

APPLICATION OF A GIBBS SAMPLER TO ESTIMATING PARAMETERS OF A  
HIERARCHICAL NORMAL MODEL WITH A TIME TREND AND  
TESTING FOR EXISTENCE OF THE GLOBAL WARMING

by

YEVHEN YANKOVSKYY

Specialist, Ukrainian Marine State Technical University, 1999  
M.A., Kyiv-Mohyla Academy, 2003  
M.S., Iowa State University, 2006

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Statistics  
College of Arts and Sciences

KANSAS STATE UNIVERSITY  
Manhattan, Kansas

2008

Approved by:

Major Professor  
Dr. Paul Nelson

## **Abstract**

This research is devoted to studying statistical inference implemented using the Gibbs Sampler for a hierarchical Bayesian linear model with first order autoregressive structure. This model was applied to global-mean monthly temperatures from January 1880 to April 2008 and used to estimate a time trend coefficient and to test for the existence of global warming. The global temperature increase estimated by Gibbs Sampler was found to be between  $0.0203^{\circ}\text{C}$  and  $0.0284^{\circ}\text{C}$  per decade with 95% credibility. The difference between Gibbs Sampler estimate and ordinary least squares estimate for the time trend was insignificant. Further, a simulation study with data generated from this model was carried out. This study showed that the Gibbs Sampler estimators for the intercept and for the time trend were less biased than corresponding ordinary least squares estimators, while the reverse was true for the autoregressive parameter and error standard deviation. The difference in precision of the estimators found by the two approaches was insignificant except for the samples of small sizes. The Gibbs Sampler estimator of the time trend has significantly smaller mean square error than ordinary least squares estimator for the smaller sample sizes studied.

This report also describes how the software package WinBUGS can be used to carry out the simulations required to implement a Gibbs Sampler.

## Table of Contents

List of Figures .....	v
List of Tables .....	vi
Acknowledgements .....	vii
Preface .....	viii
Chapter 1. Theoretical Background and Studies in the Field .....	1
1.1 Bayesian Paradigm .....	1
1.2. Theoretical Background of a Gibbs Sampler .....	1
1.3. Convergence Monitoring of the Gibbs Sampler Estimate .....	3
1.4. Frequentist Evaluation of the Bayesian Inferences .....	6
Evaluation of a Posterior Credible Set .....	6
Comparing the Properties of the Estimators .....	6
1.5. Hierarchical Bayesian Modeling .....	7
1.6. Studies on Global Warming .....	9
Chapter 2. Application of WinBUGS Software to Hierarchical Model Estimation .....	10
2.1. Running WinBUGS in a Menu/dialog Mode .....	11
2.2. Working with WinBUGS in a Batch Mode .....	16
Chapter 3. Research Results .....	20
3.1. Autoregressive Research Model .....	20
Hierarchical Bayesian Model for Testing Existence of the Global Warming .....	23
3.2. Estimation Results for the Research Models .....	24
OLS Estimation of Global-mean Monthly Temperature .....	25
Chapter 4. Simulation study comparing OLS estimators and Hierarchical Bayes estimators .....	26
4.1. Parameter Specification .....	26
4.2. Comparison of OLS Estimators and Hierarchical Bayes Estimators .....	27
Chapter 5. Conclusions .....	31
References .....	33
Appendix A. R output summary on OLS estimation of the research model .....	35

Appendix B. A code for estimation of the research model in WinBUGS .....	36
Appendix C. WinBUGS summary on Gibbs Sampler estimation for the research model .....	40
Appendix D. R code for simulation study on Gibbs Sampler and OLS estimation of the research model.....	41
Appendix E. A code with the model description from a file AR1Gamma.txt that was used in the simulation study .....	45
Appendix F. Estimation summary from a test WinBUGS run of the research model with a generated sample of 1000 observations .....	46
Appendix G. Testing hypothesis of equality of the OLS and Gibbs Sampler MSE estimates in the research model .....	47

## List of Figures

Figure 1.3.1.a. History of WinBUGS iterations for $\beta_2$ coefficient.....	5
Figure 1.3.1.b. History of WinBUGS iterations for $\beta_0$ coefficient .....	5
Figure 3.1.a. Plot of global-mean, monthly temperature from 01/1880 to 04/2008 .....	22
Figure 3.1.b. Histogram of global-mean, monthly temperature from 01/1880 to 04/2008 .....	22
Figure 3.1.c. Plot of autocorrelation function for the research temperature data .....	22
Figure 3.1.d. Plot of partial autocorrelation function for the research temperature data .....	22
Figure 3.2.a. History of WinBUGS iterations for $\beta_1$ .....	24
Figure 3.2.b. Posterior probability density function of the time trend coefficient, $\beta_1$ .....	24
Figure 4.1. History of WinBUGS iterations for $\beta_1$ .....	27
Figure 4.2. History of WinBUGS iterations for $\beta_2$ .....	27
Plot C.1. History of Winbugs iterations for $\beta_0$ .....	40
Plot C.2. History of Winbugs iterations for $\beta_1$ .....	40
Plot C.3. History of Winbugs iterations for $\beta_2$ .....	40
Plot C.4. History of Winbugs iterations for $\sigma_{\beta_0}$ .....	40
Plot C.5. History of Winbugs iterations for $\sigma_{\beta_1}$ .....	40
Plot C.6. History of Winbugs iterations for $\sigma_{\beta_2}$ .....	40
Plot C.7. History of Winbugs iterations for $\sigma_Y$ .....	40
Plot F.1. History of Winbugs iterations for $\beta_0$ .....	46
Plot F.2. History of Winbugs iterations for $\beta_1$ .....	46
Plot F.3. History of Winbugs iterations for $\beta_2$ .....	46
Plot F.4. History of Winbugs iterations for $\sigma_{\beta_0}$ .....	46
Plot F.5. History of Winbugs iterations for $\sigma_{\beta_1}$ .....	46
Plot F.6. History of Winbugs iterations for $\sigma_{\beta_2}$ .....	46
Plot F.7. History of Winbugs iterations for $\sigma_Y$ .....	46

## List of Tables

Table 4.1. Estimation results from 100 simulations with a sample size 10 data points.....	30
Table 4.2. Estimation results from 100 simulations with a sample size 30 data points.....	30
Table 4.3. Estimation results from 100 simulations with a sample size 100 data points.....	30
Table 4.4. Estimation results from 100 simulations with a sample size 1000 data points.....	30
Table F.1. WinBUGS summary on the estimators' posterior distributions.....	46

## **Acknowledgements**

I would like to thank my major advisor Dr. Nelson for his contribution in this research. His patient guidance, inspiring encouragement and valuable comments were very important for the progress of my paper. I am also thankful to Dr. Dubnicka, Dr. Song, Dr. Wang and the other instructors of the KSU Department of Statistics for their classes that helped me gain necessary knowledge in the field and sharpen my analytical skills. I also greatly appreciate Dr. Boyer's help with arranging teaching and research assistance for me. Without his support and involvement, my studying and graduation from Master's program in Statistics at KSU would be impossible.

I am very grateful to my parents and to my brother for their strong belief in me, inspiring encouragement, and invaluable support. Their involvement was a key factor for my success in studying at the Master's program and for the progress of this research.

## Preface

In a remarkably short period of time, the Gibbs Sampler, a type of Markov chain Monte Carlo simulation (hereinafter, MCMC) has emerged as an extremely popular and useful tool for the analysis of complex statistical models. This approach is often used in the field of Bayesian analysis, which requires evaluation of complex and often high-dimensional integrals to obtain posterior distributions for unobserved quantities of interest such as unknown parameters, missing data, and predicted observations. In many settings, alternative methodologies, like asymptotic approximation, numerical quadrature, or non-iterative Monte Carlo methods, are either infeasible or inaccurate.

This project is a simulation study using the Gibbs Sampler to carry out a Bayesian analysis and to assess its behavior using frequentist indicators. The original idea for my research was inspired by a hierarchical Bayesian modeling and analysis carried out by Scollnik (2001).

My project has three goals:

- (1) Modify Scollnick's model to incorporate a time trend and to apply this model to testing for the existence of global warming.
- (2) Describe the software package WinBUGS and how it can be used to implement the Gibbs Sampler.
- (3) Use the developed model to analyze global monthly temperature data, obtain the Gibbs Sampler estimates by using WinBUGS software, and draw conclusion about existence of global warming.
- (4) Design a simulation study to generate data from the research model with fixed parameters and use WinBUGS to estimate these model parameters.

In addition, I evaluate the behavior of the estimated posterior means obtained from the Gibbs Sampler in terms of bias and mean square error and compare them to the corresponding properties of the ordinary least squares estimators.



# Chapter 1. Theoretical Background and Studies in the Field

## 1.1 Bayesian Paradigm

Suppose, that an observable random variable  $\mathbf{Y}$  has a density function  $f(\mathbf{y} | \boldsymbol{\psi})$ , conditional on the parameter vector,  $\boldsymbol{\psi} = (\psi_1, \psi_2, \dots, \psi_p)$  lying in a set  $\mathbf{E}$ . Bayesian inference about parameter  $\boldsymbol{\psi}$  begins with specifying what is called a *prior* density  $\pi(\boldsymbol{\psi})$  on the parameter space  $\mathbf{E}$  which represents what is known or believed about  $\boldsymbol{\psi}$  before any data are collected. Hereinafter, I use symbols such as  $\pi(\cdot)$  to denote a probability density function of the variables appearing inside the parentheses. Having observed  $\mathbf{Y} = \mathbf{y}$ , a Bayesian analysis uses Bayes' rule to update what is now known about  $\boldsymbol{\psi}$  through what is called the *posterior* density  $\pi(\boldsymbol{\psi} | \mathbf{y})$  on  $\mathbf{E}$  having the form:

$$\pi(\boldsymbol{\psi} | \mathbf{y}) = \frac{f(\mathbf{y} | \boldsymbol{\psi})\pi(\boldsymbol{\psi})}{f(\mathbf{y})} \propto L_y(\boldsymbol{\psi})\pi(\boldsymbol{\psi}) \quad (1.1.1)$$

where  $L_y(\boldsymbol{\psi})$  is a data likelihood, and  $f(\mathbf{y}) = \int f(\mathbf{y} | \boldsymbol{\psi})\pi(\boldsymbol{\psi})d\boldsymbol{\psi}$  is called a marginal density of a variable  $\mathbf{Y}$ . A Bayesian might then use descriptions of  $\pi(\boldsymbol{\psi} | \mathbf{y})$  such as means, variances, modes, marginal distributions and sets in  $\mathbf{E}$  which have high posterior probability to summarize the posterior. Applied Bayesian analysis has been limited by the difficulty of obtaining these values, especially when the dimension of  $\boldsymbol{\psi}$  is large. The Gibbs Sampler and other simulation schemes, generally called *Markov Chain Monte Carlo Sampling (MCMC)*, which have mostly been developed over the last twenty years, provide algorithms for generating random variables that approximate observations from the posterior. Then, according to the laws of large numbers, sample averages can be used to summarize interesting features of the posterior.

## 1.2. Theoretical Background of a Gibbs Sampler

Scollnik (2001) defined MCMC sampling as a simulation technique for generating a dependent sample from a distribution of interest. Our goal here is to generate values  $\{\psi^{(i)}\}$  which asymptotically behave like observations from the posterior. Suppose that for  $j = 1, 2, \dots, p$ ,

it is relatively easy to generate observations from the conditional posterior distributions

$$\pi(\psi_j | \psi_k, k \neq j, \mathbf{y}).$$

A Gibbs Sampler produces a sequence of observations  $\{\psi_k^{(i)}\}$  as follows:

1. Select an arbitrary starting set of values of  $\psi^{(0)} = (\psi_1^{(0)}, \psi_2^{(0)}, \dots, \psi_p^{(0)})$

2. Having obtained  $\psi^{(i)}$ , simulate the sequence of random draws:

$$\begin{aligned} \psi_1^{(i+1)} &\sim \pi(\psi_1 | \psi_2^{(i)}, \dots, \psi_p^{(i)}, \mathbf{y}), \\ \psi_2^{(i+1)} &\sim \pi(\psi_2 | \psi_1^{(i+1)}, \psi_3^{(i)}, \dots, \psi_p^{(i)}, \mathbf{y}) \\ \psi_3^{(i+1)} &\sim \pi(\psi_3 | \psi_1^{(i+1)}, \psi_2^{(i+1)}, \psi_4^{(i)}, \dots, \psi_p^{(i)}, \mathbf{y}) \\ &\dots \\ \psi_p^{(i+1)} &\sim \pi(\psi_p | \psi_1^{(i+1)}, \psi_2^{(i+1)}, \dots, \psi_{p-1}^{(i+1)}, \mathbf{y}) \end{aligned}$$

3. Assign a unit-augmented value to the counter,  $i \leftarrow i+1$

4. Repeat steps 1-3  $m$  times.

Geman and Geman (1984) showed that under mild conditions the iterates  $\{\psi_k^{(i)}\}$  form a Markov chain whose stationary distribution is the desired posterior and that

$$\begin{aligned} \psi_k^{(i)} &\xrightarrow{d} \psi_k \sim \pi(\psi_k | \mathbf{y}) \\ i &\rightarrow \infty \end{aligned} \tag{1.2.1}$$

and for any intergrable function  $h$

$$\begin{aligned} \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{i=1}^m h(\psi_k^{(i)}) &\xrightarrow{a.s.} E(h(\psi_k^{(i)} | \mathbf{y})) \end{aligned} \tag{1.2.2}$$

According to Scollnik (2001), equation (1.2.1) implies that as the index  $i$  becomes moderately large, the value  $\psi_k^{(i)}$  is nearly a random draw from the distribution of interest  $\pi(\psi_k | \mathbf{y})$ . He suggested that the value of  $i$  should be at least 10-15. In addition, for some “large” integer  $d$  only values  $\psi^{(dl)}$ ,  $l = 1, 2, \dots$ , should be used to make the serial correlation for the values appearing in the subsequence negligible. This process of selecting only the  $d$  th element from a sequence is called thinning at the rate  $d$ .

The marginal density for each parameter  $\psi_s$  can be estimated by integrating the conditional distribution with respect to the other parameters which  $\psi_s$  is conditional on:

$$\hat{\pi}(\psi_j) = \frac{1}{m} \sum_{i=1}^m \pi(\psi_j | \psi_k = \psi_k^{(i)}; j \neq k) \quad (1.2.3)$$

Since, the first few observations  $\psi_j^{(i)}$  tend to have distribution quite different from the target one and the early part of the chain can significantly affect the average in (1.2.3), the first part of a chain, called a “burn-in”, is usually eliminated. The number of the observations to be discarded varies in practice and depends on a rate of chain’s convergence.

### 1.3. Convergence Monitoring of the Gibbs Sampler Estimate

Cowles and Carlin (1996) argued that sample convergence is a main issue in application of the MCMC Gibbs Sampler. The study of algorithm convergence is supposed to answer the question: At what point it is reasonable to believe that the samples are truly representative of the underlying stationary distribution of the Markov chain? The issue is complicated by the nature of the MCMC process which produces correlated observations. This correlation slows the algorithm in its attempt to sample from the entire stationary distribution and can adversely affect the determination of appropriate Monte Carlo estimates of model characteristics based on the estimation output.

Efforts at solving a problem of determining MCMC convergence algorithm have concentrated on two areas: a theoretical approach and applied diagnostic tools. Theoretical methods attempt to analyze the Markov transition kernel of the chain to find a number of iterations that will ensure convergence within a specified tolerance of the true stationary distribution. In most cases, sophisticated mathematical calculations resulted in quite loose bounds suggesting numbers of iterations beyond what is reasonable or feasible in practice. Therefore, almost all of the applied work involving MCMC methods relies on diagnostic tools, the second approach to the convergence problem. Although applied diagnostic tools do not always result in clear-cut answers because the stationary distribution is always unknown in practice, many statisticians still heavily rely on such diagnostics, believing that "a weak diagnostic is better than no diagnostic at all." (Cowles et al. (1996)). In this review, I will focus on simulations design and methods that are proven to be effective in monitoring convergence and feasible in WinBUGS at the same time.

