STAT5020 Topics in Multivariate Analysis

Project Presentation

Chan Chun Ho, Benjamin, 1155049861*

April 3, 2018

Abstract

Structural Equation Model (SEM) is applied to a real data set called Abalone. Specifically, the linear SEM with multisample data is adopted. The R package R2WinBUGS is used to call WinBUGS to do the Bayesian analysis.

Contents

*For enquiry, please email to 1155049861@link.cuhk.edu.hk.

1 Introduction

- The real data set (Abalone) is downloaded from the 10th most popular data sets in the UC Irvine Machine Learning Repository on http://archive.ics.uci.edu/ml/index.php.

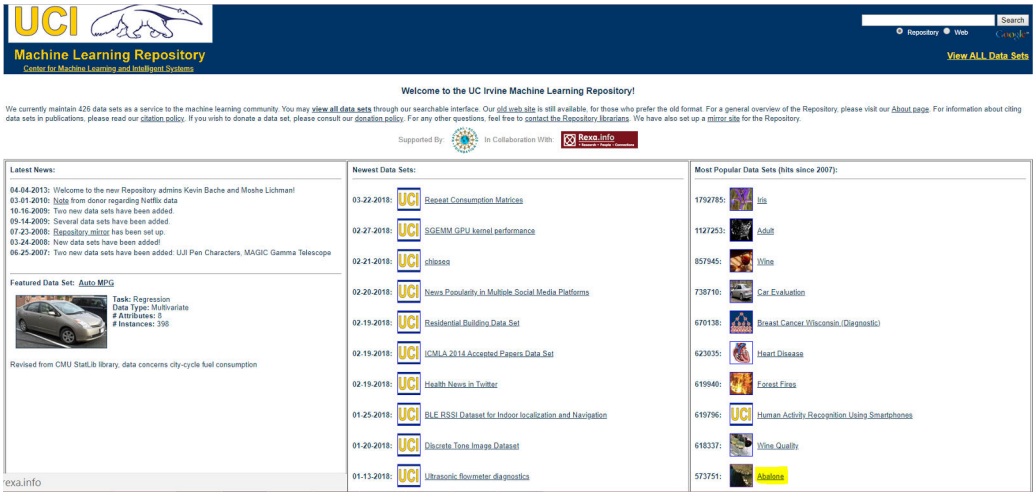

Figure 1: The UC Irvine Machine Learning Repository

- The Abalone data set is found on http://archive.ics.uci.edu/ml/datasets/Abalone. The data comes from an original study by Nash, Sellers, Talbot, Cawthorn and Ford. Its text format is stored on http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data.

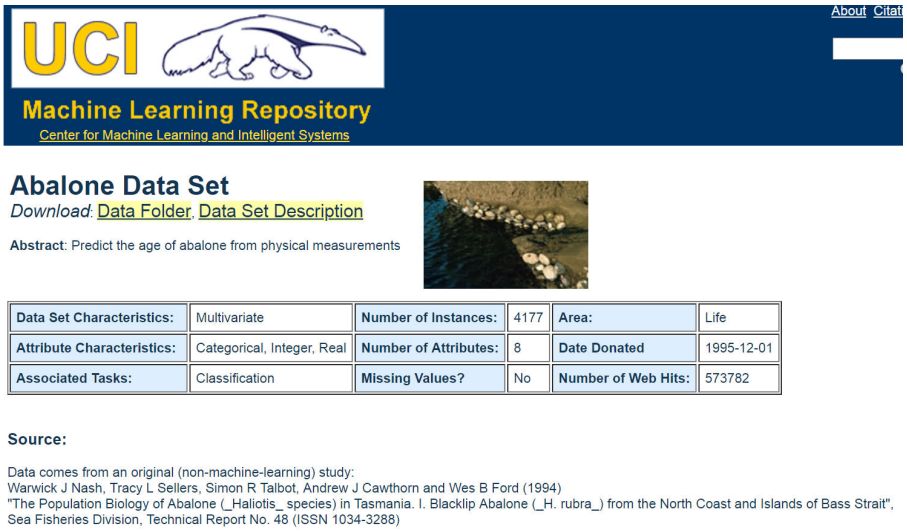

Figure 2: Abalone Data Set Online Description

- The primary objective is to predict the age of abalone from physical measurements. "The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope – a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age."

2 Data Set Description

- The attribute name, attribute type, the measurement unit and a brief description are given on the website as follows:

Name	Notation	Data Type	Measurement Unit	Description
Sex	N.A.	nominal	N.A.	M, F, and I (infant)
Rings	y_1	integer	N.A.	+1.5 gives the age in years
Length	y_2	continuous	mm	longest shell measurement
Diameter	y_3	continuous	mm	perpendicular to length
Height	y_4	continuous	mm	with meat in shell
Whole Weight	y_5	continuous	grams	whole abalone
Shucked Weight	y_6	continuous	grams	weight of meat
Viscera Weight	y_7	continuous	grams	gut weight (after bleeding)
Shell Weight	y_8	continuous	grams	after being dried

Table 1: Attribute Information

- The number of rings is the value to predict. Since it ranges from 1 to 29, it is treated as a continuous variable in the subsequent analyses. The histogram is in line with the view:

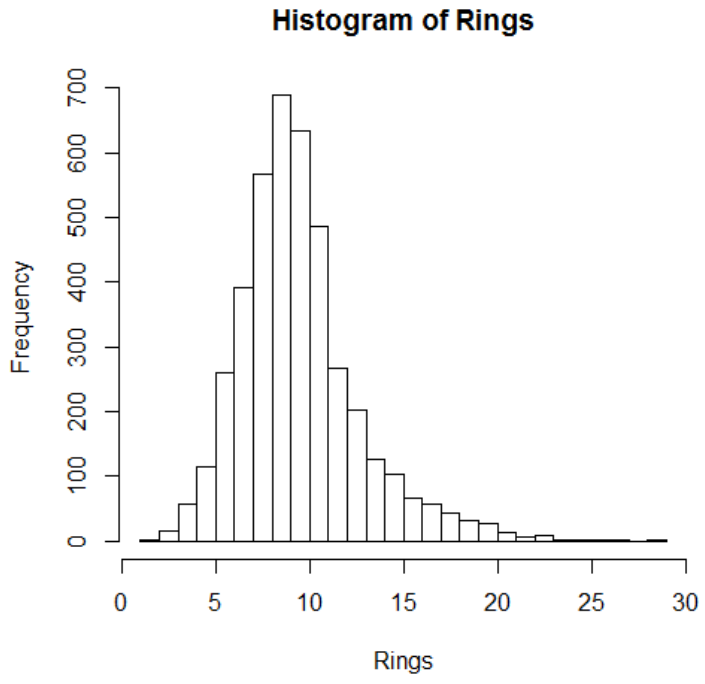

Figure 3: Histogram of Rings

- Excluding Sex, there are 8 continuous and observed variables each with 4177 observations in total without any missing value.

2.1 Latent Constructs

- The 8 observed variables are correlated as shown by the correlation matrix:

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
y_1	1	0.5567	0.5747	0.5575	0.5404	0.4209	0.5038	0.6276
y_2	0.5567	1	0.9868	0.8276	0.9253	0.8979	0.9030	0.8977
y_3	0.5747	0.9868	1	0.8337	0.9255	0.8932	0.8997	0.9053
y_4	0.5575	0.8276	0.8337	1	0.8192	0.7750	0.7983	0.8173
y_5	0.5404	0.9253	0.9255	0.8192	1	0.9694	0.9664	0.9554
y_6	0.4209	0.8979	0.8932	0.7750	0.9694	1	0.9320	0.8826
y_7	0.5038	0.9030	0.8997	0.7983	0.9664	0.9320	1	0.9077
y_8	0.6276	0.8977	0.9053	0.8173	0.9554	0.8826	0.9077	1

Table 2: Correlation Matrix for All Sexes

- In this study, structural equation model (SEM) is a powerful multivariate tool because it can simultaneously group highly correlated variables into latent variables and assess interrelationships among latent variables through a regression model of latent variables.[1]
- As the variable description suggests, "Length" (y_2), "Diameter" (y_3) and "Height" (y_4) should be combined to a latent variable "Size" of abalone.
- Moreover, "Whole Weight" (y_5), "Shucked Weight" (y_6), "Viscera Weight" (y_7) and "Shell Weight" (y_8) should be combined to a latent variable "Weight" of abalone.
- Lastly, "Rings" (y_1) plus 1.5 gives the age in years so it is related to "Age" of abalone.

Latent Variable	Notation	Related Observed Variable(s)
Age	η	y_1
Size	ξ_1	y_2, y_3, y_4
Weight	ξ_2	y_5, y_6, y_7, y_8

Table 3: Latent Constructs

2.2 Multisample Data Structure

- Multisample data come from a number of distinct groups/ populations. In this data set, there are three sexes, namely "Male", "Female" and "Infant", each belonging to a group.
- Note that the word "Sex" and "Group" are used interchangeably in multisample setting.

[1] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch1

- The number of observations for each sex is:

Sex	Notation	Sample Index	Number of Observations	
Male	M	1	N_1	1528
Female	F	2	N_2	1307
Infant	I	3	N_3	1342

Table 4: Number of Observations for Each Sex

- For multisample data, it is assumed that the observations within each group (sex) are independent rather than correlated. The number of groups is 3 and the group membership of each observation has been specified.
- The objective is to investigate the similarities or differences among the models in the different groups (sexes). The main interest is the testing of hypotheses about the different kinds of invariances among the models in different groups.
- In Bayesian analysis, each hypothesis of interest is associated with a model, and the problem is approached through model comparison and can be addressed by DIC.[2]
- The unstandardized (denoted by *) sample mean vector for each sex is:

	$\bar{y}_1^*$	$\bar{y}_2^*$	$\bar{y}_3^*$	$\bar{y}_4^*$	$\bar{y}_5^*$	$\bar{y}_6^*$	$\bar{y}_7^*$	$\bar{y}_8^*$
M	10.7055	0.5614	0.4393	0.1514	0.9915	0.4329	0.2155	0.2820
F	11.1293	0.5791	0.4547	0.1580	1.0465	0.4462	0.2307	0.3020
I	7.8905	0.4277	0.3265	0.1080	0.4314	0.1910	0.0920	0.1282
Overall	9.9337	0.5240	0.4079	0.1395	0.8287	0.3594	0.1806	0.2388

Table 5: Unstandardized Sample Mean Vector for Each Sex

- From Table 5, the observed variables for "Male" are on average smaller in values than those for "Female" but on average larger in values than those for "Infant".
- To unify scales of variables, the raw continuous data are standardized:

	$\bar{y}_1$	$\bar{y}_2$	$\bar{y}_3$	$\bar{y}_4$	$\bar{y}_5$	$\bar{y}_6$	$\bar{y}_7$	$\bar{y}_8$
M	0.2394	0.3114	0.3165	0.2837	0.3318	0.3315	0.3189	0.3099
F	0.3708	0.4588	0.4721	0.4422	0.4441	0.3911	0.4570	0.4539
I	-0.6337	-0.8014	-0.8201	-0.7536	-0.8103	-0.7584	-0.8081	-0.7949

Table 6: Standardized Sample Mean Vector for Each Sex

2.3 Foreword

- Due to heterogeneity introduced by Sex and latent constructs of variables, for the time being, consider the multisample model with linear structural equation.
- Formal hypothesis testing of constraints or model comparison with the multisample models with non-linear structural equation are done in Section 4.1 and 4.2.

[2] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch6

3 Linear SEM with Multisample Data

3.1 Model Formulation

- Linear SEMs are formulated with a measurement equation and a structural equation with linear terms of explanatory latent variables. Under the assumption that the observed variables in each group are continuous, and identically and independently distributed with a normal distribution, the linear multisample SEM is the most basic.[3]
- Consider 3 independent groups (sexes) of abalones that represent different populations. The sample indices for "Male", "Female" and "Infant" are 1, 2 and 3 respectively. Their corresponding sample sizes are $N_1 = 1528$, $N_2 = 1307$ and $N_3 = 1342$.
- For $g = 1, 2, 3$ and $i = 1, \ldots, N_g$, let $\boldsymbol{y}_i^{(g)}$ be the 8×1 random vector of observed variables that correspond to the ith observation (abalone) in the gth group. In the gth group, $\boldsymbol{y}_i^{(g)}$ are assumed to be independent.[4]

3.1.1 Measurement Equation

- For each $g = 1, 2, 3$, $\boldsymbol{y}_i^{(g)}$ is related to latent variables in a 3×1 random vector $\boldsymbol{\omega}_i^{(g)}$ through the measurement equation:

$$\boldsymbol{y}_i^{(g)} = \boldsymbol{\mu}^{(g)} + \boldsymbol{\Lambda}^{(g)}\boldsymbol{\omega}_i^{(g)} + \boldsymbol{\epsilon}_i^{(g)}, \quad g = 1, 2, 3, \tag{1}$$

  where $\boldsymbol{y}_i^{(g)}$ is a 8×1 vector of observed variables, $\boldsymbol{\mu}^{(g)}$ is a 8×1 vector of intercepts, $\boldsymbol{\Lambda}^{(g)}$ is a 8×3 parameter matrix of unknown regression coefficients, $\boldsymbol{\omega}_i^{(g)}$ is a 3×1 vector of latent variables, and $\boldsymbol{\epsilon}_i^{(g)}$ is a 8×1 random vector of error measurements.
- It is assumed that $\boldsymbol{\omega}_i^{(g)}$ and $\boldsymbol{\epsilon}_i^{(g)}$ are independent, and the distribution of $\boldsymbol{\epsilon}_i^{(g)}$ is $N[\mathbf{0}, \boldsymbol{\Psi}_{\boldsymbol{\epsilon}}^{(g)}]$, where $\boldsymbol{\Psi}_{\boldsymbol{\epsilon}}^{(g)} = \text{diag}(\psi_{\epsilon 1}^{(g)}, \ldots, \psi_{\epsilon 8}^{(g)})$ is diagonal. Let $\boldsymbol{\omega}_i^{(g)} = (\eta_i^{(g)T}, \boldsymbol{\xi}_i^{(g)T})^T$. It is natural to assume that the dimensions of $\eta_i^{(g)}$ and $\boldsymbol{\xi}_i^{(g)}$ are the same for g, which are 1 and 2 respectively.

