the sampled values of the selected parameters appears.
- Click on the *density* box. A window showing *Kernel density* plots based on the sampled values of the selected parameter appears.

4. BUGS ANALYSIS OF OVERHEAD-COSTS IN RELATION TO MACHINE-HOURS OPERATED AND PRODUCTION-RUNS IN A PRODUCTION PROCESS

A plastic manufacturing firm uses injection molding equipment to manufacture plastic products. The management believes that machine-hours operated represents the most reasonable measure of the level of activity in the firm and is interested in getting a better understanding of the firm's overhead-costs to help estimate costs for the purpose of quoting prices on orders for existing and new products (Navidi 2006).

A. Data and Model Description

The firm keeps records for all production runs so that the direct material, direct labor, and machine-hour requirements for any product are readily predictable. Data records over 30 month period on overhead-costs, machine-hours operated and production-runs are considered in this analysis. No significant cost inflation or changes in production equipment have been noted over the period under reference (Navidi 2006).

The firm's management wishes to undertake an analysis of the behavior of overhead-costs to determine whether a relationship exists between the level of overhead-costs (y) with the number of machine-hours operated (x) and the number of production-runs(z). For WinBUGS illustration, we carry out the multiple regression analysis using the firm's production data records. The overhead-cost regression model is:

$$y_i = \alpha + \beta_1 x_i + \beta_2 z_i + \varepsilon_i; \quad i = 1,2,\dots,30, \qquad (4.1.1)$$

where α and $\beta's$ are the model parameters and ε is the random error. Graphical model for WinBUGS is shown in Figure 4.

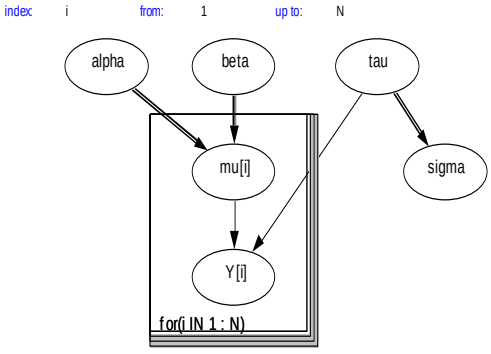


Figure 4. Model: $Y[i]$ depends on $MU[i]$, TAU. $MU[i]$ is a logical functio of model parameters $ALPHA$, $BETA$.

B. WinBUGS Application to Fit the Overhead-Cost Model

In the following we present the output obtained by performing the steps to fit the multiple regression model using WinBUGS and overhead-costs data.

1) WinBUGS Code for Model, Data and Initial values

Screenshots of the model syntax, data input and initial values are given in Figure 5. In Figure 5, we have comment (starting with #) defining the regression model with two explanatory variables. For sigma (standard deviation), we have used a non-informative *prior* that is uniform distribution. Precision parameter tau is the inverse of variance. In *data* list, n is the size of the data set, Y, X and Z denote overhead-costs, machine-hours and production-runs, respectively. We input the data using c( ) command and numerical values are separated using comma. For example, 30 data values of monthly production-runs denoted by Z variable are written as $Z = \mathrm{c}(31,22,39,\dots,26)$. Setting up the initial values: $\mathrm{list}(\mathrm{alpha} = 1, \mathrm{beta1} = 0, \mathrm{beta2} = 0, \mathrm{sigma} = 1)$, where alpha and (beta1, beta2) are the intercept and slop parameters, respectively. Sigma is the standard deviation.

2) Results and Output Discussions

WinBUGS sampling algorithm ensures that under regularity conditions, resulting sample converges to the *posterior* distribution of interest. Thus, before we summarize sample values, we must ensure that the chains have converged.

Parameter values that have been sampled at the beginning of the simulation are typically discarded so that the chain can *burn* in, or *converge* to its stationary distribution. Large, conservative *burn-in* periods are generally preferable to shorter *burn-in* periods. A 100,000 *update burn* followed by a further 50,000 *update burn* resulted in the following parameter estimates statistics.

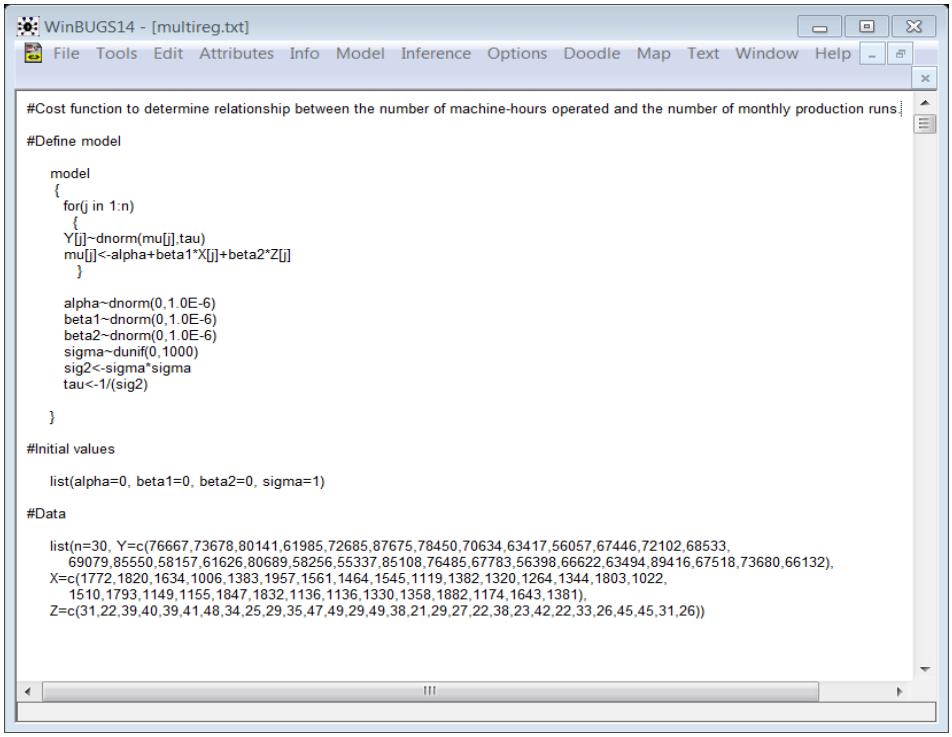
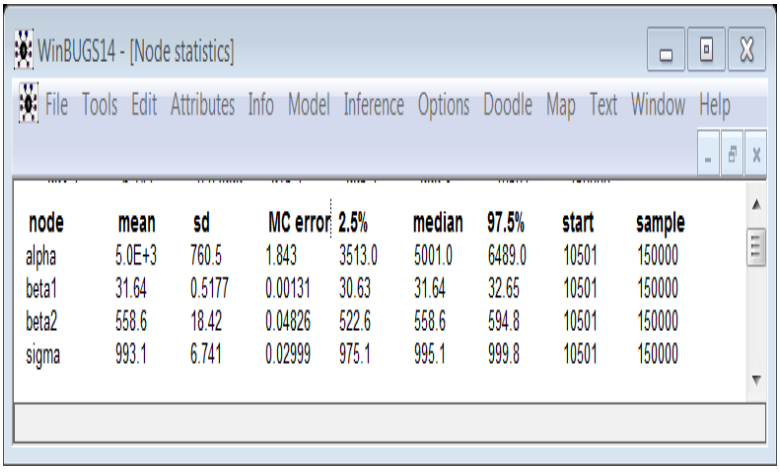