### ***Applied Methods for Monitoring Convergence***

One of the most commonly used convergence diagnostics are methods for detecting mixing properties of Markov chain Samplers. These methods make use of the fact that most MCMC algorithms, including the Gibbs Sampler, have a random walk type behavior in which a simulated chain spreads out from the starting point to cover the space of the target distribution. Brooks and Gelman (2001) argued that convergence occurs when the chain has fully spread to the target distribution, which can be judged in three basic ways:

a) Efficient implementation of the simulation algorithm to avoid potential difficulties such as a Gibbs Sampler getting stuck in a limited region of the parameter space, for example in the area near local modes.

b) Monitoring trends to judge quality of mixing. Gelman and Rubin (1992) proposed monitoring mixing of simulated sequences by comparing the variance within each sequence to the total variance of the mixture of sequences.

c) Monitoring autocorrelation to obtain approximately independent simulation draws.

Brooks and Gelman (2001) noted that the methods stated above can fail in detecting lack of convergence of a slow-moving sequence.

### ***Simulations Design that Makes Monitoring Convergence more Reliable***

Brooks and Gelman (2001) argued that the most effective approaches to diagnosing convergence are based on design considerations. They recommended the following design rules that can improve convergence in average:

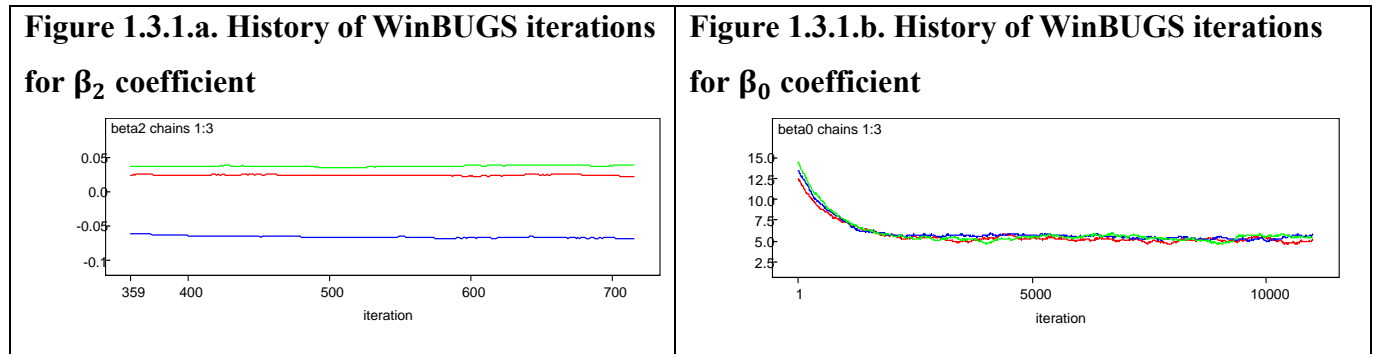
a) Simulating multiple sequences allows using between and within components to monitor between and within variance components.

b) Assigning overdispersed starting points allows a user to compare the increasing within-sequence variance to the decreasing between-sequence variance.

c) The local property of most Markov chain simulation algorithms allows identifying convergence with good mixing of the iterated chains.

In practice, these recommendations can be implemented in two ways. One of them is checking trace plots of the sample values versus iteration to look for evidence when the simulation appears to have stabilized. Another effective method of checking convergence is running multiple chains with different starting values. For example, Figures 1.3.1.a and 1.3.1.b below illustrate iteration

process for  $\beta_0$  and  $\beta_2$  coefficients with three different sets of initial values that were obtained in this research. Each chain of the generated values is marked by different color. Figure 1.3.1.a exhibits estimates lacking convergence, since each chain goes in near parallel fashion and likely results in significantly different estimates for  $\beta_2$  coefficient. In contrast, Figure 1.3.1.b shows an example of good convergence. Although the chains started at different initial points, they overlapped many times, and likely result in  $\beta_0$  estimates with insignificant difference.



To improve convergence, Gelman et al. (2004) recommended applying standardization of the covariates to have mean zero and standard deviation one, and considering better parameterization of the model to improve orthogonality of the joint posterior distribution. Since large autocorrelation between the subsequent elements of the Markov chain can be a reason of estimates' divergence, Neal (1995) offered using ordered over-relaxation of the parameters' distributions to suppress the autocorrelation in the chain. This method replaces an old value,  $\psi_i$ , by a new value,  $\psi_i'$ , in three steps:

- 1) independently generates  $K$  random values from the conditional distribution  $\pi(\psi_i | \{\psi_j\}_{j \neq i})$ ;
- 2) arranges these  $K$  values and the old value,  $\psi_i$ , in non-decreasing order:

$$\psi_i^{(0)} \leq \psi_i^{(1)} \leq \dots \leq \psi_i^{(r)} = \psi_i \leq \dots \leq \psi_i^{(K)}$$

where  $r$  is an index corresponding to the order of the old value;

- 3) selects a new value for a component  $i$ ,  $\psi_i' = \psi_i^{(K-r)}$ , where  $K$  is a parameter of this method.

Taking into account that each step of the method requires additional computational time proportional to  $K$ , Neal (1995) tried to find a minimum value of parameter  $K$  for different levels

of autocorrelation in the chain. He showed that even quite high autocorrelation can be effectively suppressed by setting a value of  $K$  to around 20.

### 1.4. Frequentist Evaluation of the Bayesian Inferences

I use objective frequentist criteria to evaluate the performance of estimators obtained by using a Bayesian approach. These frequentist properties are considered as being embedded in a sequence of repeated samples. According to Gelman et al. (2003), the Bayesian notion of stable estimation implies that for a fixed model, the posterior distribution approaches a point as more data arrive and leads in the limit to inferential certainty. Thus, if the hypothesized family of probability models contains a true distribution and assigns it a nonzero prior density, then as more information about parameter vector  $\Psi$  arrives, the posterior distribution converges to the true value of  $\Psi_0$ . Specifically, I will judge an estimator  $\hat{\psi}$  in terms of its finite sample size bias and mean square error, defined by

$$Bias[\hat{\psi}(\mathbf{y})] = E_{\psi}[\hat{\psi}(\mathbf{y}) | \Psi_0] - \Psi_0 \quad (1.4.1)$$

where  $\hat{\psi}(\mathbf{y})$  is an estimate of  $\Psi_0$  and a function of available data  $\mathbf{y}$ .

$$MSE[\hat{\psi}(\mathbf{y})] = E_{\psi}[(\hat{\psi}(\mathbf{y}) - \Psi_0)^2 | \Psi_0] \quad (1.4.2)$$

#### *Evaluation of a Posterior Credible Set*

A set  $S$  in the parameter space  $E$  is called a  $1-\alpha$  credible set for  $\psi$ , if having observed  $Y = \mathbf{y}$ ,

$$P(\psi \in S | \mathbf{y}) = 1-\alpha.$$

I will evaluate how well such credible sets behave as  $1-\alpha$  confidence sets. Specifically, I will use simulation to estimate how close the attained coverage rate of a  $1-\alpha$  credible set is to  $1-\alpha$ .

#### *Comparing the Properties of the Estimators*

I will assess and compare the frequentist properties of the Gibbs Sampler estimator to the properties of the corresponding ordinary least squares (OLS) estimator. For an estimator  $\hat{\theta}$  of a parameter  $\theta$ , I will use simulation to obtain  $N$  values  $\{\hat{\theta}_i\}$  from independently generated data sets under the distribution determined by  $\theta$  and let  $u_i = (\hat{\theta}_i - \theta)^2, i = 1, 2, \dots, N$ . Then, I will compute the following terms:

$$\widehat{MSE} = \sum_{i=1}^N \frac{(\hat{\theta}_i - \theta)^2}{N} = \frac{1}{N} \sum_{i=1}^N u_i = \bar{u}$$

$$SE(\widehat{MSE}) = \sqrt{\frac{\sum_{i=1}^N (u_i - \bar{u})^2}{N(N-1)}}$$

Approximate  $(1-\alpha)\%$  confidence interval for the MSE is then given by

$$\widehat{MSE} \pm z_{\alpha/2} SE(\widehat{MSE})$$

It is also reasonable to test hypothesis of equality of two competing MSE estimators based on independent random samples, the OLS MSE estimator and the Gibbs Sampler MSE estimator:

$$H_0: MSE_{OLS} = MSE_{Gibbs}$$

$$H_A: MSE_{OLS} \neq MSE_{Gibbs}$$

with a test statistic

$$T = \frac{\widehat{MSE}_{Gibbs} - \widehat{MSE}_{OLS}}{\sqrt{SE(\widehat{MSE}_{Gibbs})^2 + SE(\widehat{MSE}_{OLS})^2}} \sim t_{(n_{Gibbs} + n_{OLS} - k), (1-\alpha/2)}$$

where

$n_{Gibbs}$  is a size of the samples that were used in simulations to get Gibbs Sampler MSE estimate,  $\widehat{MSE}_{Gibbs}$ ;

$n_{OLS}$  is a size of the samples that were used in simulations to get OLS MSE estimate,  $\widehat{MSE}_{OLS}$ .

$k$  is a number of estimates to be estimated by Bayesian and OLS approaches to estimate the test statistic,  $T$ . In this case, there are 4 parameters to be estimated: MSE mean and variance for both Bayesian and OLS methods.

## 1.5. Hierarchical Bayesian Modeling

Broadly speaking, Bayesian hierarchical modeling arises in the context of (1.1.1) when the prior density  $\pi(\psi) = \pi(\psi | \eta)$  is specified conditional on unobserved parameters  $\eta$  that are themselves viewed as random variables having a known density, denoted  $p(\eta)$ . The joint posterior distribution of parameters  $(\psi, \eta)$  in (1.1.1) can then be expressed as

$$\pi(\psi, \eta | \mathbf{y}) = \frac{f(\mathbf{y} | \psi) \pi(\psi | \eta) p(\eta)}{f(\mathbf{y})} \propto L_y(\psi) \pi(\psi | \eta) p(\eta) \quad (1.5.1)$$

where now  $f(\mathbf{y}) = \iint f(\mathbf{y} | \psi) \pi(\psi | \eta) p(\eta) d\psi d\eta$ .

This approach allows greater flexibility in expressing what is known about the parameter  $\psi$  before any data are collected. This two level hierarchical model can be extended to a third level by specifying a prior density on  $\eta$ . This process of adding levels can be continued. Typically, a diffuse prior is used for the final level.

For example, Scollnik (2001) applied hierarchical approach to modeling the workers' insurance relative claim frequency denoted by  $X_{ij}$  for a class of workers  $i$  in a year  $j$ . He assumed that the claim frequencies are conditional on latent variables  $\{\theta_i\}$  and  $\tau_1^2$ , independent and distributed according to the following model:

**Level 1:**

$$X_{ij} | \theta_i, \tau_1^2 \sim N(\theta_i, 1/\tau_1^2), i = 1, 2, \dots, 133, j = 1, 2, \dots, 7 \quad (1.5.2)$$

Reasonably arguing that different workers' insurance classes should relate to each other, Scollnik assumed that the estimates of the latent class means  $\theta_i$  represent a sample from the population of all workers' classes means, and there is no prior information to distinguish one class mean from another. Therefore, Scollnik assigned a distribution of higher level on the parameters of the claim frequency distribution:

**Level 2:**

$$\theta_i | \mu, \tau_2^2 \sim N(\mu, 1/\tau_2^2)$$

$$\tau_1^2 \sim \text{Gamma}(0.001, 0.001)$$

To model vague knowledge of the parameters of the class means' distribution, Scollnik applied non-informative diffuse prior distributions on the precision hyperparameter,  $\tau_1^2 = 1/\sigma^2$ , and on the mean of the class means distribution:

**Level 3:**

$$\mu \sim N(0, 1000)$$

$$\tau_2^2 \sim \text{Gamma}(0.001, 0.001)$$

Further, applying Gibbs Sampler method, Scollnik estimated the parameters of the hierarchical model (1.5.2) for the first six years  $\psi^* = \{\theta_i^* = (\theta_{i1}^*, \theta_{i2}^*, \dots, \theta_{i6}^*), \tau_i^*\}, i=1, \dots, 133$ , and made predictions about year seven.



## 1.6. Studies on Global Warming

According to the free, on-line encyclopedia Wikipedia, global warming refers to the increase in the average temperature of the Earth's near-surface air and oceans in recent decades and its projected continuation. The term global warming is an example of the broader term "climate change," which can also refer to another term "global cooling." Although the United Nations Framework Convention on Climate Change (2007) uses the term "climate change" for human-caused change, and "climate variability" for other changes, in my research I will focus just on change in temperature at the Earth's surface over time, ignoring further investigation of potential reasons of these changes. Researchers investigating climate changes mostly focused on collecting reliable data, applying appropriate data transformation, and modeling time trend by using simple linear regression model. Thus, Angel (1988) applied normal linear least squares regression with a time trend to year-average temperature deviations. He estimated that over the 30-year interval, from 1958 to 1987, a global temperature increase at the Earth's surface was  $0.08^{\circ}\text{C}$  per decade and significant only at the 5% level. Over the 15-year interval, from 1973 to 1987, a global temperature increase was significant and equal  $0.28^{\circ}\text{C}$  per decade.

Before using simple linear regression with a time trend, Oort and Liu (1992) smoothed the data first by applying a filter with weights  $1/4$ ,  $1/2$ , and  $1/4$  to successive seasonal values, except for the beginning and the end points of the period from May 1963 to December 1989. Their estimated 95% confidence intervals for the temperature increase were between  $0.07^{\circ}\text{C}$  and  $0.15^{\circ}\text{C}$  per decade in the Northern Hemisphere, and between  $0.22^{\circ}\text{C}$  and  $0.36^{\circ}\text{C}$  per decade in the Southern Hemisphere. Jones (1994) applied simple linear regression to the surface data and found significant increase in temperature by about  $0.1^{\circ}\text{C}$  per decade from 1958 to 1993.

## **CHAPTER 2. Application of WinBUGS Software to Hierarchical Model Estimation**

WinBUGS1.4 (hereinafter, WinBUGS) is a free and publicly available software developed by a self-organized group of researchers concerned with flexible software for Bayesian analysis of complex statistical models using MCMC methods. This project, named the Bayesian inference Using Gibbs Sampling (BUGS) project, was launched in 1989 in the Medical Research Committee and Advisory Council (MRC) Biostatistics Unit in Cambridge, UK. A new development included also the supplementary OpenBUGS project in the University of Helsinki, Finland. Further details on the project and the software itself are available at <http://www.mrc-bsu.cam.ac.uk/bugs/>.

WinBUGS is a useful tool for estimating Bayesian models because of two reasons. First, WinBUGS has the ability to estimate the posterior distributions of the parameters based only on their marginal prior distributions and data model specified by a researcher. Thus, WinBUGS is an invaluable tool in many cases when the joint and marginal posterior distributions of the hierarchical Bayesian model are complex and hard to derive by hand. Second, WinBUGS accommodates numerous iterations for all parameters simultaneously within a short period of time and allows keeping track of simulations with different initial values of the parameters. This iterations monitoring accommodates checking convergence of the parameters and simplifies proving validity of the Bayesian estimates. However, WinBUGS does not attempt to verify if the posterior distribution is a proper distribution. Therefore, the posterior should be additionally checked to see if it is proper after iteration process is over.