3.1.2 Structural Equation

- To assess the effects of the linear terms of latent variables in $\boldsymbol{\xi}_i^{(g)}$ on $\eta_i^{(g)}$, a linear SEM is with linear structural equation:

$$\eta_i^{(g)} = \boldsymbol{\Gamma}^{(g)}\boldsymbol{\xi}_i^{(g)} + \delta_i^{(g)}, \quad g = 1, 2, 3, \tag{2}$$

  where $\eta_i^{(g)}$ is a scalar of outcome latent variable, $\boldsymbol{\Gamma}^{(g)}$ is a 1×2 parameter matrix of unknown regression coefficients, $\boldsymbol{\xi}_i^{(g)}$ is a 2×1 vector of explanatory latent variables, and $\delta_i^{(g)}$ is a scalar of residual error.
- It is assumed that $\boldsymbol{\xi}_i^{(g)}$ and $\delta_i^{(g)}$ are independent, the distribution of $\boldsymbol{\xi}_i^{(g)}$ and $\delta_i^{(g)}$ are $N[\mathbf{0}, \boldsymbol{\Phi}^{(g)}]$ and $N[\mathbf{0}, \psi_\delta^{(g)}]$ respectively, where $\boldsymbol{\Phi}^{(g)}$ is a general 2×2 covariance matrix and $\psi_\delta^{(g)}$ is a scalar.

[3] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch2
[4] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch6

3.2 Model Identification and Interpretation

- Any element of $\mathbf{\Lambda}^{(g)}$ can be a free parameter or a fixed parameter with a preassigned value. The positions and the preassigned values of fixed parameters are decided on the basis of the prior knowledge of the observed variables and latent variables. Moreover, they are related to the interpretations of latent variables.[5]

- In applications of SEMs, parameter matrices of unknown regression coefficients with non-overlapping structures are frequently used as in this study. They are used for identification purpose.

3.2.1 Structure in Measurement Equation

- In matrix representation, the measurement equation $\boldsymbol{y}_i^{(g)} = \boldsymbol{\mu}^{(g)} + \mathbf{\Lambda}^{(g)}\boldsymbol{\omega}_i^{(g)} + \boldsymbol{\epsilon}_i^{(g)}$ becomes:

$$\begin{bmatrix} y_{i1}^{(g)} \\ y_{i2}^{(g)} \\ y_{i3}^{(g)} \\ y_{i4}^{(g)} \\ y_{i5}^{(g)} \\ y_{i6}^{(g)} \\ y_{i7}^{(g)} \\ y_{i8}^{(g)} \end{bmatrix} = \begin{bmatrix} \mu_{i1}^{(g)} \\ \mu_{i2}^{(g)} \\ \mu_{i3}^{(g)} \\ \mu_{i4}^{(g)} \\ \mu_{i5}^{(g)} \\ \mu_{i6}^{(g)} \\ \mu_{i7}^{(g)} \\ \mu_{i8}^{(g)} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \lambda_{32}^{(g)} & 0 \\ 0 & \lambda_{42}^{(g)} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \lambda_{63}^{(g)} \\ 0 & 0 & \lambda_{73}^{(g)} \\ 0 & 0 & \lambda_{83}^{(g)} \end{bmatrix} \begin{bmatrix} \eta_i^{(g)} \\ \xi_{i1}^{(g)} \\ \xi_{i2}^{(g)} \end{bmatrix} + \begin{bmatrix} \epsilon_{i1}^{(g)} \\ \epsilon_{i2}^{(g)} \\ \epsilon_{i3}^{(g)} \\ \epsilon_{i4}^{(g)} \\ \epsilon_{i5}^{(g)} \\ \epsilon_{i6}^{(g)} \\ \epsilon_{i7}^{(g)} \\ \epsilon_{i8}^{(g)} \end{bmatrix}, \quad g = 1, 2, 3, \tag{3}$$

  where elements 1's and 0's in $\mathbf{\Lambda}^{(g)}$ are known parameters with the fixed values, and the other $\lambda_{jk}^{(g)}$'s are unknown parameters. The fixed value 1 is used to introduce a scale to the corresponding latent variable.

- The measurement equation is identified via fixing appropriate parameters.

- Note that $\lambda_{jk}^{(g)}$ is the coefficient linking the observed variables $y_{ij}^{(g)}$ with the latent variable $\omega_{ik}^{(g)}$. Here i is the observation index, $\omega_{i1}^{(g)} = \eta_i^{(g)}$, $\omega_{i2}^{(g)} = \xi_{i1}^{(g)}$ and $\omega_{i3}^{(g)} = \xi_{i2}^{(g)}$. For instance,

  1. $\eta_i^{(g)}$ is only linked with $y_{i1}^{(g)}$,
  2. $\xi_{i1}^{(g)}$ is only linked with $y_{i2}^{(g)}$, $y_{i3}^{(g)}$ and $y_{i4}^{(g)}$,
  3. $\xi_{i2}^{(g)}$ is only linked with $y_{i5}^{(g)}$, $y_{i6}^{(g)}$, $y_{i7}^{(g)}$ and $y_{i8}^{(g)}$,

  which is in line with the latent constructs given in Table 3. As a result, the non-overlapping structure makes the interpretation of the latent variables clear. In other words, the measurement equation is a confirmatory tool.[6]

[5] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch2
[6] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch2

3.2.2 Structure in Structural Equation

- In scalar representation, the structural equation $\eta_i^{(g)} = \mathbf{\Gamma}^{(g)}\boldsymbol{\xi}_i^{(g)} + \delta_i^{(g)}$ becomes:

$$\eta_i^{(g)} = \gamma_1^{(g)}\xi_{i1}^{(g)} + \gamma_2^{(g)}\xi_{i2}^{(g)} + \delta_i^{(g)}, \quad g = 1, 2, 3. \tag{4}$$

- The structural equation is identified with identified $\eta_i^{(g)}$, $\xi_{i1}^{(g)}$ and $\xi_{i2}^{(g)}$ for all i and g.

- Note that $\gamma_1^{(g)}$ and $\gamma_2^{(g)}$ represent the magnitude of the expected changes in $\eta_i^{(g)}$ for one unit change in $\xi_{i1}^{(g)}$ and $\xi_{i2}^{(g)}$ respectively. The outcome latent variable is only partially explained by the explanatory latent variables, the unexplained part is taken into account by the residual error $\delta_i^{(g)}$.[7]

3.3 Bayesian Estimation

- In Bayesian analysis, Markov chain Monte Carlo (MCMC) algorithms are used to simulate a large number of observations from the joint posterior distribution.

- In SEMs, the Bayesian esimates of the unknown parameters and the latent variables can be obtained from the corresponding sample means of observations simulated from the posterior distribution.[8]

- In this study, the software WinBUGS is used via the R package R2WinBUGS to do the Bayesian inferences.

- For simplicity, the conjugate prior distributions are used and hyperparameters are specified. First, prior distributions for nonconstrained parameters in different groups are naturally assumed to be independent.

- In the gth group model, let $\psi_{\epsilon k}^{(g)}$ and $\mathbf{\Lambda}_k^{(g)T}$ be the kth diagonal element of $\mathbf{\Psi}_{\boldsymbol{\epsilon}}^{(g)}$ and the kth row of $\mathbf{\Lambda}^{(g)}$ respectively. If parameters are not invariant over groups, the joint prior distribution could be taken as[9]

$$\psi_{\epsilon k}^{(g)-1} \overset{D}{=} \text{Gamma}[\alpha_{0\epsilon k}^{(g)}, \beta_{0\epsilon k}^{(g)}], \;\; [\mathbf{\Lambda}_k^{(g)} | \psi_{\epsilon k}^{(g)}] \overset{D}{=} N[\mathbf{\Lambda}_{0k}^{(g)}, \psi_{\epsilon k}^{(g)} \boldsymbol{H}_{0yk}^{(g)}], \tag{5}$$

  where $\alpha_{0\epsilon k}^{(g)}$, $\beta_{0\epsilon k}^{(g)}$, $\mathbf{\Lambda}_{0k}^{(g)}$ and $\boldsymbol{H}_{0yk}^{(g)}$ are hyperparameters.

- As pointed out by Song and Lee (2012), the joint prior distribution in (5) is not applicable under the constrained situations, e.g. when $\mathbf{\Lambda}^{(1)} = \mathbf{\Lambda}^{(2)} = \mathbf{\Lambda}^{(3)}$. The constrained situations are considered under hypothesis testing/ model comparison problems in later sections.

- To facilitate comparison in Section 4.1 and 4.2, and follow the suggestion in the book, independent prior distributions for $\mathbf{\Lambda}^{(g)}$ and $\mathbf{\Psi}_{\boldsymbol{\epsilon}}^{(g)}$ are selected with or without constraints such that

$$p(\mathbf{\Lambda}^{(g)}, \mathbf{\Psi}_{\boldsymbol{\epsilon}}^{(g)}) = p(\mathbf{\Lambda}^{(g)}) p(\mathbf{\Psi}_{\boldsymbol{\epsilon}}^{(g)}), \quad g = 1, 2, 3.$$

[7] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch2
[8] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch3
[9] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch6

- Independent prior distributions for $\boldsymbol{\Gamma}^{(g)}$ and $\psi_{\delta}^{(g)}$ are selected with or without constraints such that

$$p(\boldsymbol{\Gamma}^{(g)}, \psi_{\delta}^{(g)}) = p(\boldsymbol{\Gamma}^{(g)})p(\psi_{\delta}^{(g)}), \quad g = 1, 2, 3.$$

3.3.1 Conjugate Prior Distributions

- In the measurement equation, the conjugate prior distributions of $\boldsymbol{\mu}^{(g)}$, $\boldsymbol{\Lambda}_k^{(g)}$ and $\psi_{\epsilon k}^{(g)}$ are[10]

$$\psi_{\epsilon k}^{(g)-1} \overset{D}{=} \text{Gamma}[\alpha_{0\epsilon k}^{(g)}, \beta_{0\epsilon k}^{(g)}], \;\; \boldsymbol{\mu}^{(g)} \overset{D}{=} N[\boldsymbol{\mu}_0^{(g)}, \boldsymbol{\Sigma}_0^{(g)}] \text{ and } \boldsymbol{\Lambda}_k^{(g)} \overset{D}{=} N[\boldsymbol{\Lambda}_{0k}^{(g)}, \boldsymbol{H}_{0yk}^{(g)}] \tag{6}$$

  for $g = 1, 2, 3$, where $\alpha_{0\epsilon k}^{(g)}$, $\beta_{0\epsilon k}^{(g)}$, and elements in $\boldsymbol{\mu}_0^{(g)}$, $\boldsymbol{\Lambda}_{0k}^{(g)}$, $\boldsymbol{\Sigma}_0^{(g)}$ and $\boldsymbol{H}_{0yk}^{(g)}$ are hyperparameters, and $\boldsymbol{\Sigma}_0^{(g)}$ and $\boldsymbol{H}_{0yk}^{(g)}$ are positive definite matrices.

- In the structural equation, the conjugate prior distributions of $\boldsymbol{\Phi}^{(g)}$, $\boldsymbol{\Gamma}^{(g)}$ and $\psi_{\delta}^{(g)}$ are

$$\boldsymbol{\Phi}^{(g)-1} \overset{D}{=} W_2[\boldsymbol{R}_0^{(g)}, \rho_0], \;\; \psi_{\delta}^{(g)-1} \overset{D}{=} \text{Gamma}[\alpha_{0\delta}^{(g)}, \beta_{0\delta}^{(g)}] \text{ and } \boldsymbol{\Gamma}^{(g)} \overset{D}{=} N[\boldsymbol{\Gamma}_0^{(g)}, \boldsymbol{H}_{0\omega}^{(g)}] \tag{7}$$

  for $g = 1, 2, 3$, where ρ_0, $\alpha_{0\delta}^{(g)}$, $\beta_{0\delta}^{(g)}$, and elements in $\boldsymbol{R}_0^{(g)}$, $\boldsymbol{\Gamma}_0^{(g)}$ and $\boldsymbol{H}_{0\omega}^{(g)}$ are hyperparameters, and $\boldsymbol{R}_0^{(g)}$ and $\boldsymbol{H}_{0\omega}^{(g)}$ are positive definite matrices.

3.3.2 Specified Hyperparameters

- Since there is no available prior knowledge of the parameters, the hyperparameters are specified for convenience and/ or to have vague priors. No prior knowledge of the group difference is available, thus the hyperparameters are set to be equal for all groups.