Figure 5. Model, data initial values.

The *posterior* samples are summarized numerically by *summary statistics* such as the mean, variance and quantiles of the sample in Figure 6, or graphically by the *kernel density plots* in Figure 7. Summary statistics of posterior samples can also be produced in box-plots. Figure 8 shows boxplots of alpha, beta1, beta2 and sigma.



node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	5.0E+3	760.5	1.843	3513.0	5001.0	6489.0	10501	150000
beta1	31.64	0.5177	0.00131	30.63	31.64	32.65	10501	150000
beta2	558.6	18.42	0.04826	522.6	558.6	594.8	10501	150000
sigma	993.1	6.741	0.02999	975.1	995.1	999.8	10501	150000

Figure 6. Summary statistics of posterior samples.

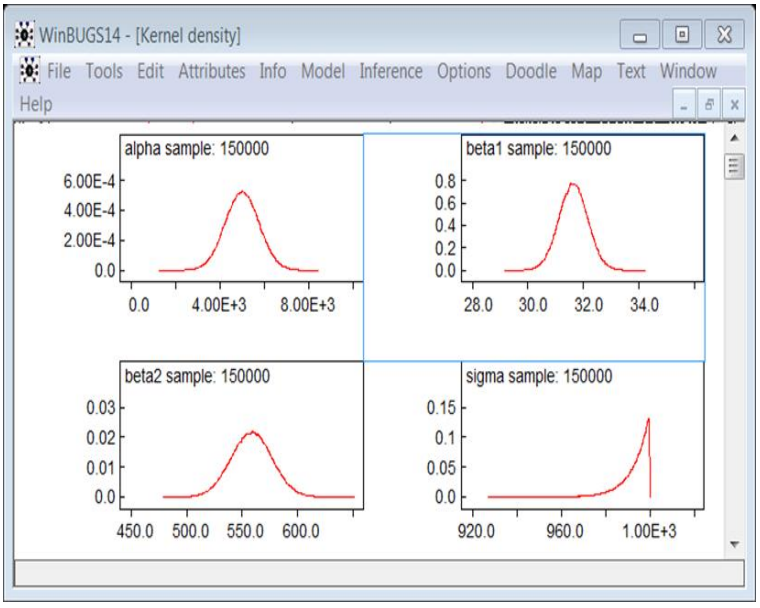


Figure 7. Kernel density plots.

For checking convergence, time series plots obtained from *history* box in the *Sample Monitor Tool* are commonly used to assess convergence. If the plot looks like a *horizontal band*, with no long *upward* or *downward* trends, then we have evidence that the chain has converged. In Figure 9, time plots indicate that chain convergences have been reached for alpha, beta1 and beta2. Autocorrelation plots, in Figure 10, assess the convergence. Higher values of autocorrelations in parameter chains often signify a model that is slow to converge. Figure 11 shows trace plots. It is indicative from trace and autocorrelation plots that chain convergence for parameters (alpha, beta1, beta2, sigma) looks reasonable.

In summary, WinBUGS analysis indicates that:

- The estimates of the regression coefficients of machine-hours operated (β_1) and monthly production-runs (β_2) are 31.64 and 558.6 with their standard errors respectively being 0.00131 and 0.04826.
- The 95% credible interval for β_1 is (30.63, 32.65) machine hours-operated and for β_2 is (522.6, 594.8) monthly-runs.
- The estimates of the standard deviation (sigma) of overhead-costs and its standard error are \$993.1 and \$0.02999. The 95% credible interval for standard deviation (sigma) is (\$975.1, \$999.8).
- The sampling distributions (kernel density) of beta0 (alpha), beta1 and beta2 are approximately normal.
- The estimated overhead-cost regression model using WinBUGS:

$$Overhead_{costs} = 5000 + 31.64(machine_{hours}) + 558.6(production_{runs});$$
$$Sigma = 993.1. \quad (4.2.2.1)$$

- The estimated overhead-cost regression model using least square method:

$$Overhead_{costs} = 11821.62 + 26.67(machine_{hours}) + 490.48(production_{runs});$$
$$Sigma = 2220.19. \quad (4.2.2.2)$$