To estimate Gibbs Sampler parameters, WinBUGS needs the following information to be supplied by a user:

- a) a data model;
- b) a data sample;
- c) parameters that should be estimated;
- d) initial values of the input variables having prior distributions assigned to, which are also called stochastic nodes;
- e) a number of iterations to simulate;
- f) a number of samples to be generated simultaneously;

g) a thinning rate with which an observation is selected into a final sample, or Markov chain, from a sequence of the generated values.

There are two ways of supplying the input information into WinBUGS:

- a) entering the inputs by pointing at the necessary parts of a code and clicking at the buttons with commands in a menu/dialog interface of WinBUGS;
- b) scripting all commands for WinBUGS in a separate file and running the script in another software in a batch mode.

Both modes require a data model, a data sample, and initial values coded in a specific BUGS programming language and supplied into WinBUGS interface. The output information can be stored either in the interface log window or saved in a separate log file. A striking advantage of the dialog mode is an opportunity to monitor iteration process and convergence of the estimates. However, the scope of using this method in simulation study is quite limited, since each run of WinBUGS should be made manually by a researcher. In contrast, the batch mode allows running decent number of simulations by invoking WinBUGS automatically from R or other software. In this case, a number of the simulations is naturally limited by the computational capacity of a computer running the simulation code. However, controlling convergence in all simulations using the batch mode is a complex technical issue that can be resolved by an advanced programmer. One of the proxy solutions to checking convergence in simulations is to use WinBUGS in the batch mode for all simulations in combination with running the same model once in a dialog mode. This test run in the dialog mode can provide useful information about flow of the data generation process in general and can help identify some problems that can be common to all iterations run in the batch model.

## **2.1. Running WinBUGS in a Menu/dialog Mode**

There are two options for running scripts in WinBUGS models in the menu/dialog mode. The most conventional one is saving a model and a script with the other necessary inputs in two separate files. Another option is highlighting parts of a single script file containing all necessary information and running them in the WinBUGS interface step by step. These methods have the same sequence of steps but differ slightly in coding. I describe below running WinBUGS in a menu/dialog mode from a single file on the example of my research model. In this example, I use

the first five observations of the applied data set. The exact code with full data set that I used in the research is provided in Appendix B.

An algorithm for estimation my research model in WinBUGS is as follows:

1. Open a file containing a model script by pointing to “File” option on the tool bar of WinBUGS interface and clicking the “Open” option once with left mouse button (hereinafter, LMB);

2. Make WinBUGS check that the model description fully defines a probability model in the next steps:

a) Highlight a part of the code containing description of the model, constants and variables used. The corresponding part of the code below starts with a word “model”, then describes constants and variables, and specifies the model with prior distributions on all model parameters, and ends with the final squiggle bracket in the attached code. To note, a sign # make WinBUGS skip a part of the row following this sign that can be used to provide brief explanatory information.

```
# 1. Research model with conjugate priors, a linear trend and AR(1) term
model
const
T=5;

var
Y[T], t;

{
for(t in 2:T)
{
Y[t] ~ dnorm(theta.Y[t], tau.y)
theta.Y[t] <- beta0+ beta1*t + beta2*Y[t-1]
}
}

# Prior
beta0 ~ dnorm(beta0.c, beta0.tau) #non-informative diffuse normal distn
beta1 ~ dnorm(beta1.c, beta1.tau) #non-informative diffuse normal distn
beta2 ~ dnorm(beta2.c, beta2.tau) #non-informative diffuse normal distn
tau.y ~ dgamma(0.001,0.001)
sigma.y <-pow((1/tau.y), 0.5)

beta0.c ~ dnorm(0, 1.0E-6) #non-informative diffuse normal distn
beta0.tau ~ dgamma(0.001, 0.001)
sigma.beta0 <- pow((1/beta0.tau), 0.5)
```

```

beta1.c ~ dnorm(0, 1.0E-6) #non-informative diffuse normal distn
beta1.tau ~ dgamma(0.001, 0.001)
sigma.beta1<-pow((1/beta1.tau), 0.5)

beta2.c ~ dnorm(0, 1.0E-6) #non-informative diffuse normal distn
beta2.tau ~ dgamma(0.001, 0.001)
sigma.beta2<-pow((1/beta2.tau), 0.5)
}

```

b) Point to “Model” option on the tool bar and highlight the “Specification...” option.

c) Check syntax of the model by clicking once with LMB on the “check model” button in the Specification Tool window. A message saying "model is syntactically correct" should appear in WinBUGS Log window.

3. Load the data by doing the next steps:

a) Highlight a part the file containing data and starting with a command word *list* and ending with the last bracket in the row:

```
# 2. Data
```

```
list(T = 5, Y = c(14.24, 14.1,13.89,13.83,13.87))
```

b) Click once with the LMB on the “load data” button in the Specification Tool window.

A message saying "data loaded" should appear in the WinBUGS Log window.

4. Select a number of chains to be simulated by typing a corresponding number in a white box following a “num of chains” note in the Specification Tool window. It is sensible to run several chains with different initial values to check convergence of the MCMC simulations. In my case, I initiated running three chains.

5. Compile the model by clicking once with the LMB on the “Compile” button in the Specification Tool window. A message saying "model compiled" should appear in the WinBUGS Log window. This step specifies internal data structure and specific MCMC updating algorithms to be used by WinBUGS for the particular model.

6. Provide WinBUGS with initial values for each stochastic node that are declared in the model statement. To load the initial values, you will need to

a) Highlight the code starting with the word *list* and ending with the last bracket of the first set of initial values:

```
list(beta0=0, beta1=0, beta2=0, beta0.c=0, beta1.c=0, beta2.c=0, beta0.tau=1,
beta1.tau=1, beta0.tau=1, beta2.tau=1, tau.y=1)
```

b) Click once with the LMB on the “load inits” button in the Specification Tool window. A message saying "chain initialized but other chain(s) contain uninitialized variables" should appear in the WinBUGS Log window.

c) Repeat the steps a) and b) for the second portion of initial values:

```
list(beta0=0, beta1=0, beta2=0, beta0.c=0, beta1.c=0, beta2.c=0, beta0.tau=0.01,
beta1.tau=0.01, beta2.tau=0.01, tau.y=0.01)
```

A message saying "chain initialized but other chain(s) contain uninitialized variables" should appear again in the WinBUGS Log window.

d) Repeat the steps a) and b) for the third portion of initial values:

```
list(beta0=0, beta1=0, beta2=0, beta0.c=0, beta1.c=0, beta2.c=0, beta0.tau=0.0001,
beta1.tau=0.0001, beta2.tau=0.0001, tau.y=0.0001)
```

A message saying "model is initialized" should now appear in the WinBUGS Log window.

To note, that there is no need in providing a list of initial values for every parameter in a model. You can have WinBUGS to generate initial values for any stochastic parameter by clicking with the LMB on the “gen inits” button in the Specification Tool window. In this case, WinBUGS will generate initial values by forward sampling from the prior distribution for each parameter. However, it is recommended to provide your own initial values for parameters having vague prior distributions assigned to avoid wildly inappropriate values generated by WinBUGS.

7. Set monitors to store the sampled values for the selected parameters by the sequence of the following steps:

a) Click on Inference option in the WinBUGS menu first, and then point to Sample option.

b) In the opened Sample Monitor window, type the name of each parameter in the white box following a word “node”. In my model, I gave the following names to the nodes: beta0, beta1, beta2, sigma.beta0, sigma.beta1, sigma.beta2, sigma.y.

c) Set a number of the iterations to burn-in in the white box following a “beg” note and the total number of iterations in the white box following an “end” note. Since the parameter estimates of my research model were lacking convergence in near 2000 first iterations, I specified 2000 iterations to be burn-in out of 30000 in total.

d) Click “set” button you enter information at steps a) and b)

8. Now WinBUGS is set to start a simulation that can be run in the following steps:

a) Select the “Update...” option from the Model menu.

b) In the opened window Update tool, type the number of updates or iterations in the white box following “updates” note, then enter a number of iteration after which you want to make a refreshment of the distribution in white box following “refresh” note, and finally type a thinning rate in the white box “thin”. In my case, I used the 30000 updates with refreshment of the iteration process after each 100 iterations, and a thinning rate of 3 iterations.

c) Click once on the “update” button. The program will start simulating values for each parameter in the model that may take a few seconds. At the same time, the box “iteration” will show how many updates have currently been completed.

d) When the number of updates in the box following “iteration” note reach the target number of iterations stated in the box near the “end” note, the iteration process will be over. At that moment, graphical and numerical summaries on iteration process for all parameters will be set for monitoring.

9. In order to check convergence and obtain posterior summaries of the model parameters, you should click Inference option in WinBUGS interface first, and then point to “Samples...” option. To see the summary of iteration process of a parameter, you will need to enter the name of this parameter in the white box following a note “node” in the Sample Monitor menu. There are the following summary options available in the menu:

a) An option “stats” describes posterior distribution by the mean, standard deviation, MC error, median, 2.5<sup>th</sup> and 97.5<sup>th</sup> percentiles.

b) An option “history” produces a plot of all parameter values in a sequence they were generated. If more than one chain run simultaneously, a history plot will show each chain in a different color. In this case, a researcher can be reasonably confident in convergence of the parameter estimator if all chains mix well.

c) An option “density” develops a graph of the density function for the estimated parameter.

d) An option “coda” provides a list of all generated values in the chain.

If all steps of the algorithm were made correctly, WinBUGS should produce an output that is similar to the iteration summary provided in Appendix C.

## **2.2. Working with WinBUGS in a Batch Mode**

Scripting instructions in a special language and calling the WinBUGS interface from the other software allow running WinBUGS programs without pointing at the specific parts of a code and clicking your way through the commands. This feature significantly simplifies using WinBUGS by automating routine analysis and helps using WinBUGS in simulations. An opportunity of working with WinBUGS in a batch mode is now available in R, Stata, SAS, Matlab, and Excel software.

To make use of the scripting language for a specific problem, a minimum of four pieces of information are required:

- 1) representation of a model in WinBUGS coding language;
- 2) a data set of observed or simulated values;
- 3) initial values for each chain;
- 4) a script with the commands combining the model, the data, and the initial values in an iteration algorithm.

In practice, these four pieces can be provided either in one file or in several files, depending on complexity of the model, size of the data set, and convenience to a user. Each file may be either in a native WinBUGS format (.odc) or in the text format (.txt). I show below an application of WinBUGS in a batch mode in R in a way I used it in my simulation study.

To make the WinBUGS session open in R, a special package called R2WinBUGS should be uploaded in the R version that is installed at the computer. This package provides opportunity of calling a model, summarize inferences and monitor convergence in a table and graphs, and save the simulations in arrays for easy access in R. The WinBUGS batch mode consists of the next basic steps:



1. Write an R code describing a model and prior distributions assigned to each stochastic model parameter in a separate file. A file AR1GammaModel.txt with detailed description of my research model in R is provided in Appendix E.

2. Go to R interface and prepare a script containing R commands entering the data and initial values, uploading a model file, and making WinBUGS run iterations and summary reports. In my simulation code, a piece of code with these commands is provided below, while the complete simulation script is attached in Appendix D.

```
# Data loading
data<-list("T","Y")

# Declaration of the model parameters to be iterated
parameters<-c("beta0", "beta1", "beta2", "sigma.y", "sigma.beta0", "sigma.beta1",
"sigma.beta2")

# Loading initial values for 3 MCMC chains
inits1<-list(beta0=0, beta1=0, beta2=0, tau.y=1, beta0.tau=1, beta1.tau=1, beta2.tau=1)
inits2<-list(beta0=0, beta1=0, beta2=0, tau.y=0.01, beta0.tau=0.01, beta1.tau=0.01,
beta2.tau=0.01)
inits3<-list(beta0=0, beta1=0, beta2=0, tau.y=0.0001, beta0.tau=0.0001, beta1.tau=0.0001,
beta2.tau=0.0001)
inits<-list(inits1, inits2, inits3)

# Invoking an auxiliary sub package "arm" of the R2WinBUGS package
library("arm")

# Applying bugs function entering inputs and running WinBUGS from R
AR1bugs.sim<- bugs(data, inits, parameters,
"D:/KSU_Research/Final/AR1GammaModel.txt", n.chains=3, n.iter=5000,
bugs.directory="c:/Program Files/WinBUGS14/",
working.directory="D:/KSU_Research/Final/Results/", clearWD=FALSE, debug=FALSE)

# Storing BUGS simulation results in the R interface operation memory
attach.bugs(AR1bugs.sim)
```

In the code above, *bugs* function plays a key role in running WinBUGS from R interface. This function takes data, initial values and assigns parameters of iteration process as inputs, automatically writes a WinBUGS script, calls the model, and saves the simulations results. In order to make this function run, a set of the function parameters should be defined:

*data* is a named list of variables having data and used in the WinBUGS model. In the code above, the data inputs are the data lists named Y and T that were earlier assigned a data set and a number of observations in this data set respectively.

*inits* is a R set of initial values for each iteration chain. Before, I assigned 3 sets called *inits1*, *inits2*, and *inits3* that include initial values for all model stochastic parameters for each of three chains iterated further by WinBUGS. A number of the Markov chains should corresponds to the *bugs* function argument *n.chain* declaring a number of chains to be generated by WinBUGS. In this, case *n.chain* is assigned a value 3. Alternatively, if *inits=NULL*, initial values are generated by WinBUGS.

*parameters* is a character vector of the names of the stochastic parameters which WinBUGS should take records on. In my code, these parameters are *beta0*, *beta1*, *beta2*, *sigma.y*, *sigma.beta0*, *sigma.beta1*, and *sigma.beta2*.

The fourth argument of the *bugs* function must be a note containing a path and a file name that contains description of the model in WinBUGS code. The extension of the model file can be either “.bug” or “.txt”. In my code, a note specifying a file with the model description is `D:/KSU_Research/Final/AR1GammaModel.txt`

*n.iter* is a number of total iterations per chain including a number of burn-in iterations that is 5000 iterations in my code.

*bugs.directory* is a directory that contains the WinBUGS executable files that were stored at `c:/Program Files/WinBUGS14/`.

*working.directory* sets a directory storing the WinBUGS' input and output files during execution of this function that in my case was `D:/KSU_Research/Final/Results/`.

*clearWD* is a logical argument indicating whether the WinBUGS' input and output files should be removed from the operation memory after WinBUGS has finished. A logical value FALSE assigned to argument *clearWD* in my code orders WinBUGS save all output in separate files in working directory assigned above.

*debug* is a logical argument with a default FALSE value making WinBUGS close automatically when the script has finished running, otherwise WinBUGS remains open for further investigation.

Further *bugs* arguments takes default values, if a user does not assign any specific values to them. In my simulation code, the following arguments are assigned default values:

*n.burnin* is a number of iterations in a sample of generated values to be burnt in at the beginning. The default for burn-in values is discarding the first half of the simulations,  $n.iter/2$ , that in my case is 2500 iterations.

*n.thin* is a thinning rate and must be a positive integer. The default is maximum value selected between 1 and rounded down to a value calculated using a formula:  $(n.chains * (n.iter - n.burnin) / 1000)$ . The default thin rate will be assigned if there are at least 2000 simulations. In my case, a thinning rate is 5.

*bin* is a number of iterations between saving of results. Default value used in my code is to save only at the end of the iteration process.

DIC is a logical argument with a default TRUE value computing deviance, pD, and DIC.

*digits* is a number of significant digits used for WinBUGS input.

*codaPkg* is a logical argument with a default FALSE value that arrange a bugs output object to be returned. If the TRUE value assigned, the file names of WinBUGS output are returned for easy access by the coda package through function `read.bugs`.

*bugs.seed* is a random seed for WinBUGS with a default no seed value.

*summary.only* with a default TRUE value limits the WinBUGS output to a short parameter summary for a quick analyses. In my case, temporary created files are not removed.

*save.history* with a default TRUE value generate trace plots for the declared parameters of interest.