- With reference to (6), the hyperparameters used in the measurement equation are:

$$\alpha_{0\epsilon k}^{(g)} = 6, \;\; \beta_{0\epsilon k}^{(g)} = 10, \;\; \boldsymbol{\mu}_0^{(g)} = \mathbf{0}, \;\; \boldsymbol{\Sigma}_0^{(g)} = 2.5\boldsymbol{I}, \;\; \boldsymbol{\Lambda}_{0k}^{(g)} = 0.8\mathbf{1} \text{ and } \boldsymbol{H}_{0yk}^{(g)} = 2.5\boldsymbol{I}$$

  for $g = 1, 2, 3$, where $\boldsymbol{I}$ is an identity matrix with appropriate dimension and $\mathbf{1}$ is a vector with every element 1 and appropriate dimension.

- With reference to (7), the hyperparameters used in the structural equation are:

$$\boldsymbol{R}_0^{(g)} = 8\boldsymbol{I}, \;\; \rho_0 = 4, \;\; \alpha_{0\delta}^{(g)} = 6, \;\; \beta_{0\delta}^{(g)} = 10, \;\; \boldsymbol{\Gamma}_0^{(g)} = 0.6\mathbf{1} \text{ and } \boldsymbol{H}_{0\omega}^{(g)} = 2.5\boldsymbol{I}$$

  for $g = 1, 2, 3$, where $\boldsymbol{I}$ is an identity matrix with appropriate dimension and $\mathbf{1}$ is a vector with every element 1 and appropriate dimension.

- Under this choice, the means of $\psi_{\epsilon k}^{(g)}$ and $\psi_{\delta}^{(g)}$ are 2 and their variances are 1 for all g. The variability of $\boldsymbol{\mu}^{(g)}$, $\boldsymbol{\Lambda}_k^{(g)}$ and $\boldsymbol{\Gamma}^{(g)}$ is large because their covariance matrices are set to be $2.5\boldsymbol{I}$.

[10] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch3

3.3.3 Initial Values of Parameters

- There are two sets of initial values of parameters (denoted by *) used in the measurement equation defined in (1) or (3) and the structural equation defined in (2) or (4).

- The first set of initial values is:

$$\boldsymbol{\mu}^{(1)*} = \boldsymbol{\mu}^{(2)*} = \boldsymbol{\mu}^{(3)*} = \mathbf{0},\ \ \lambda_{jk}^{(1)*} = \lambda_{jk}^{(2)*} = \lambda_{jk}^{(3)*} = 0 \ \text{ for valid } j,k,$$
$$\gamma_1^{(1)*} = \gamma_1^{(2)*} = \gamma_1^{(3)*} = 0,\ \ \gamma_2^{(1)*} = \gamma_2^{(2)*} = \gamma_2^{(3)*} = 0,\quad \psi_{\epsilon 1}^{(1)*} = \psi_{\epsilon 1}^{(2)*} = \psi_{\epsilon 1}^{(3)*} = 0,$$
$$\psi_{\epsilon j}^{(1)*} = \psi_{\epsilon j}^{(2)*} = \psi_{\epsilon j}^{(3)*} = 1 \ \text{ for } j = 2,\ldots,8,\ \ \psi_{\delta}^{(1)*} = \psi_{\delta}^{(2)*} = \psi_{\delta}^{(3)*} = 1,\ \ \boldsymbol{\Phi}^{(1)*} = \boldsymbol{\Phi}^{(2)*} = \boldsymbol{\Phi}^{(3)*} = \boldsymbol{I}.$$

- The second set of initial values is:

$$\mu_{i1}^{(1)*} = \mu_{i1}^{(2)*} = \mu_{i1}^{(3)*} = 0,\ \ \mu_{ij}^{(1)*} = \mu_{ij}^{(2)*} = \mu_{ij}^{(3)*} = 1 \ \text{ for } j = 2,\ldots,8,$$
$$\lambda_{jk}^{(1)*} = \lambda_{jk}^{(2)*} = \lambda_{jk}^{(3)*} = 1 \ \text{ for valid } j,k,\ \ \boldsymbol{\Phi}^{(1)*} = \boldsymbol{\Phi}^{(2)*} = \boldsymbol{\Phi}^{(3)*} = 0.5\boldsymbol{I},$$
$$\gamma_1^{(1)*} = \gamma_1^{(2)*} = \gamma_1^{(3)*} = 1,\ \ \gamma_2^{(1)*} = \gamma_2^{(2)*} = \gamma_2^{(3)*} = 1,\quad \psi_{\epsilon 1}^{(1)*} = \psi_{\epsilon 1}^{(2)*} = \psi_{\epsilon 1}^{(3)*} = 0,$$
$$\psi_{\epsilon j}^{(1)*} = \psi_{\epsilon j}^{(2)*} = \psi_{\epsilon j}^{(3)*} = 0.5 \ \text{ for } j = 2,\ldots,8,\ \ \psi_{\delta}^{(1)*} = \psi_{\delta}^{(2)*} = \psi_{\delta}^{(3)*} = 0.5.$$

3.4 Main Results

- In using R2WinBUGS to do Bayesian estimation, the burn-in iterations required for achieving convergence of the MCMC algorithms are set to be 5,000. In fact, it can be checked by plots of the simulated sequences of the individual parameters.[11]

- The number of simulated observations collected after the burn-in iterations is set to be 5,000, i.e. $\{(\boldsymbol{\theta}^{(t)}, \boldsymbol{\Omega}^{(t)}) : t = 1,\ldots,5000\}$. In other words, the total number of iterations run by the program is 10,000.

- The running mean plots of different parameters, generated by the two different initial values (given in Section 3.3.3), are shown in Appendix from Figure 4 to Figure 21.

- From Figure 4 to Figure 21, the two sequences generated with different initial values mix well. It is concluded that convergence is achieved and hence 5,000 burn-in iterations are enough for the data set.

- In this study, the posterior means (the elements of $\hat{\boldsymbol{\theta}}$) and standard error estimates (the positive square roots of the diagonal elements in $\widehat{\mathrm{Var}}(\boldsymbol{\theta}|\boldsymbol{Y})$) of parameters are reported for Bayesian analysis:

$$\hat{\boldsymbol{\theta}} = \frac{1}{5,000}\sum_{i=1}^{5,000}\boldsymbol{\theta}^{(t)}, \qquad \widehat{\mathrm{Var}}(\boldsymbol{\theta}|\boldsymbol{Y}) = \frac{1}{4,999}\sum_{i=1}^{5,000}(\boldsymbol{\theta}^{(t)} - \hat{\boldsymbol{\theta}})(\boldsymbol{\theta}^{(t)} - \hat{\boldsymbol{\theta}})^T.$$

- With the linear SEM with multisample data defined in Section 3.1 and 3.2, prior distributions, hyperparameters and initial values of parameters given in Section 3.3, the posterior means and standard error estimates of parameters in linear SEM with multisample data are shown in Appendix Table 11.

[11] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch3

3.4.1 Intuitive Interpretation of Results

- Full table of Bayesian estimates of parameters is given in Table 11. In general, the standard error estimates are small as compared with the corresponding means. It indicates the reliability of the Bayesian estimates. The running mean plots further support the claim.
- Informally speaking, it is observed that the estimates of parameters are different in the 3 groups. The formal hypothesis testing of equal $\boldsymbol{\mu}^{(g)}$, $\boldsymbol{\Lambda}^{(g)}$, $\boldsymbol{\Gamma}^{(g)}$ and $\boldsymbol{\Phi}^{(g)}$ for all g is given in Section 4.2.
- First of all, the values of $\hat{\mu}_j^{(1)}$'s are smaller than those of $\hat{\mu}_j^{(2)}$'s but larger than those of $\hat{\mu}_j^{(3)}$'s. In fact, it is in accordance with the group standardized sample means found in Table 6.
- Of course, they are not exactly equal because the Bayesian estimates have incorporated prior inputs. Since linear SEM is used, the intercepts $\hat{\mu}_j^{(g)}$ are the posterior means of $y_j^{(g)}$:

	$\bar{y}_1^{(g)}$	$\bar{y}_2^{(g)}$	$\bar{y}_3^{(g)}$	$\bar{y}_4^{(g)}$	$\bar{y}_5^{(g)}$	$\bar{y}_6^{(g)}$	$\bar{y}_7^{(g)}$	$\bar{y}_8^{(g)}$
g=1	0.2394	0.3114	0.3165	0.2837	0.3318	0.3315	0.3189	0.3099
g=2	0.3708	0.4588	0.4721	0.4422	0.4441	0.3911	0.4570	0.4539
g=3	-0.6337	-0.8014	-0.8201	-0.7536	-0.8103	-0.7584	-0.8081	-0.7949
	$\hat{\mu}_1^{(g)}$	$\hat{\mu}_2^{(g)}$	$\hat{\mu}_3^{(g)}$	$\hat{\mu}_4^{(g)}$	$\hat{\mu}_5^{(g)}$	$\hat{\mu}_6^{(g)}$	$\hat{\mu}_7^{(g)}$	$\hat{\mu}_8^{(g)}$
g=1	0.2378	0.3080	0.3131	0.2806	0.3279	0.3275	0.3152	0.3066
g=2	0.3702	0.4575	0.4708	0.4410	0.4426	0.3897	0.4556	0.4525
g=3	-0.6337	-0.8019	-0.8204	-0.7538	-0.8108	-0.7588	-0.8086	-0.7953

Table 7: Standardized Sample Means of $\boldsymbol{y}$ and Posterior Means of $\boldsymbol{\mu}^{(g)}$

- In other words, the intercepts for each group serve as a baseline for 8 observed variables for the respective group in addition to the effects of 3 latent variables.
- Second point to note in Table 11 is that the values of $\lambda_{jk}^{(g)}$'s are large (some are even close to 1). It indicates that there is a strong relationship between the observed variables and the latent variables.
- Moreover, all the values of $\hat{\lambda}_{32}^{(g)}$ and $\hat{\lambda}_{42}^{(g)}$ are smaller than 1 for all g while nearly all (with two exceptions) the value of $\hat{\lambda}_{63}^{(g)}$, $\hat{\lambda}_{73}^{(g)}$ and $\hat{\lambda}_{83}^{(g)}$ are smaller than 1 for all g.

	$\hat{\lambda}_{32}^{(g)}$	$\hat{\lambda}_{42}^{(g)}$	$\hat{\lambda}_{63}^{(g)}$	$\hat{\lambda}_{73}^{(g)}$	$\hat{\lambda}_{83}^{(g)}$
g=1	0.9912	0.8502	1.0105	0.9610	0.9147
g=2	0.9866	0.7914	0.9728	0.9681	0.9485
g=3	0.9782	0.7938	0.9537	0.9490	1.0124

Table 8: Posterior Means of $\boldsymbol{\Lambda}^{(g)}$

- As mentioned in Section 3.2.1, the fixed value 1 in $\boldsymbol{\Lambda}^{(g)}$ is used to introduce a scale to the corresponding latent variable. Therefore, $\hat{\lambda}_{jk}^{(g)}$ should be interpreted in a relative sense.

- In other words, "Length" (y_2) are more related to "Size" (ξ_1) than "Diameter" (y_3) and "Height" (y_4) while "Whole Weight" (y_5) are more related to "Weight" (ξ_2) than "Shucked Weight" (y_6), "Viscera Weight" (y_7) and "Shell Weight" (y_8). In addition, "Diameter" is more related to "Size" than "Height".

- Thirdly, $\mathbf{\Gamma}^{(1)}$, $\mathbf{\Gamma}^{(2)}$ and $\mathbf{\Gamma}^{(3)}$ are quite different, meaning that there is a heterogeneity of groups in the relationship between the outcome latent variable and the explanatory latent variables:

	$\gamma_1^{(1)}$	$\gamma_2^{(1)}$	$\gamma_1^{(2)}$	$\gamma_2^{(2)}$	$\gamma_1^{(3)}$	$\gamma_2^{(3)}$
Mean	0.5305	-0.0863	0.2607	0.0773	0.3751	0.3939
SE	0.0984	0.0870	0.1717	0.1389	0.0662	0.1023

Table 9: Posterior Means and Standard Error Estimates of $\mathbf{\Gamma}^{(g)}$

- From Table 9, the posterior means and standard error estimates suggest that the first regression parameter in the structural equation is significant for all groups. However, the second regression parameter is not so significant for Group 1 and 2 but significant for Group 3.

- In other words, "Size" is useful in explaining "Age" of all abalones. However, "Weight" is only useful in explaining "Age" of abalones for "Infant" but not for "Male" and "Female".

- The non-significance of "Weight" for "Male" and "Female" can be explained in regression terminology. All the observed variables are correlated as revealed in Table 2. In the presence of latent variable "Size", another latent variable "Weight" is no longer important in predicting "Age" for "Male" and "Female".

- Last but not least, $\mathbf{\Phi}^{(1)}$, $\mathbf{\Phi}^{(2)}$ and $\mathbf{\Phi}^{(3)}$ are quite different, meaning that there is a heterogeneity of the interrelationship among the 2 explanatory latent variables for groups:

	$\hat{\phi}_{11}^{(g)}$	$\hat{\phi}_{12}^{(g)}$	$\hat{\phi}_{22}^{(g)}$
g=1	0.7132	0.7538	0.9074
g=2	0.5020	0.5831	0.7581
g=3	0.8040	0.4906	0.3431

Table 10: Posterior Means of $\mathbf{\Phi}^{(g)}$

- In other words, the interrelationship between "Size" and "Weight" is different in each group. Of course, with $\hat{\phi}_{12}^{(g)}$ significantly above 0, "Size" and "Weight" are correlated.