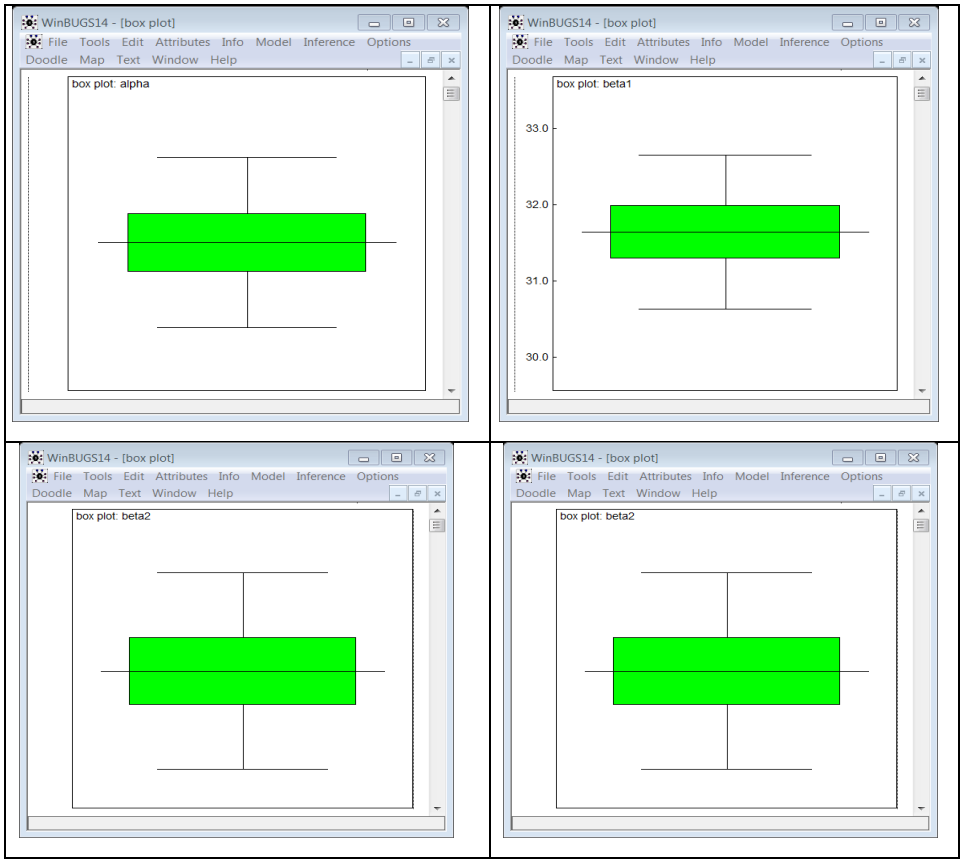


Figure 8. Box plots for model parameter estimates.

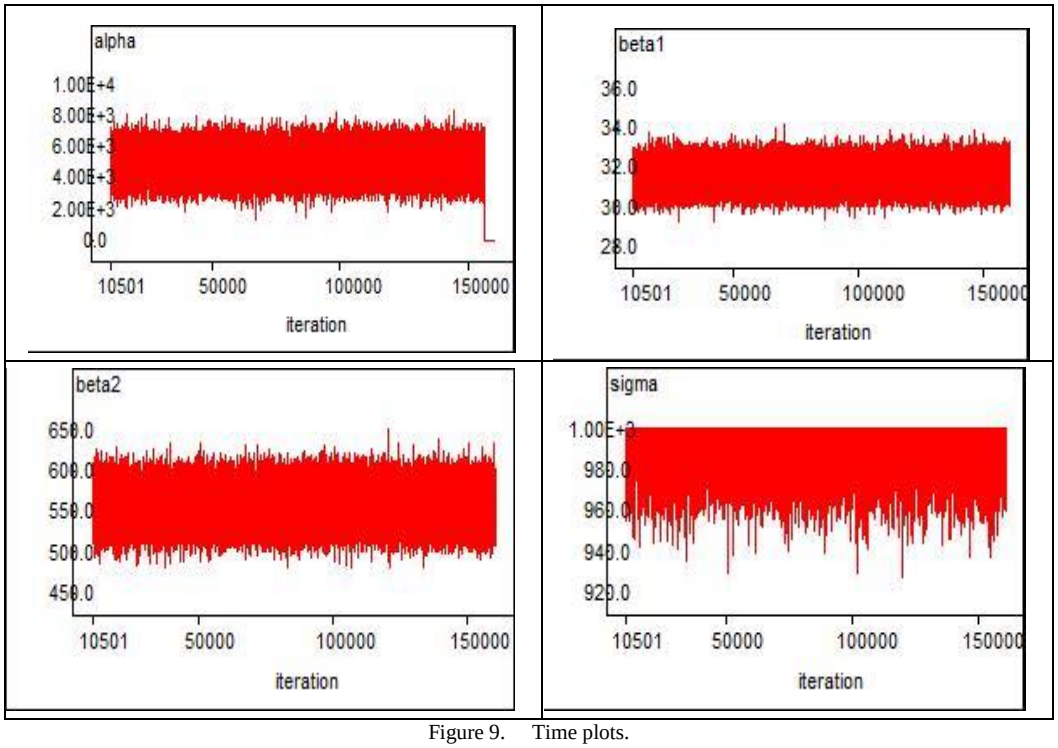


Figure 9. Time plots.