3. When a part of the code stated above runs, a WinBUGS window will appear and R interface will freeze up. At this time the model will now run in WinBUGS. Running a model might take awhile, depending on the number of iterations and complexity of the model. You will be able to see details of the iterations exhibited in the log window within WinBUGS. When WinBUGS finishes iterations, its window will close and R will work again.

4. If an error message appears at the log window, it is recommended to re-run the model in R with debug option set in "TRUE" value. After this change in debug settings, the batch mode will provide all WinBUGS output summary and graphs to the screen. Examining the tables and graphs can help identifying potential problem in coding.

## Chapter 3. Research Results

### 3.1. Autoregressive Research Model

Although, in common usage, the term “global warming” is often referred to human influence on the global climate, my research is focused on change in temperature at the Earth’s surface over time without digression into further investigation of potential causes of these changes. Santer et al. (1996) argued that the thermal structure of the atmosphere is not the same in the Northern and Southern hemispheres primarily due to an asymmetry in observed ozone changes. Therefore, stratospheric cooling in the Northern Hemisphere could be different from cooling in the Southern Hemisphere. However, in order to simplify the exposition here, I ignored these potential differences and combined the data from both hemispheres. I used global-mean monthly, annually, and seasonally adjusted temperature at the Earth’s surface in 0.01 degrees Centigrade ( $^{\circ}\text{C}$ ) from January 1880 to April 2008 that were provided by NASA’s Goddard Institute for Space Studies<sup>1</sup>. Since NASA’s data were in  $0.01^{\circ}\text{C}$  and relative to the base period temperature of  $14^{\circ}\text{C}$ , I transformed the original data by multiplying the official data by 100 and adding 14 to obtain an absolute monthly global-mean temperature in degrees Centigrade. These data are presented below in Figures 3.1.a – 3.1.d. In modeling monthly temperature data, I took into consideration autocorrelation function and partial autocorrelation function estimated for the available data. The autocorrelation function of order  $k$ , or  $\text{ACF}(k)$ , gives the gross correlation, or  $\rho_k$ , between the temperature at time  $t$ ,  $y_t$ , and the temperature at lag  $k$ ,  $y_{t-k}$ :

$$\rho_k = \frac{\text{Cov}(Y_t, Y_{t-k})}{\text{Var}(Y_t)} \quad (3.1.1)$$

A theoretical autocorrelation function is estimated in a sample by what is called a correlogram, given by:

$$r_k = \frac{\sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2} \quad (3.1.2)$$

A partial autocorrelation function of order  $k$ , or  $\text{PACF}(k)$ , is a correlation between  $y_t$  and  $y_{t-k}$  net of intervening effects of the lagged temperatures observed between these two observations:

---

<sup>1</sup> The data used in this research are extracted from the official web-site of GISS NASA:

<http://data.giss.nasa.gov/gistemp/tabledata/GLB.Ts+dSST.txt>

$$\rho_k^* = \text{Corr}[y_t - E^*(Y_t|y_{t-1}, \dots, y_{t-k+1}), y_{t-k}] \quad (3.1.3)$$

where  $E^*(Y_t|y_{t-1}, \dots, y_{t-k+1})$  is the minimum mean-squared error predictor of  $Y_t$  by  $y_{t-1}, \dots, y_{t-k+1}$ . In practice, PACF( $k$ ) is estimated using a linear predictor of  $E^*(Y_t|y_{t-1}, \dots, y_{t-k+1})$ :

$$\hat{\rho}_k^* = \text{Corr}[y_t - \hat{\beta}_1 y_{t-1} - \hat{\beta}_2 y_{t-2} - \dots - \hat{\beta}_{k-1} y_{t-k+1}, y_{t-k}] \quad (3.1.4)$$

where  $\hat{\beta} = [\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_{k-1}] = \{\text{Var}[y_{t-1}, y_{t-2}, \dots, y_{t-k+1}]\}^{-1} \text{Cov}[y, (y_{t-1}, y_{t-2}, \dots, y_{t-k+1})]'$ .

In the NASA's monthly temperature data set, the estimated ACF on Figure 3.1.c is decaying slowly with the highest correlation observed between the current observation and its first lag, around 0.8. A graph of the estimated PACF at Figure 3.1.d exhibits the first nine PACF coefficients irregularly declining from near 0.8 to 0.1, marginally significant coefficient at lag 12, and two significant PACF coefficients at lag 20 and lag 24. This oscillating pattern of PACF with near a dozen of significant partial correlation coefficients and slow decay in ACF are evidence of potential non-stationarity and significant time trend in the monthly temperature data. Although the true data generating process likely have an intricate autoregressive structure, I chose a linear model with a time trend and the first order autoregressive term denoted by AR(1) because in numerous empirical studies the first order autoregression is proved to be a reasonable model even for very complex underlying processes (Greene (2003)). Therefore, my research model is the first reliable pass to the next comparative analysis of OLS and hierarchical Bayesian estimates, while further refinement of the model by applying autoregressive terms of higher order is left for the future work in the field. Let

$$Y_t|y_{t-1}, \beta_0, \beta_1, \beta_2 = \beta_0 + \beta_1 t + \beta_2 y_{t-1} + \varepsilon_t \quad (3.1.5)$$

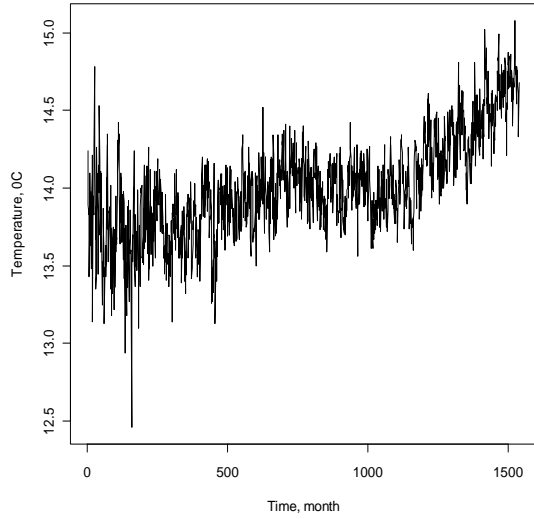
$$Y_t|y_{t-1}, \beta_0, \beta_1, \beta_2 \sim N_1(\beta_0 + \beta_1 t + \beta_2 y_{t-1}, \sigma_Y^2)$$

where  $\{\varepsilon_t\}$  is a white noise time-series process where each element of the sequence is jointly independent and normally distributed:  $\varepsilon_t \sim N(0, \sigma_Y^2)$ ,  $-\infty < t < +\infty$ ;

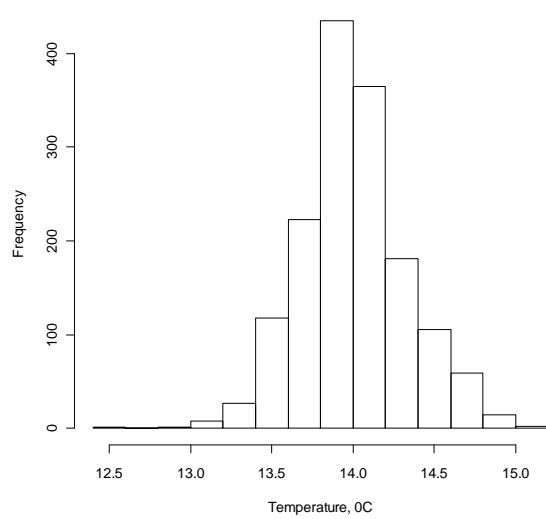
$\beta_1$  represents a linear trend in the means as time, denoted by  $t$ , increases.

Using model (3.1.5), I will build a Bayesian hierarchical model (3.1.6).

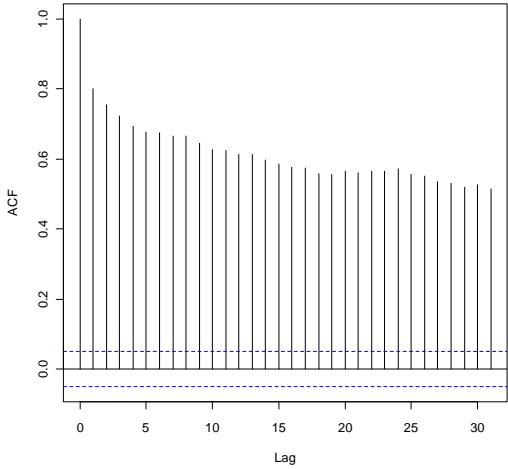
**Figures 3.1.a. Plot of global-mean, monthly temperature from 01/1880 to 04/2008**



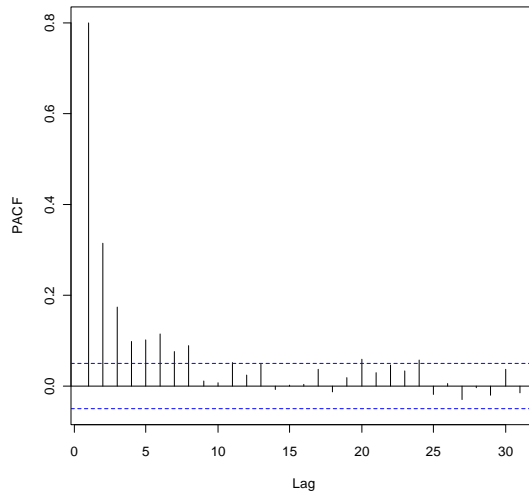
**Figure 3.1.b. Histogram of global-mean, monthly temperature from 01/1880 to 04/2008**



**Figure 3.1.c. Plot of autocorrelation function for the research temperature data**



**Figure 3.1.d. Plot of partial autocorrelation function for the research temperature data**



## ***Hierarchical Bayesian Model for Testing Existence of the Global Warming***

### **Level 1:**

$$Y_t | y_{t-1}, \beta_0, \beta_1, \beta_2, \sigma_Y^2 \sim N_1(\beta_0 + \beta_1 t + \beta_2 y_{t-1}, \sigma_Y^2) \quad (3.1.6.)$$

### **Level 2:**

$$\tau_Y = \frac{1}{\sigma_Y^2} \sim \text{Gamma}(0.001, 0.001)$$

$$\beta_0 | \mu_{\beta_0}, \sigma_{\beta_0}^2 \sim N_1(\mu_{\beta_0}, \sigma_{\beta_0}^2)$$

$$\beta_1 | \mu_{\beta_1}, \sigma_{\beta_1}^2 \sim N_1(\mu_{\beta_1}, \sigma_{\beta_1}^2)$$

$$\beta_2 | \mu_{\beta_2}, \sigma_{\beta_2}^2 \sim N_1(\mu_{\beta_2}, \sigma_{\beta_2}^2)$$

### **Level 3:**

$$\mu_{\beta_0} \sim N_1(0, 10^6)$$

$$\tau_{\beta_0} = \frac{1}{\sigma_{\beta_0}^2} \sim \text{Gamma}(0.001, 0.001)$$

$$\mu_{\beta_1} \sim N_1(0, 10^6)$$

$$\tau_{\beta_1} = \frac{1}{\sigma_{\beta_1}^2} \sim \text{Gamma}(0.001, 0.001)$$

$$\mu_{\beta_2} \sim N_1(0, 10^6)$$

$$\tau_{\beta_2} = \frac{1}{\sigma_{\beta_2}^2} \sim \text{Gamma}(0.001, 0.001)$$

where  $\text{Gamma}(a, b)$  denotes a gamma distribution with a shape parameter  $a$  and a scale parameter  $1/b$ .

WinBUGS specification of a normal distribution requires two parameters: mean and precision parameter,  $\tau = 1/\sigma^2$ . A gamma distribution of the precision parameter,  $\tau$ , implies an inverse scaled  $\chi^2$  distribution of the variance,  $\sigma^2$ . The precise form of the diffuse priors is believed to be not very important in this case. Since the collected data comprise rather large sample, 1539 observations, the data likelihood should overwhelmingly dominate in forming posterior distribution. In the model (3.1.2), parameters of interest are as follows:

$$\Psi = (\beta_0, \beta_1, \beta_2, \mu_{\beta_0}, \mu_{\beta_1}, \mu_{\beta_2}, \sigma_{\beta_0}^2, \sigma_{\beta_1}^2, \sigma_{\beta_2}^2, \sigma_Y^2)$$

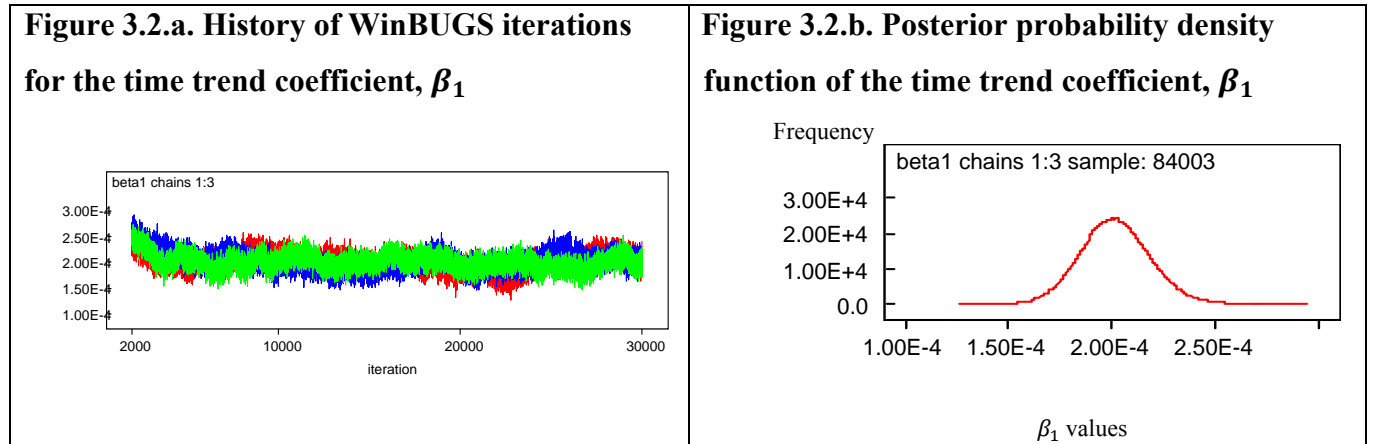
The joint posterior distribution for the parameters of interest follows equation (3.1.3):

$$\pi(\beta_0, \beta_1, \beta_2, \mu_{\beta_0}, \mu_{\beta_1}, \mu_{\beta_2}, \sigma_{\beta_0}^2, \sigma_{\beta_1}^2, \sigma_{\beta_2}^2, \sigma_Y^2 | Data) \propto f(\mu_{\beta_0})f(\mu_{\beta_1})f(\mu_{\beta_2})f(\sigma_{\beta_0}^2)f(\sigma_{\beta_1}^2)f(\sigma_{\beta_2}^2) \times f(\beta_0 | \mu_{\beta_0}, \sigma_{\beta_0}^2)f(\beta_1 | \mu_{\beta_1}, \sigma_{\beta_1}^2)f(\beta_2 | \mu_{\beta_2}, \sigma_{\beta_2}^2)f(\sigma_Y^2) \prod_{t=1}^T f(y_t | \beta_0, \beta_1, \beta_2, \sigma_Y^2) \quad (3.1.3)$$

A marginal posterior distribution for each parameter of interest can be found in theory by integrating joint posterior distribution over the other parameters. However, there are problems here in deriving an exact form for a joint posterior distribution in equation (3.1.3) and a marginal posterior distribution for each parameter because of the complex form of the model. In this case, WinBUGS will estimate these distributions by using approximations.