- In summary, due to the heterogeneity introduced by "Sex", multisample model is powerful in capturing the group difference. With linear structural equation, the interpretation of parameters is intuitive.

3.5 Sensitivity Analysis

- Since there is no useful prior information, a sensitivity analysis should be conducted to investigate whether the results are robust to prior inputs.[12]
- The linear SEM with multisample data defined in Section 3.1 and 3.2, conjugate type prior distributions and initial values of parameters given in Section 3.3 remain unchanged. However, the hyperparameter values are perturbed to give two new prior inputs.
- For the first new prior inputs, the hyperparameters used in the measurement equation are reset to be:

$$\alpha_{0\epsilon k}^{(g)} = 6,\ \beta_{0\epsilon k}^{(g)} = 10,\ \boldsymbol{\mu}_0^{(1)} = -\mathbf{1},\ \boldsymbol{\mu}_0^{(2)} = \mathbf{1},\ \boldsymbol{\mu}_0^{(3)} = \mathbf{0},\ \boldsymbol{\Sigma}_0^{(g)} = 2.5\boldsymbol{I},$$
$$\boldsymbol{\Lambda}_{0k}^{(1)} = \boldsymbol{\Lambda}_{0k}^{(2)} = 0.2\mathbf{1},\ \boldsymbol{\Lambda}_{0k}^{(3)} = 0.4\mathbf{1},\ \text{and}\ \boldsymbol{H}_{0yk}^{(g)} = 2.5\boldsymbol{I}$$

  while the hyperparameters used in the structural equation are reset to be:

$$\boldsymbol{R}_0^{(g)} = 6\boldsymbol{I},\ \rho_0 = 4,\ \alpha_{0\delta}^{(g)} = 6,\ \beta_{0\delta}^{(g)} = 10,\ \boldsymbol{\Gamma}_0^{(1)} = \boldsymbol{\Gamma}_0^{(2)} = 0.4\mathbf{1},\ \boldsymbol{\Gamma}_0^{(3)} = 0.7\mathbf{1}\ \text{and}\ \boldsymbol{H}_{0\omega}^{(g)} = 2.5\boldsymbol{I}$$

  for $g = 1, 2, 3$, where $\boldsymbol{I}$ is an identity matrix with appropriate dimension and $\mathbf{1}$ is a vector with every element 1 and appropriate dimension.
- For the second new prior inputs, the hyperparameters used in the measurement equation are reset to be:

$$\alpha_{0\epsilon k}^{(g)} = 6,\ \beta_{0\epsilon k}^{(g)} = 10,\ \boldsymbol{\mu}_0^{(1)} = 0.2\mathbf{1},\ \boldsymbol{\mu}_0^{(2)} = 0.4\mathbf{1},\ \boldsymbol{\mu}_0^{(3)} = \mathbf{0},\ \boldsymbol{\Sigma}_0^{(g)} = 2.5\boldsymbol{I},$$
$$\boldsymbol{\Lambda}_{0k}^{(1)} = \boldsymbol{\Lambda}_{0k}^{(3)} = 0.7\mathbf{1},\ \boldsymbol{\Lambda}_{0k}^{(2)} = 0.3\mathbf{1},\ \text{and}\ \boldsymbol{H}_{0yk}^{(g)} = 2.5\boldsymbol{I}$$

  while the hyperparameters used in the structural equation are reset to be:

$$\boldsymbol{R}_0^{(g)} = 9\boldsymbol{I},\ \rho_0 = 4,\ \alpha_{0\delta}^{(g)} = 6,\ \beta_{0\delta}^{(g)} = 10,\ \boldsymbol{\Gamma}_0^{(1)} = \boldsymbol{\Gamma}_0^{(2)} = 0.5\mathbf{1},\ \boldsymbol{\Gamma}_0^{(3)} = 0.7\mathbf{1}\ \text{and}\ \boldsymbol{H}_{0\omega}^{(g)} = 2.5\boldsymbol{I}$$

  for $g = 1, 2, 3$, where $\boldsymbol{I}$ is an identity matrix with appropriate dimension and $\mathbf{1}$ is a vector with every element 1 and appropriate dimension.
- The Bayesian estimates for the two new prior inputs are not reported. Alternatively, the changes in the estimates as compared with Section 3.4 are given in Table 12 and 13.
- The changes in posterior means and standard error estimates are very small. Therefore, it is concluded that the Bayesian estimation is not sensitive to prior inputs.

3.6 Data Dependent Prior Inputs (Auxiliary Estimation)

- Since there is no useful prior information, an alternative approach is to carry out Bayesian analysis by applying data dependent prior inputs that are obtained from an initial estimation with the whole data set as in Section 3.4.[13] However, the results obtained are similar and hence not reported here.

[12] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch3
[13] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch3

4 Model Comparison or Hypothesis Testing

- In this section, for model comparison purpose, most of the Bayesian estimates are not reported for simplicity. In fact, all the results have been obtained by using R and WinBUGS code in Appendix. Only DIC is used to compare model performance.
- As emphasized repeatedly, hypothesis testing in Bayesian analysis can be done in model comparison setting.[14]

4.1 Comparison with Non-linear Multisample SEMs

- Using the same measurement equation defined in (1) or (3), the structural equation defined in (2) or (4) can be modified to give various non-linear SEMs with multisample data.
- The prior distributions, hyperparameters and initial values of parameters given in Section 3.3 are changed only slightly for the purpose of having comparable results.
- Since linear SEM is the most basic, more terms (like interaction and quadratic terms) are added to the structural equation to yield non-linear models, including but not limited to

$$\begin{aligned}
\mathrm{M}_1 &: \eta_i^{(g)} = \gamma_1^{(g)}\xi_{i1}^{(g)} + \gamma_2^{(g)}\xi_{i2}^{(g)} + \gamma_3^{(g)}\xi_{i1}^{(g)}\xi_{i2}^{(g)} + \delta_i^{(g)}, \\
\mathrm{M}_2 &: \eta_i^{(g)} = \gamma_1^{(g)}\xi_{i1}^{(g)} + \gamma_2^{(g)}\xi_{i2}^{(g)} + \gamma_3^{(g)}\xi_{i1}^{(g)2} + \gamma_4^{(g)}\xi_{i2}^{(g)2} + \delta_i^{(g)}, \\
\mathrm{M}_3 &: \eta_i^{(g)} = \gamma_1^{(g)}\xi_{i1}^{(g)} + \gamma_2^{(g)}\xi_{i2}^{(g)} + \gamma_3^{(g)}\xi_{i1}^{(g)}\xi_{i2}^{(g)} + \gamma_4^{(g)}\xi_{i1}^{(g)2} + \gamma_5^{(g)}\xi_{i2}^{(g)2} + \delta_i^{(g)},
\end{aligned}$$

  for $g = 1, 2, 3$.
- Here M_0 denotes the original linear model, M_1 is called the interaction model, M_2 is called the quadratic model and M_3 is called the second order model. The DIC values for M_0, M_1, M_2 and M_3 are 12508, 12512.6, 12484.8 and 12466.7 respectively.
- Obviously, linear model is better than the interaction model because of smaller DIC value. However, the quadratic and second order models are not used for a number of reasons.
- First of all, as mentioned in Section 3.4, the interpretation of linear model is intuitive, especially the intercept estimates are the posterior means of $\boldsymbol{y}^{(g)}$ and can serve as baseline for the group difference.
- Secondly, the DIC values for M_0, M_2 and M_3 are close enough. It means the linear model can already explain the variation in data well. By the principle of parsimony, the linear SEM with multisample data is adopted.
- Moreover, some of the estimates in the quadratic and second order models are strange. For example,

$$\begin{aligned}
\mathrm{M}_2 &: \hat{\eta}_i^{(1)} = 0.0901\xi_{i1}^{(1)} + 0.2334\xi_{i2}^{(1)} - 0.1566\xi_{i1}^{(1)2} + 0.0365^{(1)}\xi_{i2}^{(1)2}, \\
\mathrm{M}_3 &: \hat{\eta}_i^{(3)} = 0.2438\xi_{i1}^{(g)} + 0.5406\xi_{i2}^{(g)} + 0.6717\xi_{i1}^{(g)}\xi_{i2}^{(g)} - 0.3085\xi_{i1}^{(g)2} - 0.3359\xi_{i2}^{(g)2}.
\end{aligned}$$

  While the main effects are positive, the quadratic terms can be negative.
- In summary, it is suggested to use linear SEM with multisample data in this data set.

[14] "Basic and Advanced Bayesian Structural Equation Modeling" (Song and Lee) Ch6

4.2 Testing Constraints under Linear Multisample SEM

- The following nonnested competing models (or hypothese) are considered:

$$\begin{aligned} &\mathrm{M}_0 : \text{No constraints}, \quad \mathrm{M}_\mathrm{A} : \boldsymbol{\mu}^{(1)} = \boldsymbol{\mu}^{(2)} = \boldsymbol{\mu}^{(3)}, \quad \mathrm{M}_\mathrm{B} : \boldsymbol{\Lambda}^{(1)} = \boldsymbol{\Lambda}^{(2)} = \boldsymbol{\Lambda}^{(3)} \\ &\mathrm{M}_\mathrm{C} : \boldsymbol{\Gamma}^{(1)} = \boldsymbol{\Gamma}^{(2)} = \boldsymbol{\Gamma}^{(3)}, \quad \mathrm{M}_\mathrm{D} : \boldsymbol{\Phi}^{(1)} = \boldsymbol{\Phi}^{(2)} = \boldsymbol{\Phi}^{(3)}. \end{aligned}$$

- The DIC values for M_0, M_A, M_B, M_C and M_D are 12508, 12885.5, 12603.6, 12603.5 and 14018.8 respectively. It is crystal clear that the linear multisample SEM with no constraints is selected because its DIC value is far smaller than the others.

- For intuitive explanation of the rationale behind, please refer to Section 3.4.1.