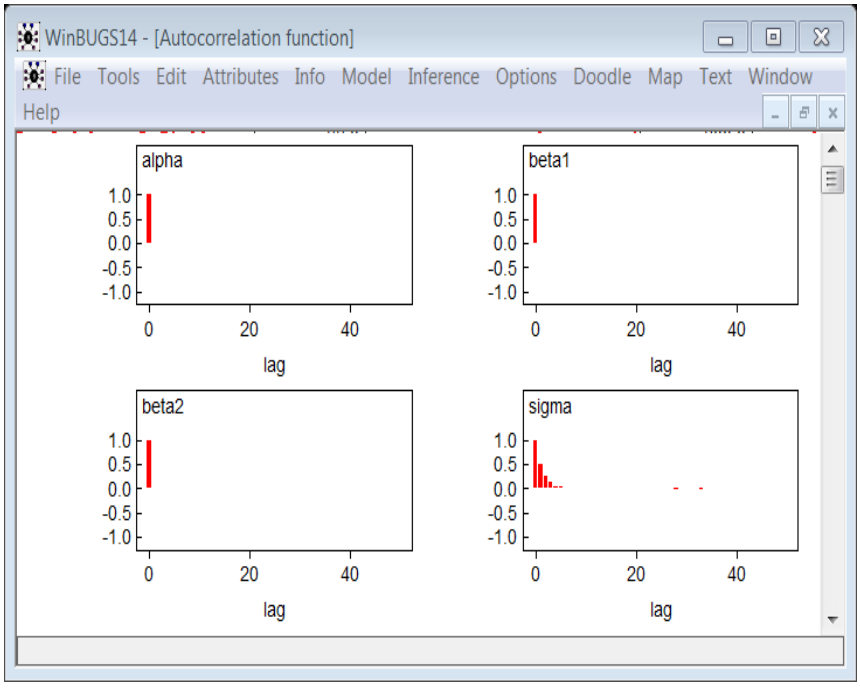


Figure 10. Autocorrelation plots.

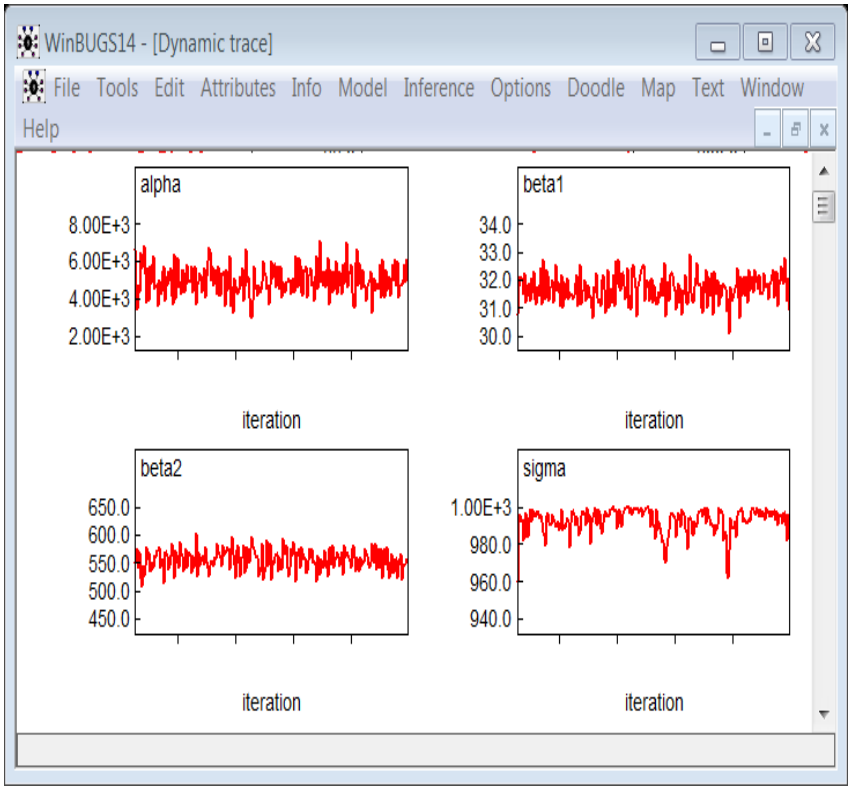


Figure 11. Trace plots.

5. BUGS Analysis of Relative Importance of Flux Variations due to Sampling and Analytic Errors using Differing Chemical Compositions in Flux Preparations

For the development and improvement of engineering methods, a metallurgist investigated four different fluxes for their relative importance by analyzing between fluxes variance and sampling variation in a welding experiment where welding fluxes had differing chemical compositions. A number of welds using each flux were made on AISI-1018 steel base metal. The hardness measurements of five welds using each of four fluxes, measured on the Brinell scale, were recorded (Navidi 2006).

A. Data and Model Description

It is noted that four sample means of hardness measurements differ. There is, of course, uncertainty in the sample means, however, question of interest is whether the sample means differ from each other by a greater amount than could be accounted for by uncertainty alone. We analyze between fluxes variation ($sigma2.between$) versus variation due to sampling and analytic ($sigma2.within$) using one-way analysis of variance. The one-factor variance component model using flux as the grouping factor:

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij}, i = 1, \ldots 4, j = 1, . . ,5, \quad (5.1.1)$$

where y_{ij} is the hardness measurement of the i-th flux and j-th weld, μ is the grand mean, α_i is the i-th flux-effect and ε_{ij} is the random error. The intraclass correlation coefficient (ICC) or the variance partition coefficient (VPC) symbolized by ρ and defined in terms of variance components is

$$\rho = sigma2.between/(sigma2.between + sigma2.within), \quad (5.2.2)$$

which means that in terms of percentages how much of the total variance is attributable to flux differences. Graphical model is depicted in Figure 12.

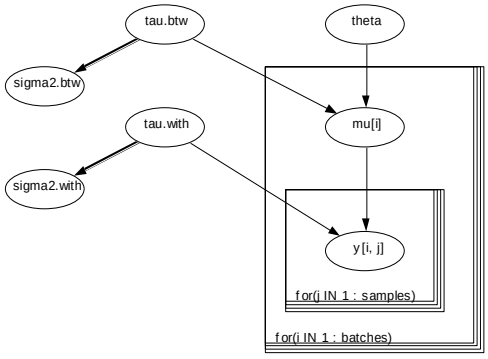


Figure 12. Model: y[i,j] depends on mu[i] and sigma2.with (tau.with) mu[i] is a functiojn of theta, siga2.btw (tau.btw).