### 3.2. Estimation Results for the Research Models

Initial trial runs of the model in WinBUGS showed that it took near 2000 iterations of the model parameters' estimates to converge. Therefore, I assigned 2000 iterations out of 30000 in total to be burnt in. The final estimation results presented below are based on the WinBUGS output excluding burn-in iterations and having a thinning rate of 3 iterations. The iteration history for all Gibbs Sampler estimates suggests about quite good mixing of the iterations in the chains and convergence of the estimates. Estimation summary of the hierarchical Bayesian model and plots of the  $\beta_1$  distribution formation are provided below, while detailed WinBUGS output summary on all model parameters is provided in Appendix C.



$$Y_t | y_{t-1}, \beta_0, \beta_1, \beta_2, \sigma_Y^2 \sim N_1(\beta_0 + \beta_1 t + \beta_2 y_{t-1}, \sigma_Y^2)$$

$$\overline{E}(Y_t)^{Gibbs} | y_{t-1} = \hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta}_2 y_{t-1} = 5.383 + 0.000202t + 0.6047y_{t-1}$$

$$\text{MC error } (0.0167) \quad (0.00006) \quad (0.00123)$$



$$\widehat{\sigma_Y^2}^{Gibbs} = E(\widehat{\sigma_Y^2})^{Gibbs} = 0.184$$

The estimated 95% credibility interval for the time trend coefficient is (0.000169, 0.000237). Since a plot of the posterior distribution of the time trend coefficient  $\beta_1$  on Figure 3.2.b. shows that 100% of the distribution of  $\beta_1$  estimates is above 0 values, there is strong evidence of global warming. The 95% credibility interval for  $\beta_1$  also suggest that there is 95% chance that the global temperature increase is between 0.000169°C and 0.000237°C per month, or between 0.0203°C and 0.0284°C per decade.

### ***OLS Estimation of Global-mean Monthly Temperature***

A brief summary of the estimation results for the OLS model is given below, while detailed WinBUGS output summary is attached in Appendix C. Now letting  $\{\hat{\beta}_i, i = 0, 1, 2\}$  denote ordinary least squares estimators:

$$\begin{aligned} \widehat{E}(Y_t)^{OLS} | y_{t-1} &= \hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta}_2 y_{t-1} = 5.283 + 0.0002t + 0.612y_{t-1} \\ R_{adjusted}^2 &= 0.6805 \quad \text{SE} \quad (0.274) \quad (0.000015) \quad (0.020) \\ \widehat{\sigma_Y^2}^{OLS} &= \text{MSE} = 0.182 \end{aligned}$$

The approximate 95% confidence interval for the time trend coefficient includes values from 0.000103 to 0.000339.

Testing hypothesis for existence of the global warming, we obtain:

$$\begin{aligned} H_0: \beta_1 &= 0 \\ H_1: \beta_1 &\neq 0 \\ T &= \frac{\hat{\beta}_1 - 0}{se(\hat{\beta}_1)} = 13.59 \sim t_{n-2} = t_{1537} \\ p\text{-value} &= \text{Prob}(|t_{1537}| \geq 13.59) < 2 * 10^{-16} \end{aligned}$$

Since the  $p$ -value of the time trend coefficient,  $\beta_1$ , is less than 1%, there is strong evidence of the global warming. At 95% confidence, the global temperature tends to grow from 0.000103°C to 0.000339°C per month or 0.0124-0.0406°C per decade. Since 95% credibility interval and 95% confidence interval overlap, there is no evidence of significant difference in the Gibbs Sampler estimate and OLS estimate.

# Chapter 4. Simulation study comparing OLS estimators and Hierarchical Bayes estimators

## 4.1. Parameter Specification

In the simulation study, I generated samples of 10, 30, 100, and 1000 observations from the model (3.1.1) with the regression parameters fixed at the specific values:

$$\beta_0 = 0, \beta_1 = 1, \beta_2 = 0.5, \sigma_Y^2 = 1.$$

Given different sample sizes and data generated from the model (3.1.1), the Bayesian and OLS estimators were obtained, and their performance was evaluated in terms of bias and mean square error. The Bayesian simulations were run for three chains starting with the following initial values:

1) The first set of the initial values, hereinafter initials 1:

$$\beta_0 = 0, \beta_1 = 0, \beta_2 = 0, \sigma_{\beta_0}^2 = 1, \sigma_{\beta_1}^2 = 1, \sigma_{\beta_2}^2 = 1.$$

2) The second set of the initial values, hereinafter initials 2:

$$\beta_0 = 0, \beta_1 = 0, \beta_2 = 0, \sigma_{\beta_0}^2 = 100, \sigma_{\beta_1}^2 = 100, \sigma_{\beta_2}^2 = 100.$$

3) The third set of the initial values, initials 3:

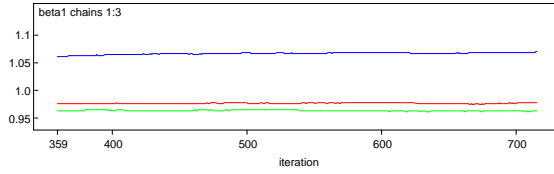
$$\beta_0 = 0, \beta_1 = 0, \beta_2 = 0, \sigma_{\beta_0}^2 = 1000, \sigma_{\beta_1}^2 = 1000, \sigma_{\beta_2}^2 = 1000.$$

Since starting values for  $\sigma_Y^2$  were not assigned, they were randomly selected by WinBUGS from the model prior distribution such that the precision parameter,  $\tau_Y$ , following the Gamma distribution:

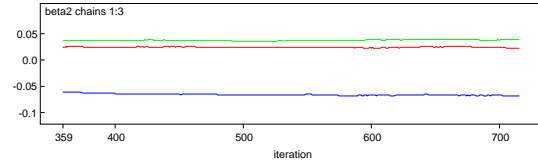
$$\tau_Y = \frac{1}{\sigma_Y^2} \sim \text{Gamma}(0.001, 0.001)$$

Tables 4.1, 4.2, 4.3 and 4.4 below summarize estimation results from 100 simulations of the samples with 10, 30, 100, and 1000 data points. The history of iteration process for  $\beta_1$  and  $\beta_2$  coefficients in the test run of the research model with a generated sample of 1000 data points is exhibited on Figure 4.1 and Figure 4.2 below, while more detailed WinBUGS output summary for the test run is provided in Appendix F. The graphs below suggest that the estimates obtained in the test run do not converge and depend on the initial values, since all three lines representing three different chains for  $\beta_1$  and  $\beta_2$  coefficients are almost parallel.

**Figure 4.1. History of WinBUGS iterations for  $\beta_1$**



**Figure 4.2. History of WinBUGS iterations for  $\beta_2$**



Problems with the estimates' convergence in the test run of the model suggest potential issues with the estimates' divergence in all runs of the research model. Therefore, further research in the field can focus on improving convergence by considering the following options:

- a) Developing a better parameterization of the model to improve orthogonality of the joint posterior distribution.
- b) Standardization of the covariates to have mean zero and standard deviation one.
- c) Using ordered over-relaxation of the parameters distributions to suppress random walk autocorrelation between two subsequent elements in a Markov chain. According to Neal (1995), this method generates a  $K$  number of candidates for a subsequent element of the chain, ranks the previous element and the candidates for the next element in non-decreasing order, and selects the next element with a rank that equals the difference between the maximum rank in the overall list and the rank of the previous element of the chain.

## 4.2. Comparison of OLS Estimators and Hierarchical Bayes Estimators

Results of the simulation study for the sample sizes of 10, 30, 100, and 1,000 observations are summarized below in Tables 4.1, 4.2, 4.3, and 4.4 respectively. The estimated biases of the Gibbs Sampler estimator of  $\beta_0$  and  $\beta_1$  tend to be lower in absolute values than the biases of the corresponding OLS estimators for all sample sizes. The estimated MSE of the Gibbs Sampler estimators of  $\beta_0$  and  $\beta_1$  are lower in average than the MSE of the OLS estimators in all sample sizes. In contrast, the biases of Gibbs Sampler estimators  $\beta_2$  and  $\sigma_Y^2$  are higher in absolute values than the corresponding OLS estimators. The estimated MSE of the Gibbs Sampler for  $\beta_1$  and  $\sigma_Y^2$  estimators are in average higher than the MSE of the OLS estimators. However, the formal tests detected significant difference in MSE only for  $\beta_1$  estimators for the

small samples of size 10 and 30 observations. The other differences in MSE are statistically insignificant. All details on testing significance of difference in MSE are given in Appendix G.

To recap, my simulation study showed that the Gibbs Sampler estimator for a time trend coefficient  $\beta_1$  tends to be less biased than the OLS estimator in samples of all sizes, while the Gibbs Sampler estimator is more precise for a time trend only in small samples of 10 and 30 observations. In the bigger samples, Gibbs Sampler and OLS estimators are similar in their precision.

**Table 4.1. Estimation results from 100 simulations with a sample size 10 data points**

Indicator	$\beta_0$ estimate					$\beta_1$ estimate				
	Gibbs Sampler				OLS	Gibbs Sampler				OLS
	Initials 1	Initials 2	Initials 3	Average		Initials 1	Initials 2	Initials 3	Average	
<b>True value</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
Estimate	0.435	0.424	0.560	0.473	-1.019	0.933	0.934	0.990	0.952	1.633
Bias	0.435	0.424	0.560	0.473	-1.019	-0.067	-0.066	-0.010	-0.048	0.633
MSE estimate	0.611	0.550	0.680	0.614	4.030	0.365	0.299	0.388	0.351	1.520
MSE SE	0.098	0.053	0.120	0.090	2.447	0.150	0.199	0.171	0.173	0.076
95% MSE lower limit	0.419	0.446	0.440	0.435	-0.766	0.072	-0.090	0.053	0.012	1.371
95% MSE upper limit	0.804	0.653	0.920	0.792	8.825	0.658	0.689	0.723	0.690	1.668

Indicator	$\beta_2$ estimate					$\sigma_v$ estimate				
	Gibbs Sampler				OLS	Gibbs Sampler				OLS
	Initials 1	Initials 2	Initials 3	Average		Initials 1	Initials 2	Initials 3	Average	
<b>True value</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
Estimate	0.020	0.011	-0.077	-0.015	0.168	1.071	1.130	1.097	1.099	0.906
Bias	-0.480	-0.489	-0.577	-0.515	-0.332	0.071	0.130	0.097	0.099	-0.094
MSE estimate	0.605	0.664	0.732	0.667	0.494	0.474	0.475	0.433	0.461	0.337
MSE SE	0.092	0.168	0.175	0.145	1.107	0.040	0.022	0.126	0.063	1.265
95% MSE lower limit	0.425	0.335	0.389	0.383	-1.674	0.395	0.431	0.186	0.337	-2.141
95% MSE upper limit	0.784	0.993	1.074	0.950	2.663	0.553	0.519	0.680	0.584	2.816

**Table 4.2. Estimation results from 100 simulations with a sample size 30 data points**

Indicator	$\beta_0$ estimate					$\beta_1$ estimate				
	Gibbs Sampler				OLS	Gibbs Sampler				OLS
	Initials 1	Initials 2	Initials 3	Average		Initials 1	Initials 2	Initials 3	Average	
<b>True value</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
Estimate	0.190	0.188	0.146	0.175	-0.630	0.987	0.990	1.010	0.996	1.352
Bias	0.190	0.188	0.146	0.175	-0.630	-0.013	-0.010	0.010	-0.004	0.352
MSE estimate	0.509	0.521	0.545	0.525	2.217	0.340	0.341	0.338	0.340	0.820
MSE SE	0.059	0.064	0.083	0.069	1.321	0.111	0.117	0.125	0.118	0.083
95% MSE lower limit	0.394	0.396	0.382	0.391	-0.371	0.122	0.111	0.093	0.109	0.657
95% MSE upper limit	0.625	0.646	0.707	0.659	4.805	0.558	0.570	0.583	0.570	0.983

Indicator	$\beta_2$ estimate					$\sigma_v$ estimate				
	Gibbs Sampler				OLS	Gibbs Sampler				OLS
	Initials 1	Initials 2	Initials 3	Average		Initials 1	Initials 2	Initials 3	Average	
<b>True value</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
Estimate	0.004	0.001	-0.017	-0.004	0.323	1.044	1.041	1.026	1.037	0.977
Bias	-0.496	-0.499	-0.517	-0.504	-0.177	0.044	0.041	0.026	0.037	-0.023
MSE estimate	0.587	0.594	0.620	0.600	0.281	0.366	0.374	0.347	0.362	0.293
MSE SE	0.137	0.137	0.158	0.144	0.625	0.085	0.084	0.116	0.095	0.613
95% MSE lower limit	0.318	0.325	0.310	0.318	-0.943	0.200	0.210	0.119	0.176	-0.907
95% MSE upper limit	0.856	0.862	0.930	0.883	1.506	0.532	0.538	0.575	0.548	1.494

**Table 4.3. Estimation results from 100 simulations with a sample size 100 data points**

Indicator	$\beta_0$ estimate					$\beta_1$ estimate				
	Gibbs Sampler				OLS	Gibbs Sampler				OLS
	Initials 1	Initials 2	Initials 3	Average		Initials 1	Initials 2	Initials 3	Average	
<b>True value</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
Estimate	-0.018	0.000	0.000	-0.006	-0.021	1.025	1.002	1.002	1.010	1.012
Bias	-0.018	0.000	0.000	-0.006	-0.021	0.025	0.002	0.002	0.010	0.012
MSE estimate	0.581	0.567	0.567	0.571	0.601	0.332	0.314	0.314	0.320	0.322
MSE SE	0.126	0.128	0.128	0.127	0.246	0.124	0.126	0.126	0.126	0.034
95% MSE lower limit	0.333	0.316	0.316	0.322	0.119	0.090	0.066	0.066	0.074	0.255
95% MSE upper limit	0.828	0.817	0.817	0.821	1.083	0.574	0.561	0.561	0.566	0.389

Indicator	$\beta_2$ estimate					$\sigma_v$ estimate				
	Gibbs Sampler				OLS	Gibbs Sampler				OLS
	Initials 1	Initials 2	Initials 3	Average		Initials 1	Initials 2	Initials 3	Average	
<b>True value</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
Estimate	-0.025	-0.002	-0.002	-0.010	0.494	0.999	0.999	0.999	0.999	0.998
Bias	-0.525	-0.502	-0.502	-0.510	-0.006	-0.001	-0.001	-0.001	-0.001	-0.002
MSE estimate	0.595	0.565	0.565	0.575	0.190	0.313	0.313	0.313	0.313	0.311
MSE SE	0.140	0.126	0.126	0.131	0.167	0.143	0.128	0.128	0.133	0.045
95% MSE lower limit	0.320	0.318	0.318	0.319	-0.136	0.033	0.063	0.063	0.053	0.223
95% MSE upper limit	0.870	0.812	0.812	0.831	0.517	0.593	0.563	0.563	0.573	0.400

**Table 4.4. Estimation results from 100 simulations with a sample size 1000 data points**

Indicator	$\beta_0$ estimate					$\beta_1$ estimate				
	Gibbs Sampler				OLS	Gibbs Sampler				OLS
	Initials 1	Initials 2	Initials 3	Average		Initials 1	Initials 2	Initials 3	Average	
<b>True value</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
Estimate	-0.018	0.000	0.000	-0.006	-0.021	1.025	1.002	1.002	1.010	1.012
Bias	-0.018	0.000	0.000	-0.006	-0.021	0.025	0.002	0.002	0.010	0.012
MSE estimate	0.581	0.567	0.567	0.571	0.601	0.332	0.314	0.314	0.320	0.322
MSE SE	0.126	0.128	0.128	0.127	0.246	0.124	0.126	0.126	0.126	0.034
95% MSE lower limit	0.333	0.316	0.316	0.322	0.119	0.090	0.066	0.066	0.074	0.255
95% MSE upper limit	0.828	0.817	0.817	0.821	1.083	0.574	0.561	0.561	0.566	0.389

Indicator	$\beta_2$ estimate					$\sigma_v$ estimate				
	Gibbs Sampler				OLS	Gibbs Sampler				OLS
	Initials 1	Initials 2	Initials 3	Average		Initials 1	Initials 2	Initials 3	Average	
<b>True value</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
Estimate	-0.025	-0.002	-0.002	-0.010	0.494	0.999	0.999	0.999	0.999	0.998
Bias	-0.525	-0.502	-0.502	-0.510	-0.006	-0.001	-0.001	-0.001	-0.001	-0.002
MSE estimate	0.595	0.565	0.565	0.575	0.190	0.313	0.313	0.313	0.313	0.311
MSE SE	0.140	0.126	0.126	0.131	0.167	0.143	0.128	0.128	0.133	0.045
95% MSE lower limit	0.320	0.318	0.318	0.319	-0.136	0.033	0.063	0.063	0.053	0.223
95% MSE upper limit	0.870	0.812	0.812	0.831	0.517	0.593	0.563	0.563	0.573	0.400

## Chapter 5. Conclusions

I created a model including a time trend and a first order autoregressive term, and applied it to testing for the existence of the global warming. My hierarchical Bayesian research model with diffuse conjugate prior distributions took into account natural high autocorrelation between a monthly temperature observation and its first lag by incorporating an AR(1) term and placing a prior distribution on it. The parameters of the model were estimated by applying OLS and Gibbs Sampler with 30,000 iterations and three chains with the different starting values. The Gibbs Sampler estimate for the global temperature increase was between 0.0203°C and 0.0284°C per decade with 95% credibility. The OLS estimate for the temperature increase was between 0.0124°C and 0.0406°C per decade at 95% confidence, while the difference between Gibbs Sampler estimate and OLS estimate was insignificant. Moreover, since all percentiles of the posterior distribution of the Gibbs Sampler estimate for a time trend were above 0, there is strong evidence of global warming.