5 Appendix

5.1 Figures, Tables and R/WinBUGS Code

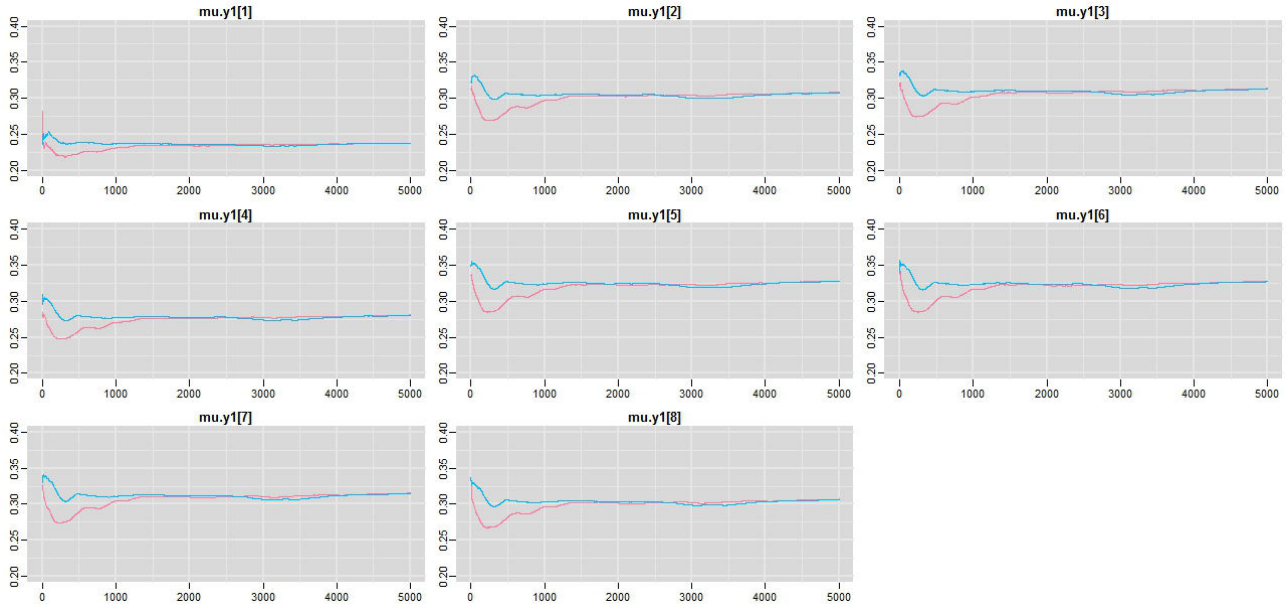

Figure 4: Running Mean Plots Corresponding to $\boldsymbol{\mu}^{(1)}$

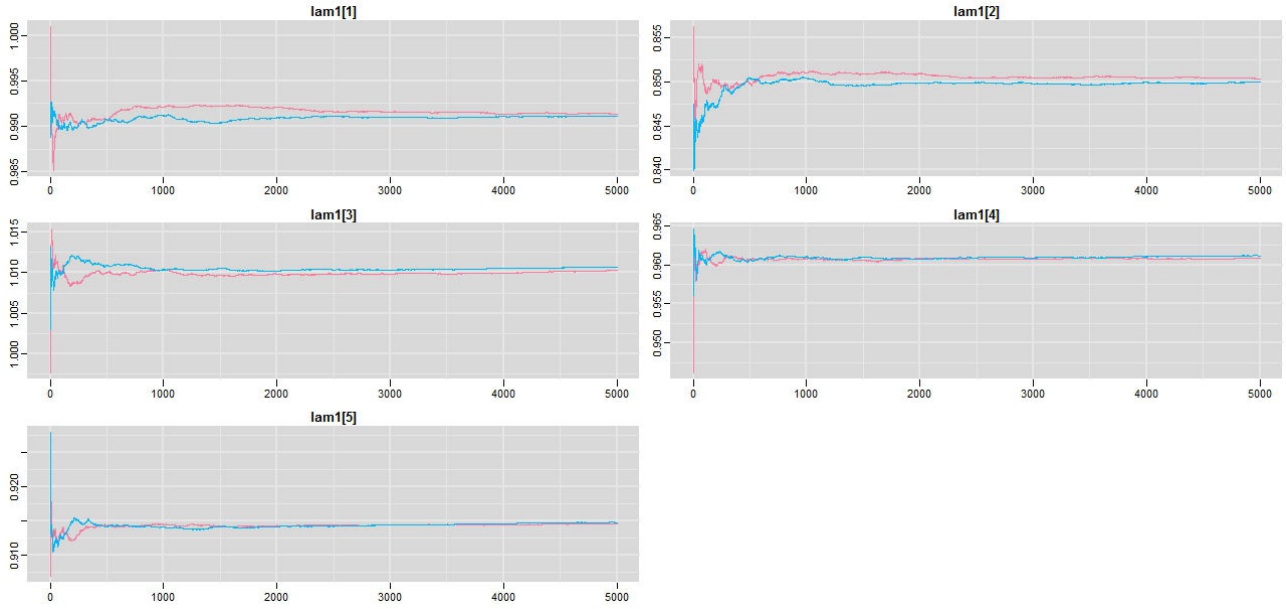

Figure 5: Running Mean Plots Corresponding to $\boldsymbol{\Lambda}^{(1)}$

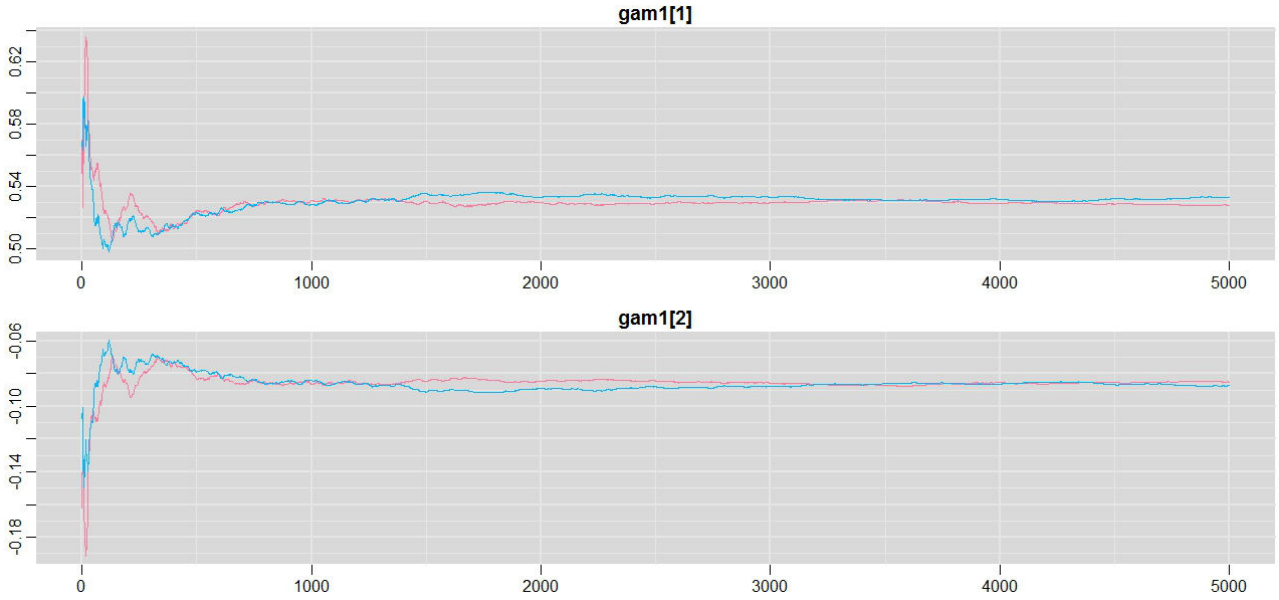

Figure 6: Running Mean Plots Corresponding to $\mathbf{\Gamma}^{(1)}$

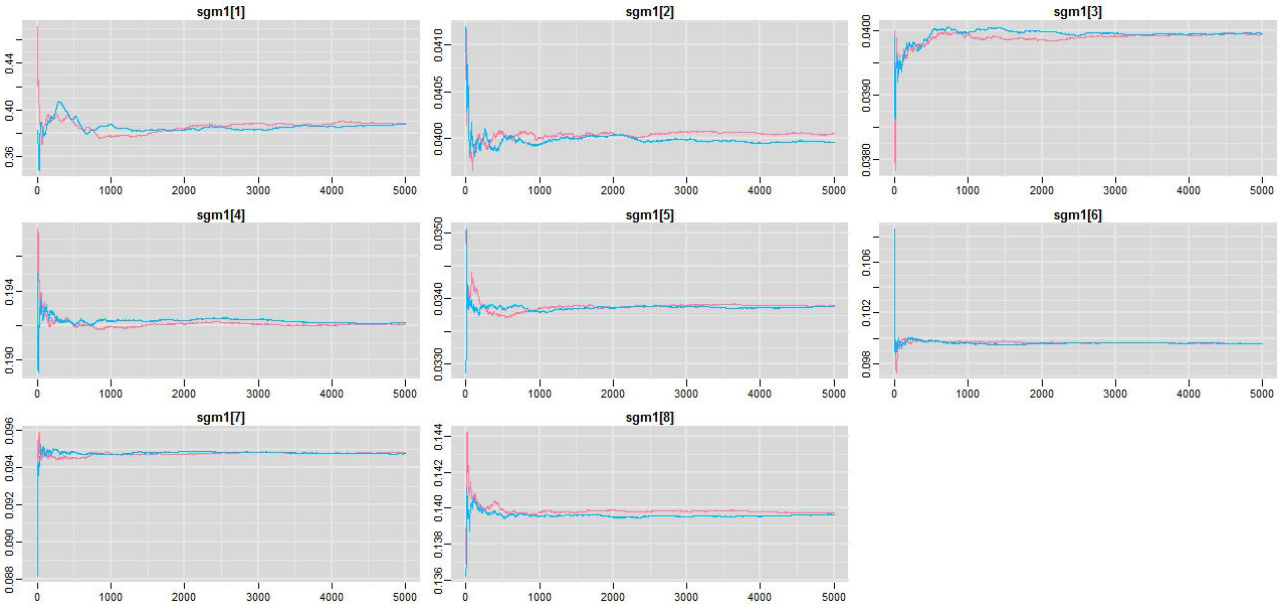

Figure 7: Running Mean Plots Corresponding to $\mathbf{\Psi}_{\boldsymbol{\epsilon}}^{(1)}$