B. WinBUGS Analysis of One-factor Variance Component Model

In the following we present the output obtained by performing the steps to fit the multiple regression model using WinBUGS and overhead-costs data.

1) WinBUGS Code for Model, Data and Initial values

In Figure 13, we have defined a model for one-factor variance component. For ICC (intra-class correlation), we have used a noninformative prior that is uniform distribution and for within-variation, prior is a gamma distribution. Precision parameter tau is the inverse of the variance. In data list, hardness measurements of four fluxes and five welds are entered in a 4x5 matrix format. The initial values:

$$\text{list(theta} = 250, \text{tau.within} = 1, \text{ICC} = 0.5). \quad (5.2.1.1)$$

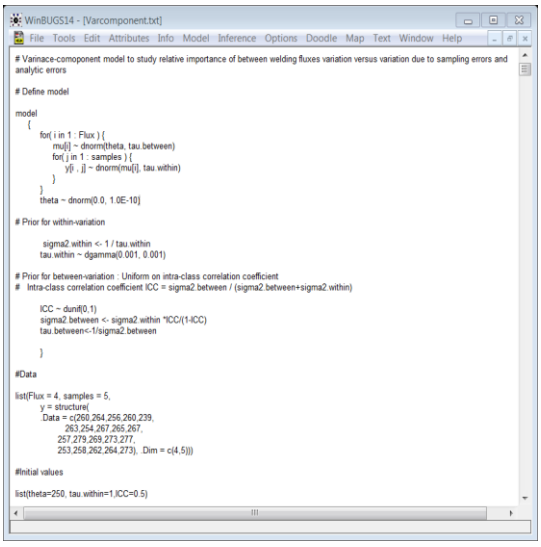


Figure 13. Model, data, initial values.

2) ***Results and Output Discussions***

A 100,000 *update burn* followed by a further 400,000 *update burn* resulted in the lower table of node statistics in Figure 14. We noted a relatively long run was necessary because of the high autocorrelation between successively sample values of some parameters especially sigma2.between and sigma2.within. The MC errors of both parameter estimates were higher than 5%. We further have 1,500,000 *update burns.* Node statistics, ignoring first 100,000 update burns, are in the upper table of Figure 14, or graphically by kernel density plots in Figure 15.

We noted that the MC errors for all parameter estimates except sigma2.between were smaller than 5%. Box-plots of hardness measurements are given in Figure 16. For checking convergence, in Figure 17, time plots indicate that chain convergences have been reached for ICC, sigma2.within and theta. However, the posterior distribution of sigma2.between has a very long right tail.

Autocorrelation plots in Figure 18 and trace plots in Figure 19 also assess convergence. Higher values of autocorrelations in parameter chains often signify a model that is slow to converge. It is obvious from high autocorrelation of sigma2.between that chain convergence is very slow. The MC error changed very little from 0.2558 in 500,000 runs to 0.15 in 1,500,000 runs.

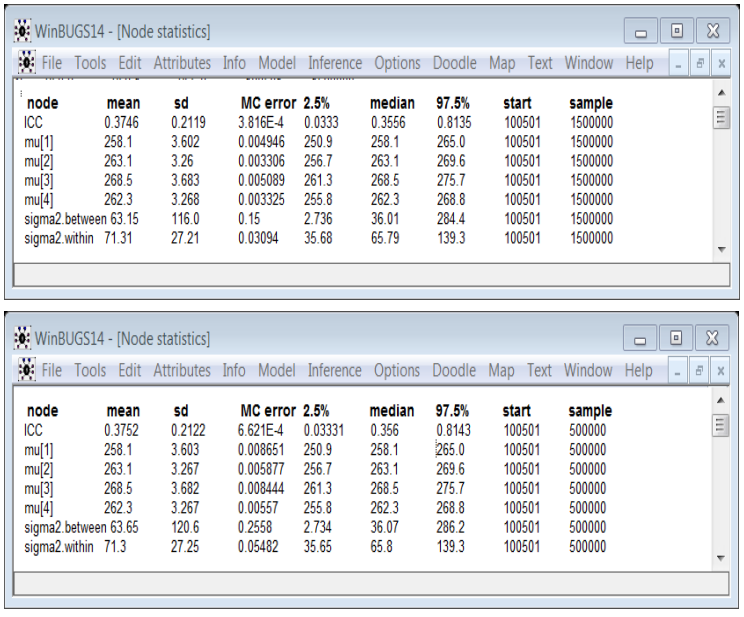


Figure 14. Node statistics.

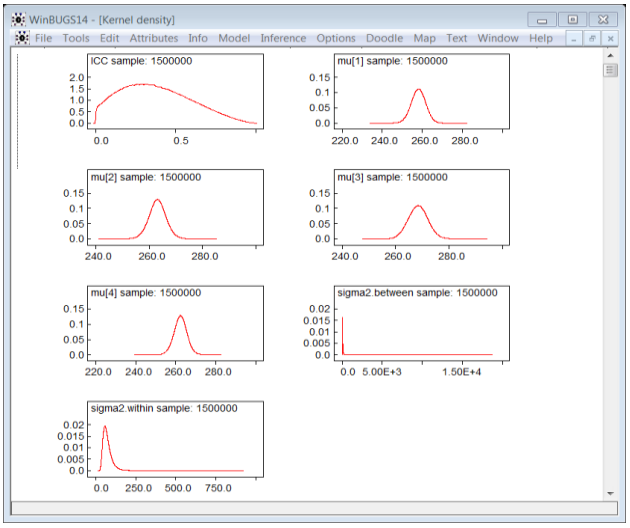


Figure 15. Kernel density plots.