Second, in my simulation study I randomly generated 100 samples of 10, 30, 100, and 1000 observations from a normal distribution with a mean having a linear time trend and an AR(1) term, and the model parameters fixed at the following values:  $\beta_0 = 0$ ,  $\beta_1 = 1$ ,  $\beta_2 = 0.5$ ,  $\sigma_Y = 1$ . Further, these parameters were estimated by OLS and by Gibbs Sampler. The Gibbs Sampler estimators with three different initial values failed to converge and were considered separately. In 100 simulations, the Gibbs Sampler estimators for an intercept and for the time trend were less biased than corresponding OLS estimators, while the reverse was true for the autoregressive parameter and error standard deviation. The difference in precision of the estimators found by two approaches was insignificant except for the estimator of the time trend. The Gibbs Sampler estimators for MSE of the time trend coefficient were significantly smaller than the OLS estimators of MSE only in the small samples of 10 and 30 observations.

Third, although the test run in simulation study detected lack of convergence in the Gibbs Sampler estimates for the time trend and AR(1) term coefficients, my research based on the real temperature data did not show significant issues with convergence. The difference in estimates' convergence can occur for several reasons. The distribution of the real data can differ significantly from the assumed distribution with the mean following AR(1) model with a time trend. Therefore, the same mix of the prior distributions and the research model can result in

different convergence outcomes in the simulation study and a study based on the real data. On the other hand, a problem with convergence can be related to the parameters of the iterations process, like number of iterations, thinning rate, and different initial values. Further research can focus on improving convergence by applying standardization of the covariates, developing a better parameterization of the model, or by using ordered over-relaxation of the parameter distributions. The over-relaxation method selecting the next element of a Markov chain from a fixed number of the generated candidates in a specific order should improve performance of the Gibbs Sampler by suppressing random walk autocorrelation among the elements of the chain.

Fourth, the simulation study validated the results of the hierarchical Bayesian model that used the real data of 1539 global-mean monthly temperature observations. On average, the Gibbs Sampler estimators for the intercept and for the time trend tend to be less biased than the corresponding OLS estimators, while the Gibbs Sampler estimators for the AR(1) term and the error variance are to be more biased. The difference in precision of all Gibbs Sampler estimators and OLS estimators tends to be insignificant.



## References

- Angel J., Variation and trends in tropospheric and stratospheric Global temperatures, 1958-87, *Journal of climate*, Vol.1, 1988
- Brooks S., Gelman A., Some issues in monitoring convergence of iterative simulations, working paper, <http://www.mrc-bsu.cam.ac.uk/bugs/>, 2001
- Cowles M., and Carlin B., Markov Chain Monte Carlo Convergence Diagnostics: A comparative Review, *Journal of the American Statistical Association*, 1996, Vol.91, No.434, pp.883-904
- Gelfand A., Smith A., Sampling-Based Approaches to Calculating Marginal Densities, *Journal of the American Statistical Association*, 1990, Vol. 85, No.410, pp.398-409
- Gelman A., and Carlin J., Stern H., Rubin D., *Bayesian Data Analysis*, 2<sup>nd</sup> edition, Chapman & Hall/CRC, 2004
- Gelman A., and Rubin D., Inference From Iterative Simulation Using Multiple Sequences, *Statistical Science*, 7, pp. 457-511
- Geman, S., and Geman, D. (1984), Stochastic Relaxation, Gibbs Distributions and Bayesian Restoration of Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721-741.
- Greene W., *Econometric Analysis*, Pearson Education Inc., 2003
- Jones P., Recent warming in Global temperature series, *Geophysical research letters*, Vol. 21, No. 12, p.1149-1152, 1994
- Oort A., and Liu H., Upper-air temperature trends over the Globe, 1958-1989, *Journal of climate*, Vol.6, 1992
- Klugman S., *Bayesian Statistics in Actuarial Science with Emphasis on Credibility*, Kluwer Academic Publishers, 1992.
- Meyers G., Empirical Bayesian credibility for workers' compensation classification ratemaking, *PCAS LXXI*, 1984, p. 96.
- Neal R., Suppressing random walks in Markov Chain Monte Carlo using ordered overrelaxation, Technical Report No.9508, Department of Statistics, University of Toronto, 1995

Ritter, C., and Tanner, M., Facilitating the Gibbs Sampler: The Gibbs Stopper and Griddy-Gibbs Sampler, *Journal of the American Statistical Association*, 1992, Vol.87, pp.861-868

Robert, C., *The Bayesian Choice. A Decision-Theoretic Motivation*, Springer-Verlag, 1994

Santer B., Taylor K., Wigley T., Johns T., Jones P., Karoly D., Mitchell J., Oort A., Penner J., Ramaswamy V., Schwarzkopf, Stouffer R., Tett S., A search for human influences on the thermal structure of the atmosphere, *Nature*, Vol. 382, 4 July, 1996

Scollnik D., An introduction to Markov Chain Monte Carlo Methods and their actuarial applications, *North American Actuarial Journal*, Society of Actuaries (2001): 96-125

Tierney L., Markov Chains for Exploring Posterior Distributions, *The Annals of Statistics*, 1994, Vol. 22, No.4, 1704-1762

United Nations Framework Convention on Climates Change , Article, retrieved on 01/15/2007.

## Appendix A. R output summary on OLS estimation of the research model

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	5.283e+00	2.743e-01	19.26	<2e-16 ***
t	1.981e-04	1.457e-05	13.59	<2e-16 ***
Y[t-1]	6.120e-01	2.013e-02	30.41	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.182 on 1535 degrees of freedom  
Multiple R-squared: 0.6809, Adjusted R-squared: 0.6805  
F-statistic: 1638 on 2 and 1535 DF, p-value: < 2.2e-16

## Appendix B. A code for estimation of the research model in WinBUGS

```

# 1. Research model with conjugate priors, a linear trend and AR(1) term
model
const
T=1539;

var
Y[T], t;

{
for(t in 2:T)
  {
  Y[t] ~ dnorm(theta.Y[t], tau.y)
  theta.Y[t] <- beta0+ beta1*t + beta2*Y[t-1]
  }
# Prior
beta0 ~ dnorm(beta0.c, beta0.tau) #non-informative diffuse normal distn
beta1 ~ dnorm(beta1.c, beta1.tau) #non-informative diffuse normal distn
beta2 ~ dnorm(beta2.c, beta2.tau) #non-informative diffuse normal distn
tau.y ~ dgamma(0.001,0.001)
sigma.y <-pow((1/tau.y), 0.5)

beta0.c ~ dnorm(0, 1.0E-6) #non-informative diffuse normal distn
beta0.tau ~ dgamma(0.001, 0.001)
sigma.beta0 <- pow((1/beta0.tau), 0.5)

beta1.c ~ dnorm(0, 1.0E-6) #non-informative diffuse normal distn
beta1.tau ~ dgamma(0.001, 0.001)
sigma.beta1<-pow((1/beta1.tau), 0.5)

beta2.c ~ dnorm(0, 1.0E-6) #non-informative diffuse normal distn
beta2.tau ~ dgamma(0.001, 0.001)
sigma.beta2<-pow((1/beta2.tau), 0.5)
}

# 3. Initial values for 3 chains with slightly different original values
list(beta0=0, beta1=0, beta2=0, beta0.c=0, beta1.c=0, beta2.c=0,
beta0.tau=1, beta1.tau=1, beta0.tau=1, beta2.tau=1, tau.y=1)
list(beta0=0, beta1=0, beta2=0, beta0.c=0, beta1.c=0, beta2.c=0,
beta0.tau=0.01, beta1.tau=0.01, beta2.tau=0.01, tau.y=0.01)
list(beta0=0, beta1=0, beta2=0, beta0.c=0, beta1.c=0, beta2.c=0,
beta0.tau=0.0001, beta1.tau=0.0001, beta2.tau=0.0001, tau.y=0.0001)

# 2. Data

list(T = 1539,
Y = c(14.24, 14.1,13.89,13.83,13.87,13.43,13.81,14.05,14.1,13.63,13.71,13.98,
13.48,13.95,13.86,14.09,14.21,13.14,13.95,13.93,13.86,13.83,13.84,14.27,
14.78,14.42,14.26,13.67,13.88,13.35,13.44,13.83,14.26,13.88,14.09,13.99,
13.84,13.45,13.96,14.15,13.97,14.53,14.18,13.98,13.7,13.85,13.59,14.11,
13.81,14.09,13.74,13.25,13.25,13.53,13.43,13.87,13.77,13.58,13.38,13.13,

```

13.4,13.49,13.43,13.51,13.63,13.43,13.88,14.07,13.89,14.28,13.74,14.35,  
13.53,13.55,13.58,13.81,13.85,13.84,13.88,13.8,13.76,13.35,14.02,13.8,  
13.42,13.18,13.74,13.6,13.54,13.32,13.75,13.62,13.51,13.73,13.22,13.73,  
13.59,13.44,13.36,13.95,13.59,13.75,13.64,13.68,13.67,13.96,14.31,14.1,  
14.21,14.42,14.26,14.35,14.33,13.97,13.64,13.7,13.82,14.07,13.78,14.1,  
13.85,13.65,13.88,13.95,13.71,13.96,13.54,13.8,13.98,13.91,13.42,13.74,  
13.24,12.94,13.53,13.36,13.76,13.52,13.18,13.35,13.52,13.36,13.67,13.88,  
13.48,13.92,13.55,13.27,13.5,13.6,13.68,13.86,13.58,13.6,13.59,13.47,  
12.46,13.19,13.54,13.7,13.53,13.54,13.94,14,13.81,14.24,13.69,13.51,  
13.37,13.37,13.83,13.53,13.5,13.7,13.72,13.69,13.47,13.95,13.95,13.99,  
13.47,13.1,13.55,13.5,13.73,13.76,13.37,13.79,13.88,13.97,14,13.97,  
14.02,13.78,13.55,13.57,13.52,13.63,13.69,13.51,13.8,14.15,13.61,13.87,  
13.87,13.72,13.65,14.08,13.94,13.73,14.14,13.74,13.79,13.58,13.99,13.93,  
14.26,13.68,13.41,13.68,13.48,13.81,13.93,14.12,14.05,13.57,13.7,13.88,  
13.6,13.59,13.84,13.67,13.5,13.75,13.66,13.72,13.99,13.72,14.11,13.83,  
13.55,13.97,14.01,13.92,14.04,14.11,13.79,13.86,13.94,14.19,13.9,13.95,  
13.89,13.95,13.96,14.08,13.97,14.12,13.72,13.94,14.06,13.97,13.93,13.85,  
13.71,13.97,13.94,13.81,13.71,13.76,13.71,13.55,13.6,13.63,13.56,13.45,  
13.72,13.88,13.96,13.52,13.62,13.41,13.77,13.63,13.6,13.51,13.55,13.58,  
13.45,13.37,13.7,13.69,13.44,13.54,13.43,13.66,13.59,13.57,13.79,13.73,  
13.59,13.14,13.84,13.82,13.9,13.85,13.71,13.8,13.77,13.67,14.09,13.83,  
13.77,13.75,13.6,14.12,13.83,13.95,13.91,13.93,13.74,13.97,13.68,13.95,  
13.64,13.55,13.68,13.61,13.55,13.55,13.72,13.57,13.71,13.76,13.5,13.39,  
13.7,13.85,13.55,13.69,13.83,13.7,13.81,13.73,13.88,13.6,13.4,13.68,  
13.32,13.59,13.59,13.44,13.6,13.83,13.82,13.91,13.82,13.92,13.96,13.55,  
13.76,13.86,13.67,13.94,13.91,13.77,13.93,13.92,14.04,13.77,13.6,13.42,  
13.49,13.5,13.54,13.64,13.79,13.73,13.92,13.82,13.86,13.86,14.03,13.83,  
13.89,13.89,13.55,13.81,13.81,13.78,13.59,13.43,13.57,13.53,13.51,13.73,  
13.67,13.53,13.59,13.83,13.4,13.45,13.73,13.79,13.83,13.8,13.92,14.07,  
14.2,14.03,14.1,13.82,14.02,13.95,13.85,14,13.97,14.15,13.81,13.87,  
13.94,14.15,14.07,14.09,13.99,14.02,14.18,13.85,14.19,14.03,14.17,13.94,  
14.04,13.98,13.68,13.82,13.8,13.81,13.83,13.99,13.92,13.85,13.62,13.26,  
13.49,13.27,13.33,13.54,13.33,13.6,14.16,13.68,13.7,13.55,13.66,13.13,  
13.47,13.46,13.54,13.59,13.4,13.73,13.56,13.59,13.89,14.05,13.82,13.77,  
13.78,13.85,13.97,13.98,14,13.99,14.09,13.99,13.89,14.09,13.67,13.76,  
14.01,13.8,13.84,13.95,13.73,13.78,14.02,13.94,13.89,13.64,13.76,13.6,  
14.15,13.89,13.98,14.02,13.94,14.1,14.13,13.69,13.94,13.96,13.79,13.89,  
13.74,13.7,13.82,14.14,13.78,13.86,14.12,13.88,13.99,13.87,13.99,13.96,  
13.84,13.82,13.87,13.65,13.87,13.81,13.63,13.73,13.75,14,14.04,14.03,  
13.77,13.66,13.98,13.67,13.86,13.96,13.98,13.97,14.07,13.92,13.99,13.76,  
13.67,13.7,13.82,13.95,13.78,13.73,13.73,13.98,13.72,13.76,14.05,14.25,  
14.29,14.34,14.2,13.99,13.81,13.9,13.93,14.1,14.09,13.93,14.11,13.79,  
13.91,13.89,13.75,13.77,13.86,13.87,14.08,13.82,14.02,14.26,14.24,13.87,  
14.16,14.12,14.02,14.01,13.68,13.56,14.01,14.05,14.1,14.02,14.11,14.06,  
13.78,13.65,13.74,13.68,13.72,13.72,13.75,13.74,13.85,14.1,14.07,13.5,  
13.84,14,14.08,13.84,14.02,13.89,13.98,13.96,13.89,13.8,14.23,14.04,  
13.98,13.8,14.06,13.88,13.93,14.08,14.2,14.19,13.92,14.11,13.93,14.21,  
14.52,13.9,13.97,14.23,14,13.71,13.95,13.91,14.15,14.2,14.01,13.95,  
13.8,13.93,13.93,13.93,13.95,13.88,14.04,13.87,14,14.02,13.73,13.59,  
13.99,14.07,13.82,13.77,14.18,13.98,14.17,14.2,14.02,14.06,14.15,14.19,  
13.8,14.34,14.21,13.67,13.86,13.94,13.92,13.84,13.69,14.03,13.81,13.92,  
13.86,13.71,13.92,14.1,13.8,13.96,14.34,14.11,14.07,14.09,14.09,14.18,  
14.02,14.15,13.93,13.97,14.12,14.11,14.05,14.13,14.3,14.28,14.35,14.05,  
14.1,14.11,14.3,14.37,14.18,13.98,14.11,13.98,14.14,14.41,14.18,13.97,  
14.08,13.97,13.75,13.93,14.11,14.03,13.98,13.81,13.92,13.94,13.88,14.4,  
13.87,14.11,14.15,14.1,14.14,14.23,14.32,14.13,14.21,14.08,14.09,14.29,