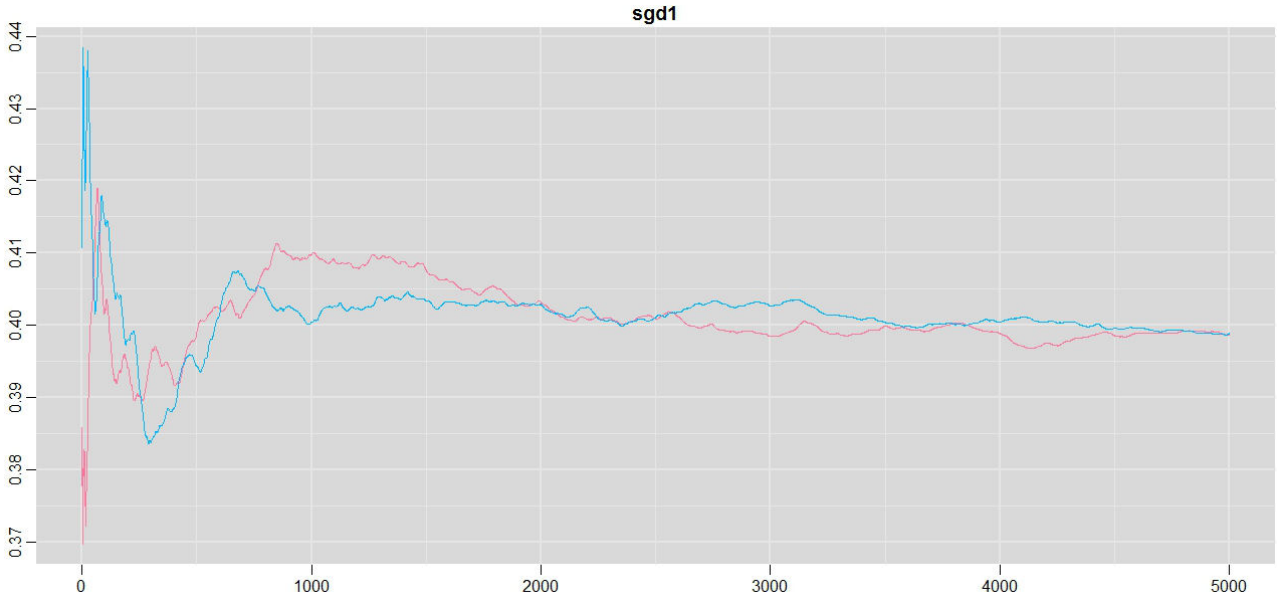

Figure 8: Running Mean Plots Corresponding to $\psi_\delta^{(1)}$

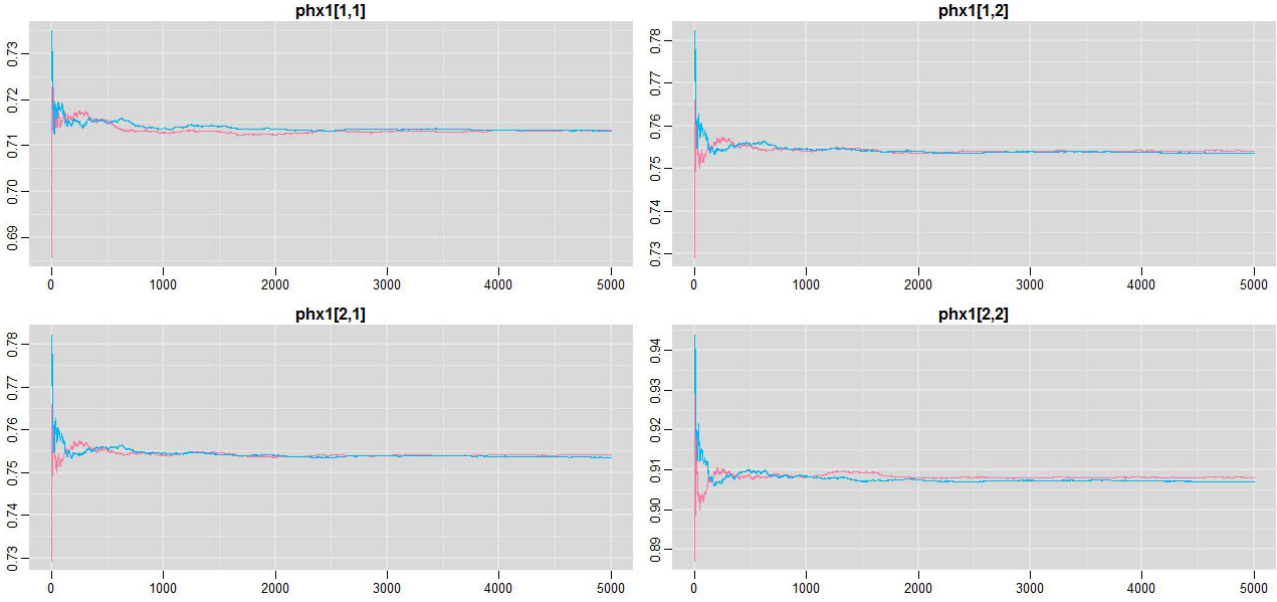

Figure 9: Running Mean Plots Corresponding to $\mathbf{\Phi}^{(1)}$

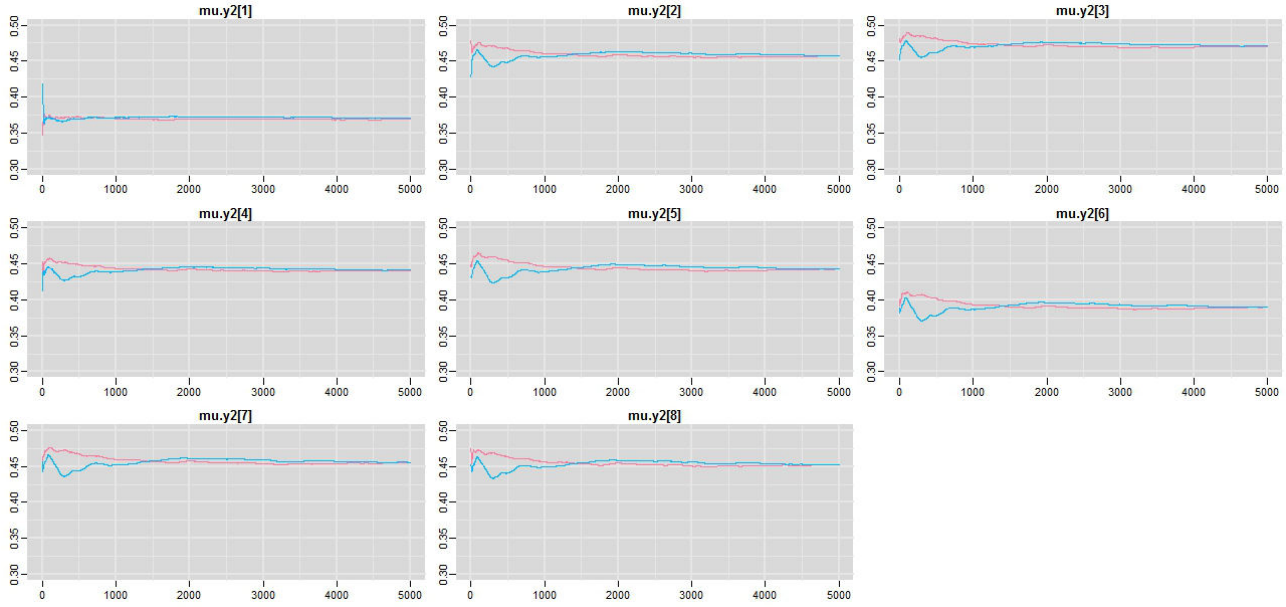

Figure 10: Running Mean Plots Corresponding to $\boldsymbol{\mu}^{(2)}$

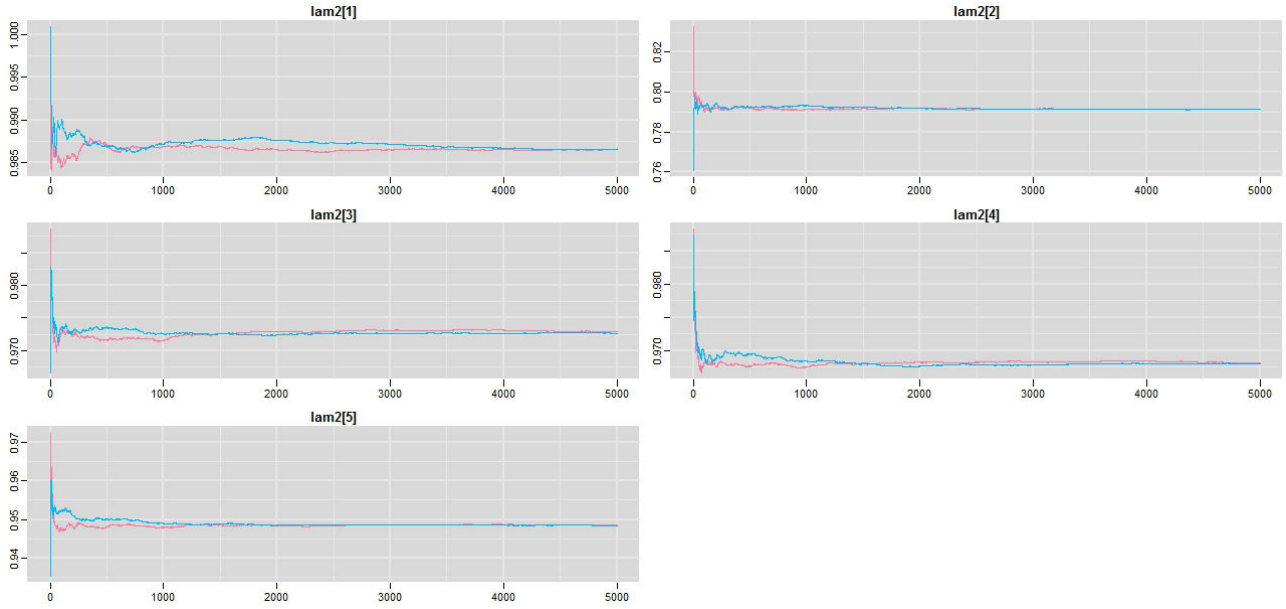

Figure 11: Running Mean Plots Corresponding to $\boldsymbol{\Lambda}^{(2)}$

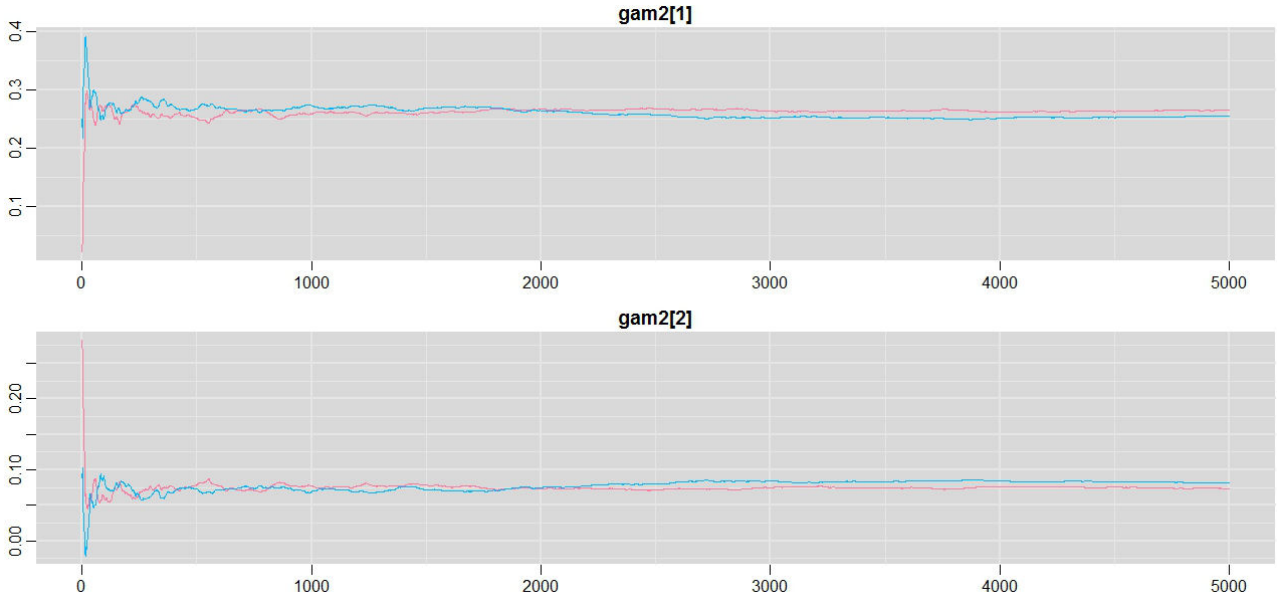

Figure 12: Running Mean Plots Corresponding to $\mathbf{\Gamma}^{(2)}$

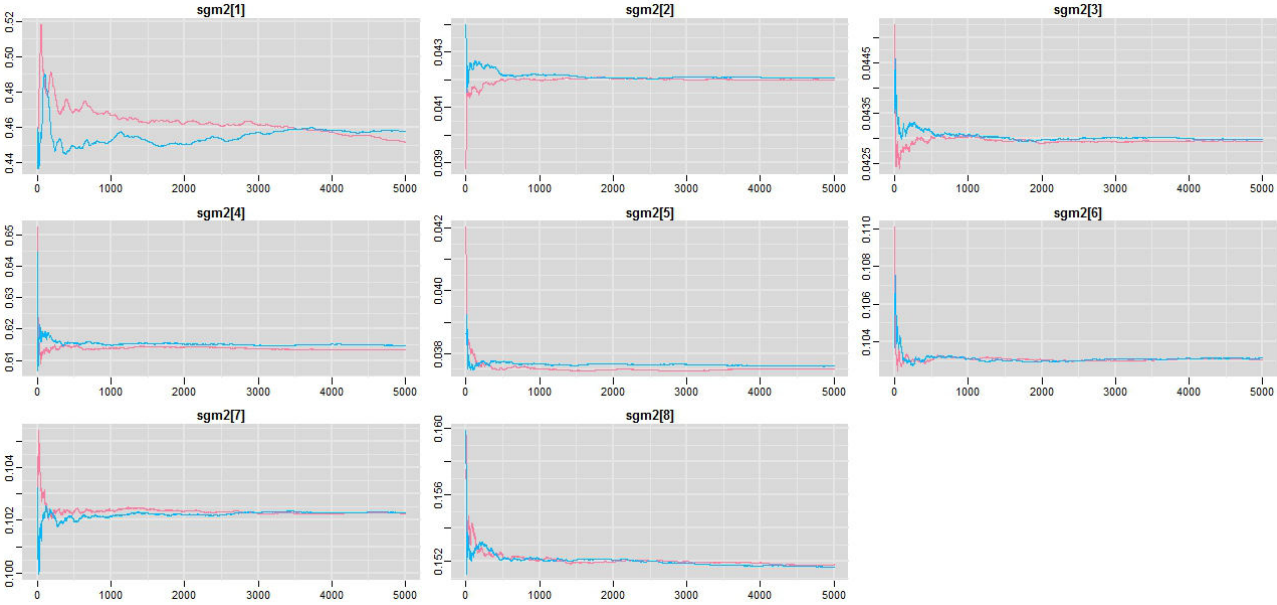

Figure 13: Running Mean Plots Corresponding to $\mathbf{\Psi}_{\boldsymbol{\epsilon}}^{(2)}$