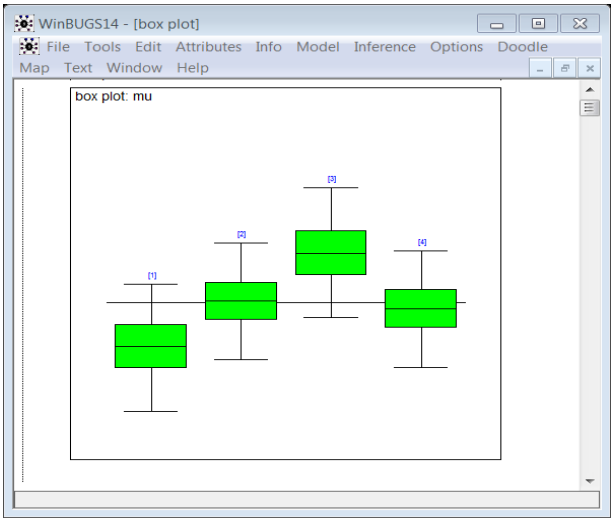


Figure 16. Box plots for hardness measurements.

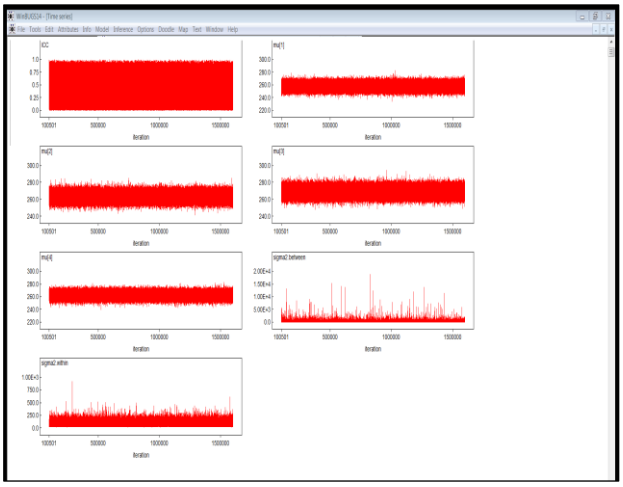


Figure 17. Time plots for chain convergence.

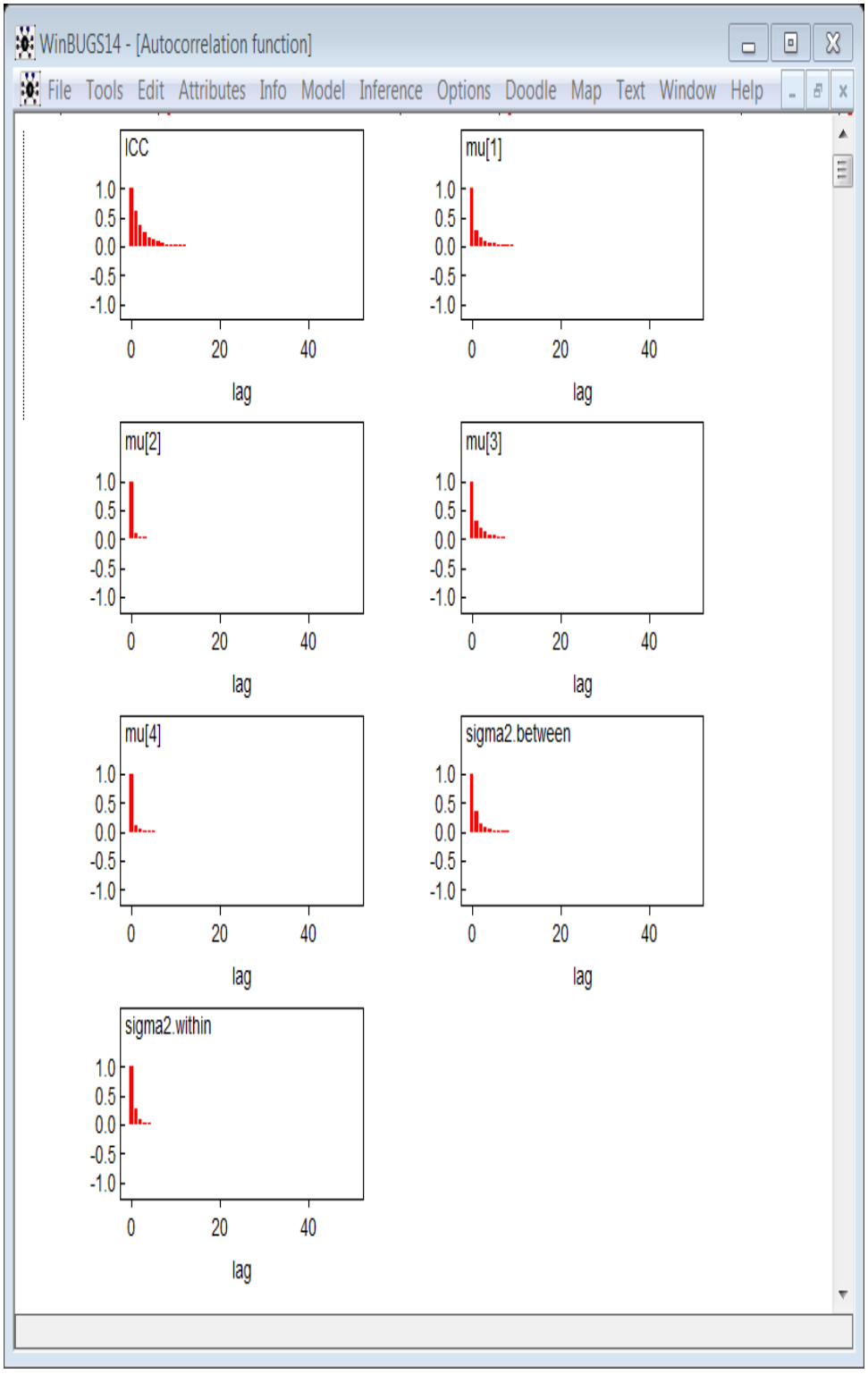


Figure 18. Autocorrelation plots.