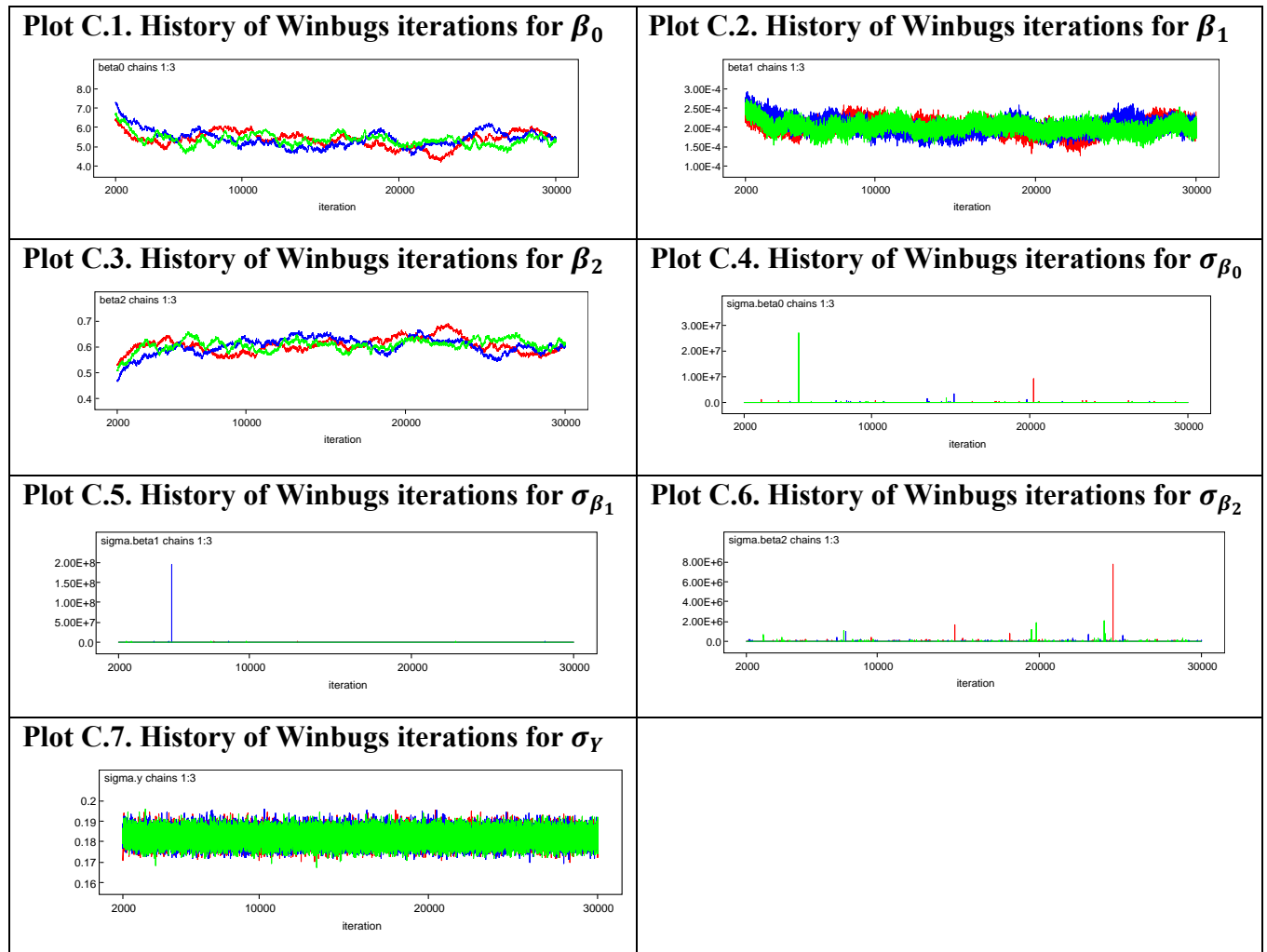
14.15,14.23,14,14.07,14.12,14.2,14.37,14.13,13.84,14.16,14.02,14.12,  
14.29,14.02,14.02,14.05,14.1,14.12,14.13,14.1,14.11,14.03,14.19,14.13,  
13.85,14.13,13.91,14.07,13.97,13.93,14.19,13.8,14.13,14.36,14.14,14.29,  
14.4,14.19,14.15,13.99,13.95,13.92,14.1,14.12,14.26,14.27,14.04,13.91,  
14.07,13.97,14.03,14.18,13.77,13.79,13.87,14.3,14.11,14.07,13.98,13.74,  
14.1,14.06,13.94,14.13,14.12,13.92,14.13,13.95,14.07,13.88,14,13.78,  
14,14.14,14.25,14.2,14.07,14.07,14.1,14.1,14.06,14.29,14.27,13.94,  
14.27,13.94,13.79,13.94,14.06,14.14,13.92,13.88,14,14.02,13.86,13.89,  
14.11,13.73,13.92,13.75,13.9,13.83,14,13.89,13.99,14.04,13.94,13.9,  
13.78,13.73,13.95,13.81,13.82,13.93,13.91,13.72,13.88,13.85,13.67,13.89,  
13.69,13.59,13.9,13.99,14.12,13.93,13.94,14.17,14.07,14.12,14.08,14.22,  
14.15,14.16,13.91,14,13.93,13.97,14.12,14.15,14.16,14.04,13.84,14,  
14.13,14.11,14.2,14.16,14.03,14.08,14.03,14.16,14.13,14.1,14.07,14.22,  
13.76,13.96,13.93,13.9,13.85,13.85,13.78,13.85,13.84,14.04,14.22,13.93,  
14.26,13.95,13.68,13.87,13.87,13.86,13.91,14.15,13.91,14.05,13.79,13.75,  
13.81,13.74,13.71,13.78,13.77,13.95,13.93,13.7,13.93,13.82,13.84,13.86,  
13.89,13.99,14,14.14,14.11,14.17,13.94,14.13,14.14,14.02,14.19,14.21,  
14.42,14.25,14.2,14.13,14.11,13.86,14.14,13.92,13.94,14.11,14.11,13.95,  
14.07,14.08,14.26,14.13,13.99,14.1,14.08,14,13.87,13.95,13.92,14.11,  
14.07,14.28,13.56,13.8,13.87,13.96,13.9,14.1,14.06,14.12,13.9,14.2,  
14.1,14.22,14.05,14.1,14.24,14.11,14.01,14.05,14.16,14.17,14.09,13.88,  
14.08,14.22,14.11,14.12,13.88,14.1,13.89,13.93,14.06,14.05,14.11,14.02,  
13.98,14.25,13.79,13.8,13.87,13.99,14.08,14.17,14.27,14.05,14.04,13.98,  
13.93,13.84,13.7,13.62,13.61,13.87,13.92,13.71,13.61,13.67,13.8,13.67,  
13.97,13.81,13.89,13.71,13.89,13.88,13.73,13.92,13.8,13.92,13.88,13.81,  
13.76,13.92,14.14,13.87,13.9,13.95,14.12,13.89,13.93,13.75,13.94,13.92,  
13.93,13.72,14.01,13.93,14.11,13.85,14.05,14,14.02,14.15,14.01,13.98,  
13.8,13.84,14.28,13.91,13.87,13.93,13.9,13.93,13.75,14.06,13.92,13.77,  
13.81,13.72,13.85,14.12,14.1,14.07,13.93,13.93,14.02,14.07,14.08,14.33,  
14.08,14.28,14.08,14.05,13.93,14.08,14.06,13.89,14.15,14.05,14.02,13.87,  
13.96,13.85,13.77,13.92,13.89,13.77,13.91,14.05,14,13.95,13.86,13.87,  
13.65,13.66,13.92,13.94,13.95,14.05,14.03,14.18,14.05,13.9,13.91,14.11,  
14.23,14.31,14.32,14.34,14.28,14.23,14.16,14.03,14.07,14.15,14.06,13.95,  
13.93,13.74,13.96,13.88,13.98,14.01,14,14.11,13.87,13.93,13.92,13.95,  
14.06,14.07,14.14,14.01,14.26,14.01,14,13.74,13.94,13.91,13.87,13.77,  
13.92,13.88,13.64,13.74,13.75,13.8,13.8,13.75,13.81,13.6,13.79,13.95,  
14.08,14.2,14.24,14.26,14.31,14.28,14.24,14.27,13.87,13.93,14.17,14.07,  
14.06,14.14,14.21,14.22,14.16,13.96,14.15,13.8,14.09,13.98,14.06,14.03,  
14.13,13.85,14.19,14.19,13.95,14.15,13.99,14.08,14.19,14.21,14.25,14.46,  
14.29,14.35,14.36,14.38,14.43,14.18,14.27,14.23,14.28,14.14,14.26,14.15,  
14.57,14.53,14.61,14.46,14.27,14.35,14.51,14.54,14.24,14.08,14.19,14.44,  
14.08,14.2,13.94,14.03,14.26,13.99,14.16,14,14,13.98,14.11,14.34,  
14.44,14.38,14.38,14.33,14.45,14.27,14.2,14.47,14.49,14.23,14.29,14.16,  
14.2,14.06,14.3,14.08,14.45,13.9,14.16,14.22,14.21,14.16,14.05,13.95,  
14.24,13.98,14.21,14.14,14.11,14.18,14,14.32,14.18,14.06,14,14.09,  
14.28,14.46,14.31,14.33,14.2,14.12,14.11,14.18,14.03,14.05,14.04,14.12,  
14.35,14.49,14.24,14.39,14.24,14.47,14.5,14.11,14.29,14.32,14.28,14.48,  
14.5,14.4,14.53,14.4,14.51,14.5,14.34,14.38,14.42,14.41,14.02,14.26,  
14.08,14.34,14.33,14.28,14.07,14.07,14.36,14.41,14.43,14.31,14.12,14.29,  
14.45,14.35,14.81,14.58,14.45,14.41,14.66,14.38,14.18,14.51,14.55,14.47,  
14.44,14.52,14.39,14.63,14.34,14.62,14.59,14.49,14.5,14.26,14.24,14.25,  
14.45,14.34,14.4,14.19,14.14,14.23,13.99,14.1,13.9,14,13.92,14.13,  
14.3,14.31,14.33,14.24,14.25,14.17,14.25,14.16,14.03,14.16,14.04,14.03,  
14.35,14.04,14.23,14.42,14.24,14.56,14.32,14.21,14.42,14.35,14.33,14.31,  
14.47,14.81,14.47,14.46,14.14,14.4,14.6,14.5,14.35,14.6,14.44,14.33,  
14.37,14.5,14.38,14.32,14.27,14.21,14.5,14.64,14.34,14.27,14.43,14.41,

14.39,14.38,14.5,14.33,14.32,14.54,14.19,14.38,14.33,14.43,14.57,14.53,  
14.58,15.02,14.7,14.71,14.79,14.84,14.9,14.76,14.48,14.58,14.54,14.7,  
14.55,14.75,14.35,14.32,14.35,14.54,14.51,14.44,14.43,14.43,14.48,14.36,  
14.14,14.61,14.47,14.63,14.4,14.54,14.45,14.49,14.39,14.22,14.36,14.31,  
14.51,14.54,14.6,14.4,14.6,14.57,14.56,14.58,14.59,14.5,14.75,14.63,  
14.84,14.82,14.99,14.77,14.73,14.51,14.73,14.57,14.62,14.57,14.62,14.45,  
14.79,14.67,14.59,14.59,14.74,14.52,14.57,14.76,14.71,14.76,14.59,14.76,  
14.61,14.84,14.77,14.67,14.43,14.52,14.21,14.55,14.51,14.71,14.75,14.64,  
14.87,14.79,14.85,14.78,14.65,14.72,14.67,14.65,14.85,14.86,14.76,14.72,  
14.56,14.76,14.65,14.6,14.4,14.64,14.57,14.71,14.65,14.74,14.74,14.82,  
15.08,14.81,14.73,14.78,14.74,14.54,14.65,14.78,14.72,14.75,14.67,14.61,  
14.34,14.33,14.68))

## Appendix C. WinBUGS summary on Gibbs Sampler estimation for the research model

The first run of the model in WinBUGS showed that it takes near 2000 iterations of the model parameters to converge. Therefore, 2000 iterations out of 30000 were specified to be burn in. The final summary is based on the WinBUGS output excluding burn-in iterations and having a thinning rate of 3 iterations.