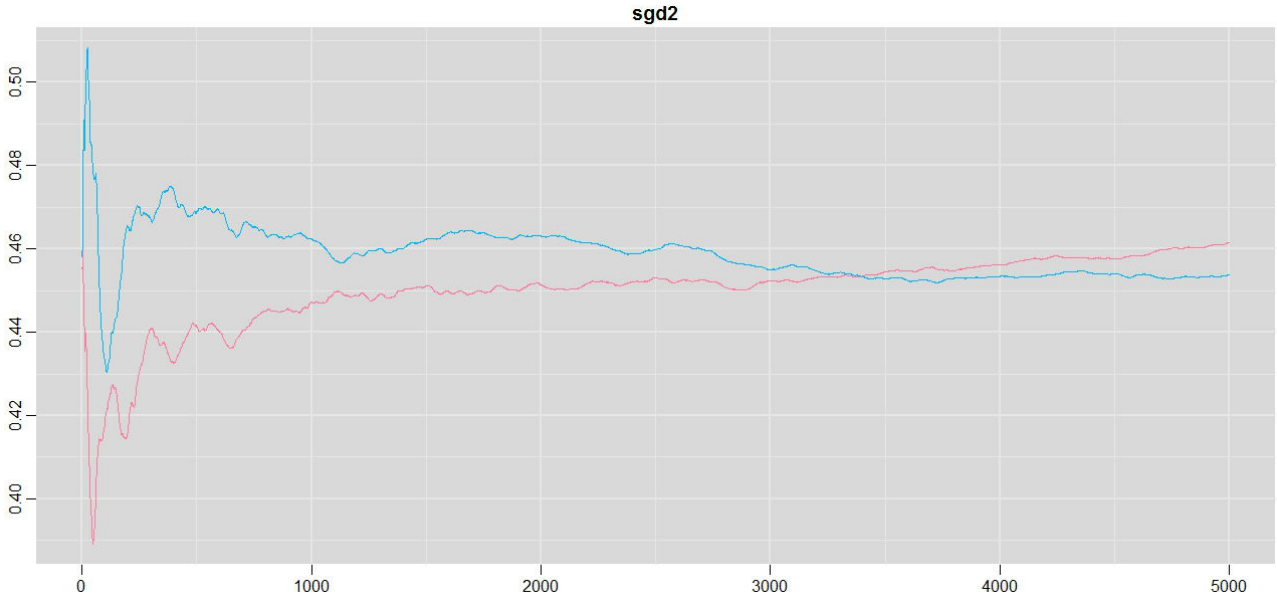

Figure 14: Running Mean Plots Corresponding to $\psi_\delta^{(2)}$

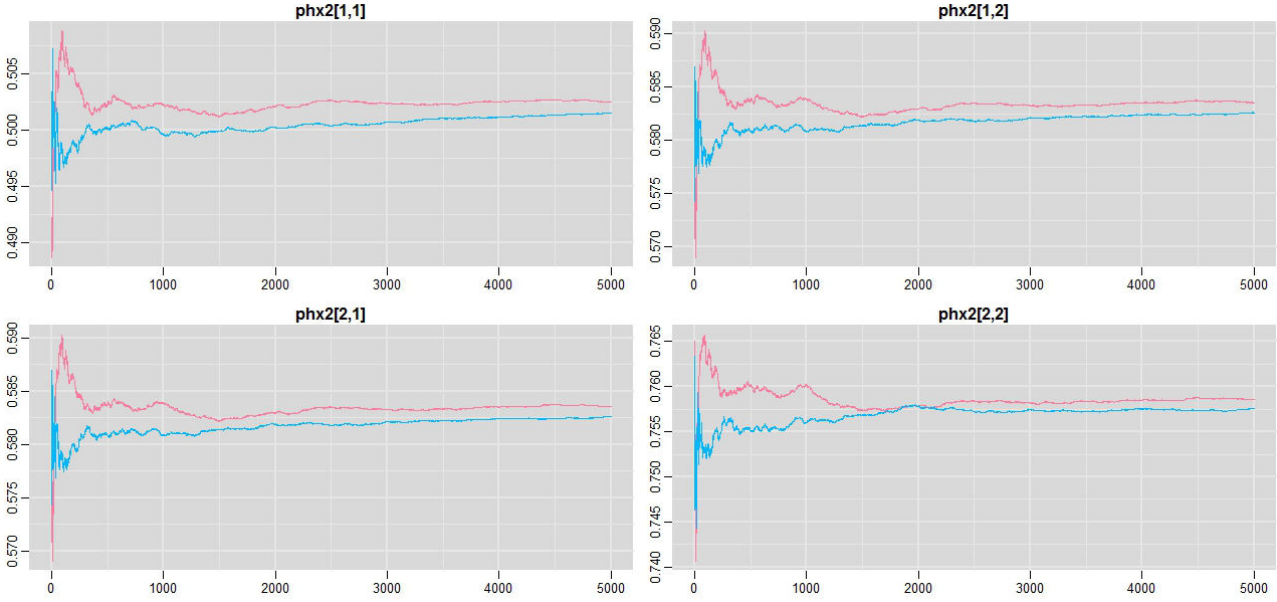

Figure 15: Running Mean Plots Corresponding to $\mathbf{\Phi}^{(2)}$

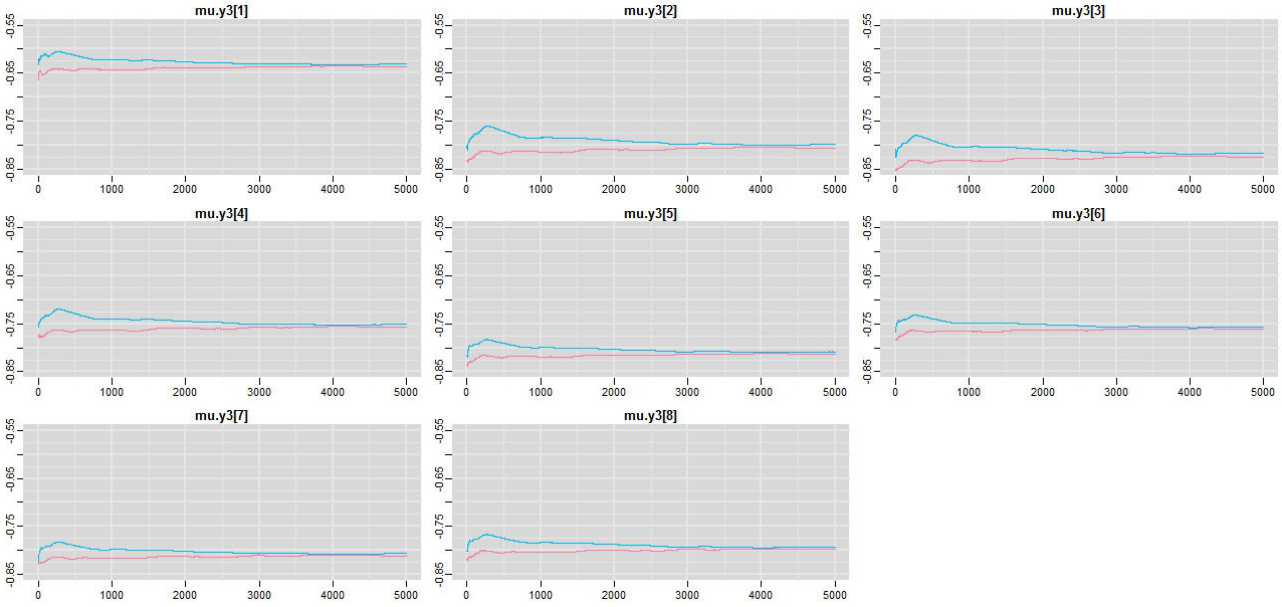

Figure 16: Running Mean Plots Corresponding to $\boldsymbol{\mu}^{(3)}$

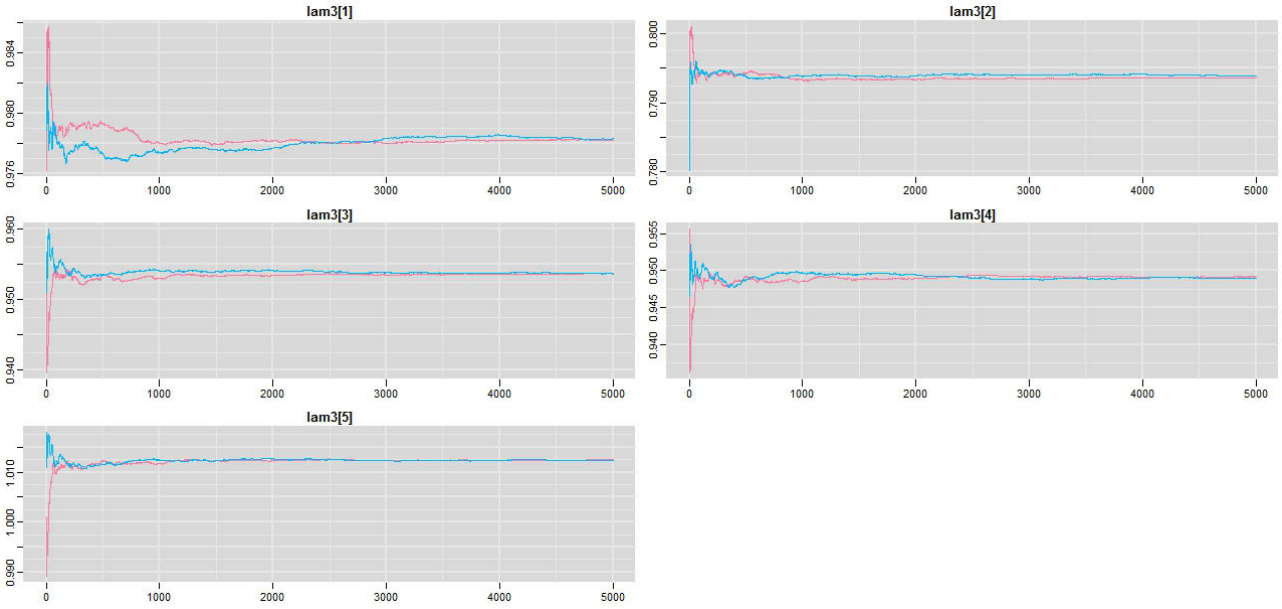

Figure 17: Running Mean Plots Corresponding to $\boldsymbol{\Lambda}^{(3)}$

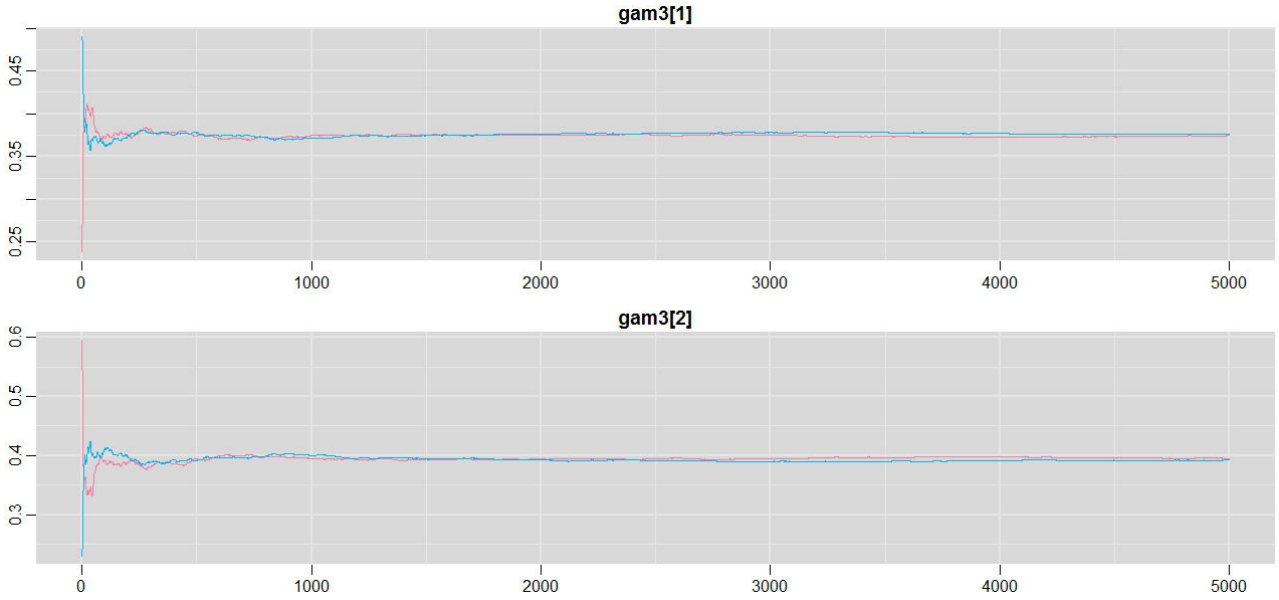

Figure 18: Running Mean Plots Corresponding to $\mathbf{\Gamma}^{(3)}$

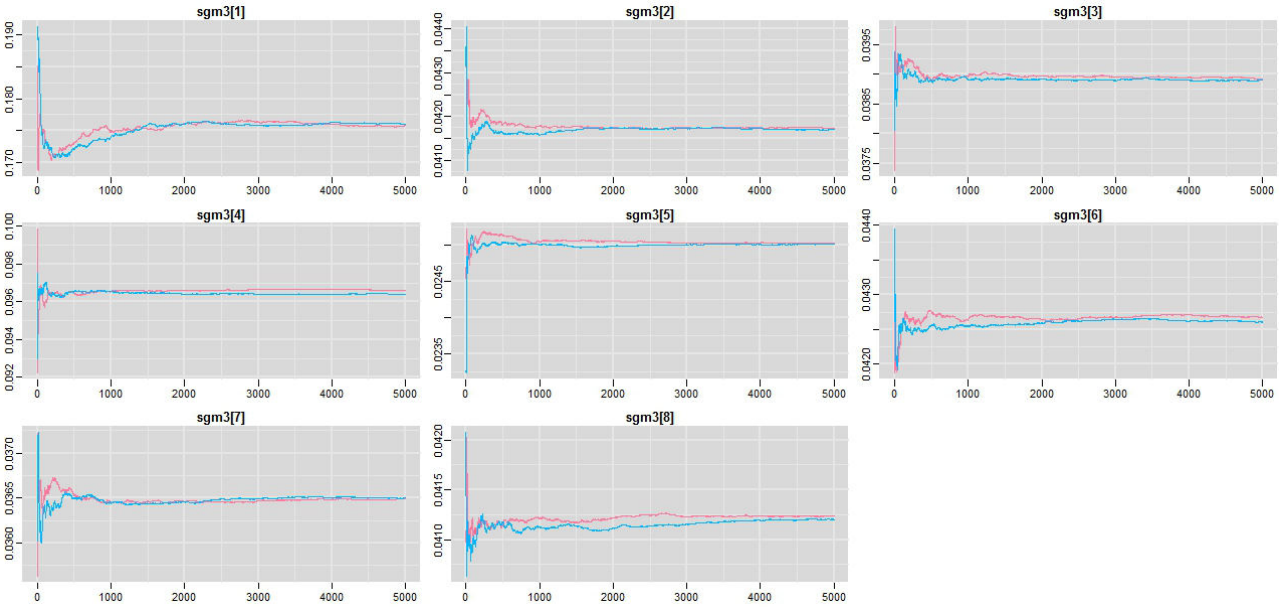

Figure 19: Running Mean Plots Corresponding to $\mathbf{\Psi}_{\boldsymbol{\epsilon}}^{(3)}$