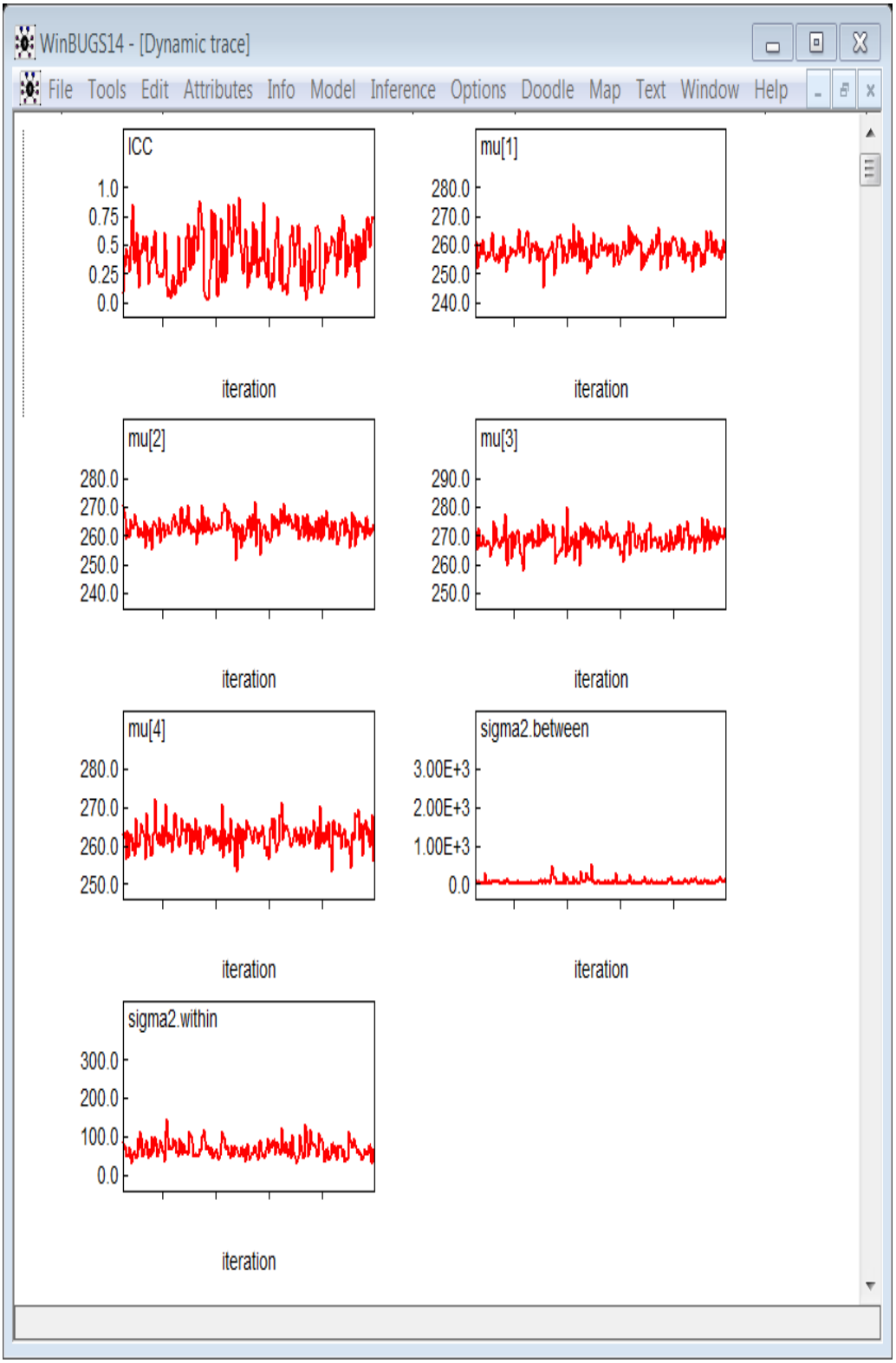


Figure 19. Trace plots.

Summary of the analysis of the node statistics with 1,500,000 runs in Figure 14 is as follows:

- The estimates of the mean hardness measurements on the Brinell scale of four fluxes are 258.1, 263.1, 268.5 and 262.3 with their standard errors respectively being 0.008651, 0.005877, 0.008444 and 0.00557. The 95% credible intervals respectively for the four flux means are: (250.9, 265.0), (256.7, 269.6), (261.3, 275.7) and (255.8, 268.8).
- The estimates of the intra-class correlation coefficient (ICC), sigma2.between and sigma2.within are 0.3746, 63.15 and 71.31 with their standard errors respectively being 0.0003816, 0.15 and 0.03094.
- There is a high order of between fluxes variability. The estimated standard error of sigma2.between is 15% after 1,500,000 iterations.
- The estimate of intra-class correlation 0.3746 in terms of percentages implies that 37.47 percent of the total variance is attributable to flux differences and remaining 53.01 percent to the error variability within the fluxes.
- The sampling distributions (kernel density) of mean hardness measurements on the Brinell scale of four fluxes are approximately normal (Figure 15).

6. CONCLUDING REMARKS

Hierarchical modeling techniques are often used as a popular tool which combines the estimation techniques and data mining techniques to model systems efficiently and ensuring reliability with multiple degrees of abstraction. A reliable model is often the one that remains sufficiently accurate over the whole input interval and provides a higher level of confidence in predictions. Reliability of a regression model is considered in terms of whether or not the model is able to predict accurately. The open-source software WinBUGS is a very useful Windows based computer program for the Bayesian analysis of complex statistical models using Markov Chain Monte Carlo numerical simulations. This is a powerful novel statistical analytical tool for performing the statistical analysis of engineering, industrial and scientific data using Bayesian inference and Markov Chain Monte Carlo (MCMC) sampling. In our paper, we have given a short introduction and description of WinBUGS technique and its implementation by analyzing real data sets from two industrial applications. The first application undertakes the multiple regression analysis to study the behavior of the overhead-costs with the number of machine-hours operated and the number of production-runs in a production process. In the second illustration, we carry out the variance-component analysis to investigate the relative importance of between fluxes variance and sampling variation in a weld experiment in which welding fluxes were prepared with differing chemical compositions.