Node statistics	mean	sd	MC error	2.5%	median	97.5%	start	sample
node								
beta0	5.383	0.378	0.01671	4.712	5.361	6.216	2000	84003
beta1	2.017E-4	1.734E-5	6.109E-7	1.69E-4	2.012E-4	2.374E-4	2000	84003
beta2	0.6047	0.02773	0.001226	0.5435	0.6064	0.6539	2000	84003
sigma.beta0	1194.0	99300.0	345.1	0.04328	8.339	3456.0	2000	84003
sigma.beta1	3291.0	667600.0	2312.0	0.04483	9.163	3657.0	2000	84003
sigma.beta2	853.4	30900.0	107.5	0.0437	8.994	3521.0	2000	84003
sigma.y	0.1822	0.003299	1.424E-5	0.1758	0.1821	0.1888	2000	84003





## Appendix D. R code for simulation study on Gibbs Sampler and OLS estimation of the research model

```
# Assumed true parameters

beta0<-0
beta1<-1
beta2<-0.5
sigma<-1
N<-1010           #1700
nburn<-10        #A number of the first simulated obs to be burnt=200

truevalue<-c(beta0, beta1, beta2, sigma)
library(boot)

# Bayesian simulation using WinBugs with AR1 gamma specification

B<-200
BUGSest<-matrix(0,nrow=B,ncol=4) #ncol=4 parameters of the distribution
out<-matrix(0,nrow=B,ncol=3)
time<-matrix(1,nrow=N,ncol=1)
X<-matrix(0,nrow=(N-nburn),ncol=2)
Y<-matrix(0,nrow=(N-nburn),ncol=1)
Y1<-matrix(0,nrow=(N-nburn),ncol=1)

# Data simulation
for (j in 1:B){
  y<-matrix(0,nrow=N,ncol=1)
  ylag<-matrix(0,nrow=N,ncol=1)
  mu<-matrix(0,nrow=N,ncol=1)
  t<-1
  mu[t]<-beta0+beta1*t+beta2*0
  y[t]<-rnorm(1,mu[t],sigma)

  for (t in 2:N){
    mu[t]<-beta0+beta1*t+beta2*y[t-1]
    y[t]<-rnorm(1,mu[t],sigma)
    ylag[t]<-y[t-1]
    time[t]<-t
  }

  Y<-y[(nburn+1):N]
  #Y1<-ylag[(nburn+1):N]

# BUGS regression estimation

T<-N-nburn

# Data loading
data<-list("T","Y")

# Declaration of the model parameters to be iterated
# Loading initial values for 3 MCMC chains
```

```

parameters<-c("beta0", "beta1", "beta2", "sigma.y", "sigma.beta0",
"sigma.beta1", "sigma.beta2")

# Invoking an auxiliary sub package "arm" of the R2WinBUGS package
inits1<-list(beta0=0, beta1=0, beta2=0, tau.y=1, beta0.tau=1, beta1.tau=1, beta2.tau=1)
inits2<-list(beta0=0, beta1=0, beta2=0, tau.y=0.01, beta0.tau=0.01, beta1.tau=0.01, beta2.tau=0.01)
inits3<-list(beta0=0, beta1=0, beta2=0, tau.y=0.0001, beta0.tau=0.0001, beta1.tau=0.0001, beta2.tau=0.0001)
inits<-list(inits1, inits2, inits3)

# Invoking an auxiliary sub package "arm" of the R2WinBUGS package
library("arm")
AR1bugs.sim<- bugs(data, inits, parameters, "D:/KSU_Research/Final/AR1GammaModel.txt",
  n.chains=3, n.iter=5000,
  bugs.directory="c:/Program Files/WinBUGS14/", working.directory="D:/KSU_Research/Final/Results/",
  clearWD=FALSE, debug=FALSE)

# Applying bugs function entering inputs and running WinBUGS from R
attach.bugs(AR1bugs.sim)

# Saving key results
BUGSest[j,1]<-mean(beta0)
BUGSest[j,2]<-mean(beta1)
BUGSest[j,3]<-mean(beta2)
BUGSest[j,4]<-mean(sigma.y)
}
detach.bugs()

# Calculation of a bias
average<-apply(BUGSest,2,ave)
BUGSsimest<-average[1,]
BUGSsimest

# Results formulation
BUGSresults<-matrix(0,nrow=7,ncol=4)
dimnames(BUGSresults)<-list(c("True value", "BUGS estimate", "BUGS bias", "MSE estimate",
"95% MSE lower bound", "95% MSE upper bound"),c("beta0", "beta1", "beta2", "sigma"))

BUGSresults[1,]<-truevalue
BUGSresults[2,]<-BUGSsimest
BUGSbias<-BUGSsimest-truevalue
BUGSresults[3,]<-BUGSbias

#Calculation of MSE
MSEwork<-apply((BUGSest-truevalue)^2,2,ave)
MSEest<-MSEwork[1,]
BUGSresults[4,]<-MSEest

# Calculation of the 95% approximate interval for MSE:
u<- (BUGSest-truevalue)^2
ubar<-apply(u,2,sum)/B
seMSEest<-sqrt(apply((u-ubar),2,sum)^2/(B*(B-1)))
BUGSresults[5,]<-seMSEest

lowbMSE<-MSEest-1.96*seMSEest
BUGSresults[6,]<-lowbMSE
upbMSE<-MSEest+1.96*seMSEest

```

```
BUGSresults[7,]<-upbmSE
cat("There are the following results for sample size of",T,"in",B,"simulations ")
BUGSresults
```

```
#####
```

```
# OLS AR(1) simulation
# Sampling from the specified distribution
```

```
beta0<-0
beta1<-1
beta2<-0.5
sigma2<-1
N<-1000
nburn<-0          #A number of the first simulated obs to be burnt
```

```
B<-100
OLSest<-matrix(0,nrow=B,ncol=4) #ncol=4 parameters of the distribution
out<-matrix(0,nrow=B,ncol=3)
time<-matrix(1,nrow=N,ncol=1)
X<-matrix(0,nrow=(N-nburn),ncol=2)
```

```
# Data simulation
```

```
for (j in 1:B){
  y<-matrix(0,nrow=N,ncol=1)
  ylag<-matrix(0,nrow=N,ncol=1)
  mu<-matrix(0,nrow=N,ncol=1)
  t<-1
  mu[t]<-beta0+beta1*t+beta2*0          #y[t-1] is unknown
  y[t]<-rnorm(1,mu[t],sigma)

  for (t in 2:N){
    mu[t]<-beta0+beta1*t+beta2*y[t-1]
    y[t]<-rnorm(1,mu[t],sigma)
    ylag[t]<-y[t-1]
    time[t]<-t
  }
}
```

```
X[,1]<-time[(nburn+1):N]
X[,2]<-ylag[(nburn+1):N]
Y<-y[(nburn+1):N]
```

```
# OLS regression
out<-lm(Y~X) #The 1st 200 obs are burnt out
summary(out)
out<-summary(out)
out
```

```
OLSest[j,1]<-out$coeff[1,1]
OLSest[j,2]<-out$coeff[2,1]
OLSest[j,3]<-out$coeff[3,1]
OLSest[j,4]<-out$sigma
```

```
}
# Calculation of a bias
average<-apply(OLSest,2,ave)
OLSsimest<-average[1,]
```

```

# Results formulation
results<-matrix(0,nrow=7,ncol=4)
dimnames(results)<-list(c("True value", "OLS estimate", "OLS bias", "MSE estimate",
"MSE SE", "95% MSE lower bound", "95% MSE upper bound"),c("beta0", "beta1", "beta2", "sigma"))

results[1,]<-truevalue
results[2,]<-OLSsimest

OLSbias<-OLSsimest-truevalue
OLSbias

results[3,]<-OLSbias

#Calculation of MSE
MSEwork<-apply((OLSest-truevalue)^2,2,ave)
MSEest<-MSEwork[1,]
results[4,]<-MSEest

# Calculation of the 95% approximate interval for MSE:
u<-(OLSest-truevalue)^2
ubar<-apply(u,2,sum)/B
seMSEest<-sqrt(apply((u-ubar),2,sum)^2/(B*(B-1)))
results[5,]<-seMSEest
lowbMSE<-MSEest-1.96*seMSEest
results[6,]<-lowbMSE
upbMSE<-MSEest+1.96*seMSEest
results[7,]<-upbMSE

cat("There are the following results for sample size",N-nburn,"in",B,"simulations ")
results

```

## Appendix E. A code with the model description from a file AR1Gamma.txt that was used in the simulation study

```
# 1. Model AR(1) specification
model #AR(1) with diffuse normal disns

var
Y[T], t;

{
theta.Y[1] <- beta0 + beta1*0 +beta2*1
Y[1]~ dnorm(theta.Y[1], tau.y)
for(t in 2:T)
  {
    Y[t] ~ dnorm(theta.Y[t], tau.y)
    theta.Y[t] <- beta0 + beta1*t + beta2*Y[t-1]
  }

# Priors
beta0 ~ dnorm(beta0.c, beta0.tau)    #non-informative diffuse normal distn
beta1 ~ dnorm(beta1.c, beta1.tau)    #non-informative diffuse normal distn
beta2 ~ dnorm(beta2.c, beta2.tau)    #non-informative diffuse normal distn
tau.y ~ dgamma(0.001,0.001)
sigma.y <-pow((1/tau.y), 0.5)

beta0.c ~ dnorm(0, 1.0E-6)  #non-informative diffuse normal distn
beta0.tau ~ dgamma(0.001, 0.001)
sigma.beta0 <- pow((1/beta0.tau), 0.5)

beta1.c ~ dnorm(0, 1.0E-6)  #non-informative diffuse normal distn
beta1.tau ~ dgamma(0.001, 0.001)
sigma.beta1<-pow((1/beta1.tau), 0.5)

beta2.c ~ dnorm(0, 1.0E-6)  #non-informative diffuse normal distn
beta2.tau ~ dgamma(0.001, 0.001)
sigma.beta2<-pow((1/beta2.tau), 0.5)
}
```

## Appendix F. Estimation summary from a test WinBUGS run of the research model with a generated sample of 1000 observations

**Table F.1. WinBUGS summary on the estimators' posterior distributions**

Node statistics								
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta0	0.04547	0.07509	0.00603	-0.1011	0.0497	0.1929	359	1074
beta1	1.002	0.04578	0.006088	0.962	0.9764	1.068	359	1074
beta2	-0.001982	0.04579	0.006088	-0.06801	0.02355	0.03792	359	1074
sigma.beta0	639.5	8306.0	247.7	0.04333	9.805	3937.0	359	1074
sigma.beta1	504.9	3050.0	90.06	0.04427	11.33	3735.0	359	1074
sigma.beta2	453.1	2044.0	60.28	0.05066	11.13	4037.0	359	1074
sigma.y	0.9809	0.02216	7.179E-4	0.9391	0.9809	1.027	359	1074

dic.stats()

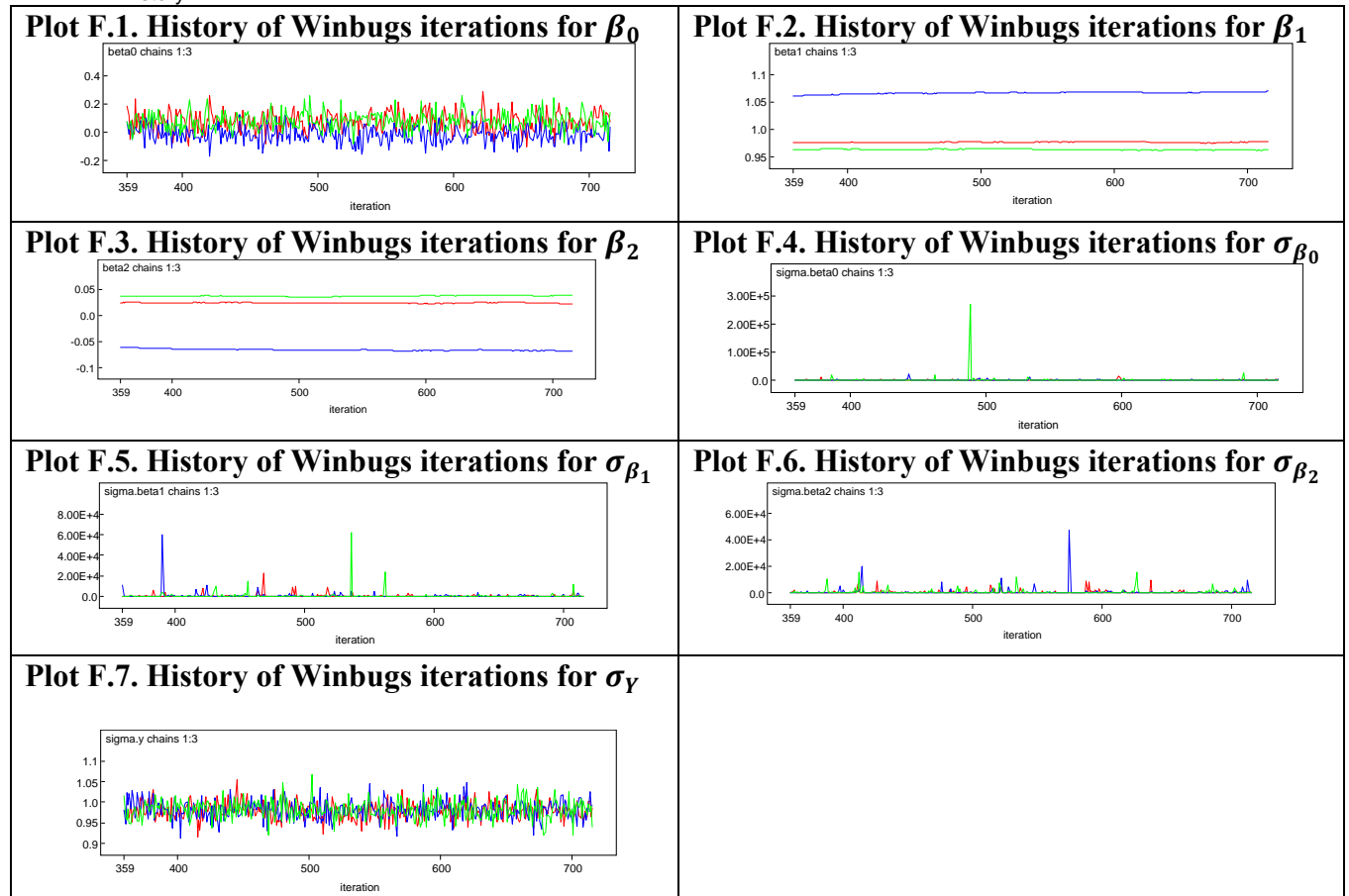
DIC

**Dbar = post.mean of -2logL; Dhat = -2LogL at post.mean of stochastic nodes**

	Dbar	Dhat	pD	DIC
Y	2800.030	2794.980	5.044	2805.070
total	2800.030	2794.980	5.044	2805.070

history(\*,D:/KSU\_Research/Final/Results/history.odc)

History



## Appendix G. Testing hypothesis of equality of the OLS and Gibbs Sampler MSE estimates in the research model

$$Y_t | y_{t-1}, \beta_0, \beta_1, \beta_2, \sigma_Y^2 \sim N_1(\beta_0 + \beta_1 t + \beta_2 y_{t-1}, \sigma_Y^2)$$

$$H_0: MSE_{OLS} = MSE_{Gibbs}$$

$$H_A: MSE_{OLS} \neq MSE_{Gibbs}$$

with a test statistic

$$T = \frac{\widehat{MSE}_{Gibbs} - \widehat{MSE}_{OLS}}{\sqrt{SE(\widehat{MSE}_{Gibbs})^2 + SE(\widehat{MSE}_{OLS})^2}} \sim t_{df=(n_{Gibbs}+n_{OLS}-2), (1-\alpha/2)}$$

where

$n_{Gibbs}$  is a size of the samples that were used in simulations to estimate Bayesian Gibbs Sampler MSE estimator,  $\widehat{MSE}_{Gibbs}$ ;

$n_{OLS}$  is a size of the samples that were used in simulations to estimate OLS MSE estimator,  $\widehat{MSE}_{OLS}$ .

The calculations below are based on the average Gibbs Sampler MSE estimate obtained from three chains with different initial values and on OLS MSE estimate for time trend coefficient  $\beta_1$ . All values used in these calculations are provided in Excel tables.

For a sample size  $n=10$

$$T = -6.186 \sim t_{df=16}$$

$$p\text{-value} = \text{Prob}(|t| \geq 6.186) = 1.3 \cdot 10^{-5}$$

### Conclusion

Since  $p$ -value of time trend coefficient,  $\beta_1$ , is less than 1%, there is strong evidence of significant difference between Gibbs Sampler MSE estimate and OLS MSE estimate. Gibbs Sampler MSE estimate tends to be smaller in average than OLS MSE.

For sample size  $n=30$

$$T = -3.327 \sim t_{df=56}$$

$$p\text{-value} = \text{Prob}(|t| \geq 3.327) = 0.0016$$

### Conclusion

Since  $p$ -value of time trend coefficient,  $\beta_1$ , is less than 1%, there is strong evidence of significant difference between Gibbs Sampler MSE estimate and OLS MSE estimate. Gibbs Sampler MSE estimate tends to be smaller in average than OLS MSE.

For sample size  $n=100$

$$T = -0.480 \sim t_{df=196}$$

$$p\text{-value} = \text{Prob}(|t| \geq 0.480) = 0.632$$

**Conclusion**

Since  $p$ -value of time trend coefficient,  $\beta_1$ , is more than 1%, there is no evidence of significant difference between Gibbs Sampler MSE estimate and OLS MSE estimate.

For sample size  $n=1,000$

$$T = -0.0153 \sim t_{df=1996}$$

$$p\text{-value} = \text{Prob}(|t| \geq 0.0153) = 0.988$$

**Conclusion**

Since  $p$ -value of time trend coefficient,  $\beta_1$ , is more than 1%, there is no evidence of significant difference between Gibbs Sampler MSE estimate and OLS MSE estimate.