Appendix 1. Commonly Used Probability Distributions

Distribution	Expression
Normal	x ~ dnorm(mu,tau); tau= 1/sigma*sigma
Log-normal	x ~ dlnorm(mu,tau)
Student - t	x~ dt(mu,tau,k)
Beta	x~ dbeta(a,b)
Gamma	x ~ dgamma(a,b)
Exponential	x~dexp(lambda)
Weibull	x~dweib(v,lambda)
Uniform	x ~ dunif(a,b)
Binomial	x~dbin(p,n)
Poisson	x~ dpois(lambda)
Multinomial	x~ dmulti(p[],N)
Multivariate Normal	x~ dmnorm(mu[],T[])
Multivariate Student-t	x ~ dmt(mu[], T[], k)

Appendix 2. Data Format

i. Rectangular format

x [ ]	y [ ]
47	0
148	18
...	...
360	24
END	

ii. S-Plus format

list(N=12,x= c(47,148,119,810,211,196,148,215,207,97,256,360),
y = c(0,18,8,46,8,13,9,31,14,8,29,24))

We need a "list" to indicate data which consist of mixture of scalars and vectors/arrays, or vectors/arrays of different lengths.

Appendix 3. WinBUGS (Other sources)

i. Interfaces for R, SPlus, SAS, Matlab.
http://www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml

ii. R2winbugs function for R is most developed- Reads in data, writes script, monitors output etc.
http://cran.r-project.org/src/contrib/Descriptions/R2WinBUGS.html

iii. OpenBUGS site provides an open source version including BRugs which works from within R. http://mathstat.helsinki.fi/openbugs/

REFERENCES

[1]. Gelman, J.C. Carlin, H. Stern, and D.B. Rubin, "Bayesian Data Analysis", Chapman and Hall, New York,1995.

[2]. Colosimo, and Enrique del Castilo, "Modern numerical methods in Bayesian computation", Bayesian Process Monitoring Control and Optimization, 2006.

[3]. Vidakovic, "Bayesian Inference using Gibbs Sampling – BUGS Project", Springer Texts in statistics, 2011.

[4]. B.P. Carlin, and T.A. Louis, "Bayes and Empirical Bayes Methods for Data Analysis", Chapman and Hall, London, 1996.

[5]. C-F Sheu, and S.L. O'Curry, "Simulation-based Bayesian inference using BUGS", Behavior Research Methods, Instruments & Computers,30,pp.232–237,1998. doi:10.3758/ BF03200649.

[6]. Lunn, C. Jackson, N. Best, Thomas, D. Spiegelhalter, "BUGS book: A practical introduction to Bayesian analysis",CRC Press / Chapman and Hall, 2012.

[7]. H. Ahmadi, T. Abdelzaher, J. Han, N. Pham, and R. K. Ganti, "The sparse regression cube: A reliable modeling technique for open cyber-physical systems", IEEE/ACM Second Int. Conf. Cyber-Physical Sys., pp.87-96, 2011.

[8]. H.S.B. Herath, "Crack spread option pricing with copulas", Jour. Econ. Finance, 02/01/2011.

[9]. H.S.B. Herath, and P. Kumar, "Using copula functions in Bayesian analysis: A comparison of the lognormal conjugate", Eng. Econ., 2014. doi: 10.1080 /0013791X.2014.962719.

[10]. King, "Programming in WinBUGS", Chapman & Hall/ CRC Interdisciplinary Statistics Series, 2009.

[11]. L. Breima, J. Friedman, R. Olshen, and C. Stone, "Classification and Regression Trees", Wadsworth International Group, 1984.

[12]. Lawson, "WinBUGS Basics", Statistics in Practice, 08/08/2003.

[13]. M. Srivastava, "Human –centric sensing", Phil. Trans. Roy. Soc. A, Math. Phys. Engg. Sci., 01/13/2012.

[14]. M.C. Wardle, "Amphetamine as a social drug: effects of d-amphetamine on social processing and behavior", Psychopharmacology, 04/13/2012.

[15]. P. Congdon, "Bayesian Statistical Modelling", John Wiley & Sons, Chichester, 2001.

[16]. P.J. Kelly, "A review of software packages for analyzing correlated survival data (Buyers Guide)", American Statistician, 01/13/2012.

[17]. R. Xi, "Compression and aggregation of Bayesian estimates for data intensive computing", Jour. Knowl. Inform., 11/29/2011.

[18]. R.S. Kaplan R S, and A.A. Atkinson, "Advanced Management Accounting",Prentice-Hall Inc, 1989.

[19]. S. Stringer, D. Borsboom, and E.J. Wagenmakers, "Bayesian inference for the information gain model", Behavior Research Methods 43(2), pp. 297–309, 2011. doi: 10.3758/s13428-010-0057-5.

[20]. S. Weisberg, "Applied Linear Regression", Wiley NY, 1980.

[21]. W. Navidi, " Statistics for Engineers and Scientists", McGraw Hill,2006.

[22]. W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, "Markov Chain Monte Carlo in practice",Chapman and Hall, London,1996.

[23]. Y.G. Chen, J. Han, J. Pei J, B.W. Wah, and J. Wang, "Regression cubes with lossless compression and aggregation", IEEE Trans. on Knowlegde Data Eng. 18(12), pp. 1585–1599, 2006.