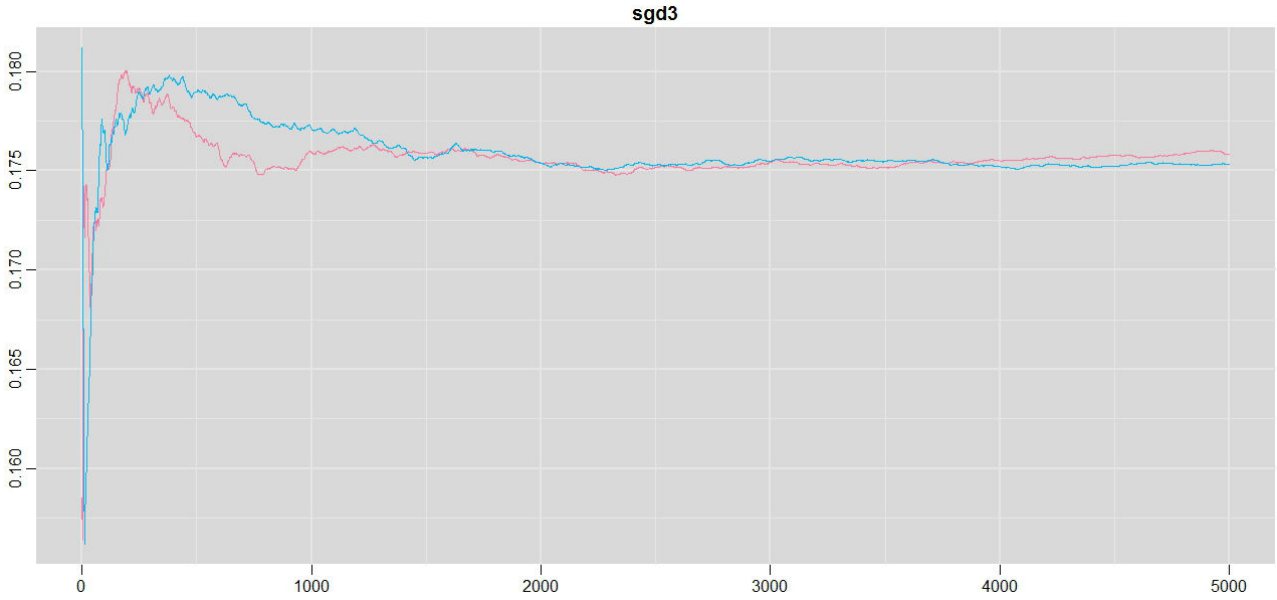

Figure 20: Running Mean Plots Corresponding to $\psi_{\delta}^{(3)}$

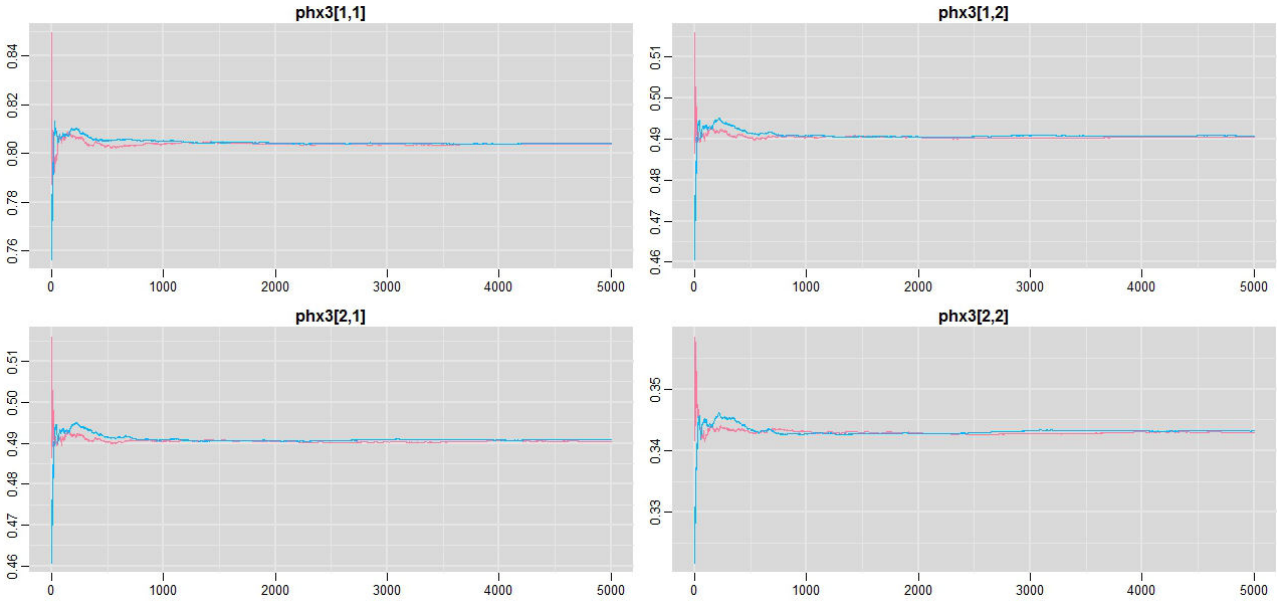

Figure 21: Running Mean Plots Corresponding to $\mathbf{\Phi}^{(3)}$

Parameter	Mean	SE	Parameter	Mean	SE
$\mu_1^{(1)}$	0.2378	0.0250	$\psi_{\epsilon 1}^{(2)}$	0.4546	0.0512
$\mu_2^{(1)}$	0.3080	0.0224	$\psi_{\epsilon 2}^{(2)}$	0.0420	0.0020
$\mu_3^{(1)}$	0.3131	0.0223	$\psi_{\epsilon 3}^{(2)}$	0.0430	0.0021
$\mu_4^{(1)}$	0.2806	0.0217	$\psi_{\epsilon 4}^{(2)}$	0.6141	0.0246
$\mu_5^{(1)}$	0.3279	0.0247	$\psi_{\epsilon 5}^{(2)}$	0.0375	0.0018
$\mu_6^{(1)}$	0.3275	0.0258	$\psi_{\epsilon 6}^{(2)}$	0.1031	0.0047
$\mu_7^{(1)}$	0.3152	0.0245	$\psi_{\epsilon 7}^{(2)}$	0.1023	0.0046
$\mu_8^{(1)}$	0.3066	0.0239	$\psi_{\epsilon 8}^{(2)}$	0.1517	0.0066
$\lambda_{32}^{(1)}$	0.9912	0.0087	$\psi_{\delta}^{(2)}$	0.4576	0.0522
$\lambda_{42}^{(1)}$	0.8502	0.0145	$\phi_{11}^{(2)}$	0.5020	0.0213
$\lambda_{63}^{(1)}$	1.0105	0.0100	$\phi_{12}^{(2)}$	0.5831	0.0245
$\lambda_{73}^{(1)}$	0.9610	0.0097	$\phi_{22}^{(2)}$	0.7581	0.0314
$\lambda_{83}^{(1)}$	0.9147	0.0110	$\mu_1^{(3)}$	-0.6337	0.0223
$\gamma_1^{(1)}$	0.5305	0.0984	$\mu_2^{(3)}$	-0.8019	0.0251
$\gamma_2^{(1)}$	-0.0863	0.0870	$\mu_3^{(3)}$	-0.8204	0.0246
$\psi_{\epsilon 1}^{(1)}$	0.3882	0.0433	$\mu_4^{(3)}$	-0.7538	0.0213
$\psi_{\epsilon 2}^{(1)}$	0.0400	0.0019	$\mu_5^{(3)}$	-0.8108	0.0169
$\psi_{\epsilon 3}^{(1)}$	0.0399	0.0019	$\mu_6^{(3)}$	-0.7588	0.0166
$\psi_{\epsilon 4}^{(1)}$	0.1921	0.0074	$\mu_7^{(3)}$	-0.8086	0.0163
$\psi_{\epsilon 5}^{(1)}$	0.0339	0.0016	$\mu_8^{(3)}$	-0.7953	0.0174
$\psi_{\epsilon 6}^{(1)}$	0.0996	0.0043	$\lambda_{32}^{(3)}$	0.9782	0.0088
$\psi_{\epsilon 7}^{(1)}$	0.0948	0.0040	$\lambda_{42}^{(3)}$	0.7938	0.0109
$\psi_{\epsilon 8}^{(1)}$	0.1397	0.0057	$\lambda_{63}^{(3)}$	0.9537	0.0119
$\psi_{\delta}^{(1)}$	0.3988	0.0437	$\lambda_{73}^{(3)}$	0.9490	0.0113
$\phi_{11}^{(1)}$	0.7132	0.0272	$\lambda_{83}^{(3)}$	1.0124	0.0122
$\phi_{12}^{(1)}$	0.7538	0.0288	$\gamma_1^{(3)}$	0.3751	0.0662
$\phi_{22}^{(1)}$	0.9074	0.0340	$\gamma_2^{(3)}$	0.3939	0.1023
$\mu_1^{(2)}$	0.3702	0.0271	$\psi_{\epsilon 1}^{(3)}$	0.1759	0.0135
$\mu_2^{(2)}$	0.4575	0.0186	$\psi_{\epsilon 2}^{(3)}$	0.0417	0.0021
$\mu_3^{(2)}$	0.4708	0.0185	$\psi_{\epsilon 3}^{(3)}$	0.0389	0.0019
$\mu_4^{(2)}$	0.4410	0.0260	$\psi_{\epsilon 4}^{(3)}$	0.0965	0.0040
$\mu_5^{(2)}$	0.4426	0.0225	$\psi_{\epsilon 5}^{(3)}$	0.0250	0.0011
$\mu_6^{(2)}$	0.3897	0.0232	$\psi_{\epsilon 6}^{(3)}$	0.0426	0.0019
$\mu_7^{(2)}$	0.4556	0.0230	$\psi_{\epsilon 7}^{(3)}$	0.0365	0.0016
$\mu_8^{(2)}$	0.4525	0.0234	$\psi_{\epsilon 8}^{(3)}$	0.0412	0.0018
$\lambda_{32}^{(2)}$	0.9866	0.0112	$\psi_{\delta}^{(3)}$	0.1756	0.0136
$\lambda_{42}^{(2)}$	0.7914	0.0323	$\phi_{11}^{(3)}$	0.8040	0.0324
$\lambda_{63}^{(2)}$	0.9728	0.0122	$\phi_{12}^{(3)}$	0.4906	0.0200
$\lambda_{73}^{(2)}$	0.9681	0.0118	$\phi_{22}^{(3)}$	0.3431	0.0140
$\lambda_{83}^{(2)}$	0.9485	0.0137			
$\gamma_1^{(2)}$	0.2607	0.1717	DIC:12508		
$\gamma_2^{(2)}$	0.0773	0.1389			

Table 11: Bayesian Estimates of Parameters in Linear SEM with Multisample Data

Parameter	ΔMean	ΔSE	Parameter	ΔMean	ΔSE
$\mu_1^{(1)}$	-0.00080	-0.00001	$\psi_{\epsilon 1}^{(2)}$	-0.00003	-0.00001
$\mu_2^{(1)}$	-0.00138	-0.00005	$\psi_{\epsilon 2}^{(2)}$	0.00022	0.00001
$\mu_3^{(1)}$	-0.00137	-0.00003	$\psi_{\epsilon 3}^{(2)}$	0.00031	0.00002
$\mu_4^{(1)}$	-0.00122	-0.00003	$\psi_{\epsilon 4}^{(2)}$	-0.00057	-0.00003
$\mu_5^{(1)}$	-0.00157	-0.00003	$\psi_{\epsilon 5}^{(2)}$	0.00013	0.00001
$\mu_6^{(1)}$	-0.00160	-0.00002	$\psi_{\epsilon 6}^{(2)}$	0.00023	0.00000
$\mu_7^{(1)}$	-0.00152	-0.00002	$\psi_{\epsilon 7}^{(2)}$	0.00017	0.00000
$\mu_8^{(1)}$	-0.00146	-0.00002	$\psi_{\epsilon 8}^{(2)}$	-0.00019	-0.00002
$\lambda_{32}^{(1)}$	0.00064	0.00004	$\psi_{\delta}^{(2)}$	-0.00003	-0.00001
$\lambda_{42}^{(1)}$	0.00098	0.00001	$\phi_{11}^{(2)}$	-0.00335	-0.00012
$\lambda_{63}^{(1)}$	0.00028	0.00002	$\phi_{12}^{(2)}$	0.00008	-0.00006
$\lambda_{73}^{(1)}$	0.00034	0.00001	$\phi_{22}^{(2)}$	-0.00260	-0.00009
$\lambda_{83}^{(1)}$	0.00045	0.00000	$\mu_1^{(3)}$	0.00000	-0.00001
$\gamma_1^{(1)}$	0.00515	0.00144	$\mu_2^{(3)}$	-0.00001	-0.00003
$\gamma_2^{(1)}$	-0.00433	0.00122	$\mu_3^{(3)}$	-0.00001	-0.00001
$\psi_{\epsilon 1}^{(1)}$	-0.00005	0.00000	$\mu_4^{(3)}$	-0.00001	-0.00001
$\psi_{\epsilon 2}^{(1)}$	0.00019	0.00001	$\mu_5^{(3)}$	0.00001	-0.00008
$\psi_{\epsilon 3}^{(1)}$	0.00025	0.00002	$\mu_6^{(3)}$	0.00001	-0.00004
$\psi_{\epsilon 4}^{(1)}$	-0.00038	-0.00001	$\mu_7^{(3)}$	0.00001	-0.00004
$\psi_{\epsilon 5}^{(1)}$	0.00010	0.00001	$\mu_8^{(3)}$	0.00001	-0.00005
$\psi_{\epsilon 6}^{(1)}$	0.00022	0.00001	$\lambda_{32}^{(3)}$	0.00057	0.00003
$\psi_{\epsilon 7}^{(1)}$	0.00008	0.00000	$\lambda_{42}^{(3)}$	0.00072	0.00001
$\psi_{\epsilon 8}^{(1)}$	-0.00015	-0.00001	$\lambda_{63}^{(3)}$	0.00171	0.00005
$\psi_{\delta}^{(1)}$	-0.00005	-0.00001	$\lambda_{73}^{(3)}$	0.00166	0.00005
$\phi_{11}^{(1)}$	-0.00248	-0.00008	$\lambda_{83}^{(3)}$	0.00198	0.00004
$\phi_{12}^{(1)}$	0.00007	-0.00004	$\gamma_1^{(3)}$	0.00119	0.00150
$\phi_{22}^{(1)}$	-0.00205	-0.00007	$\gamma_2^{(3)}$	-0.00104	0.00244
$\mu_1^{(2)}$	0.00070	0.00000	$\psi_{\epsilon 1}^{(3)}$	-0.00003	0.00000
$\mu_2^{(2)}$	0.00120	-0.00005	$\psi_{\epsilon 2}^{(3)}$	0.00023	0.00001
$\mu_3^{(2)}$	0.00118	-0.00002	$\psi_{\epsilon 3}^{(3)}$	0.00018	0.00001
$\mu_4^{(2)}$	0.00112	-0.00001	$\psi_{\epsilon 4}^{(3)}$	-0.00022	-0.00001
$\mu_5^{(2)}$	0.00147	-0.00003	$\psi_{\epsilon 5}^{(3)}$	0.00006	0.00000
$\mu_6^{(2)}$	0.00146	-0.00002	$\psi_{\epsilon 6}^{(3)}$	0.00008	0.00000
$\mu_7^{(2)}$	0.00145	-0.00002	$\psi_{\epsilon 7}^{(3)}$	0.00010	0.00000
$\mu_8^{(2)}$	0.00143	-0.00002	$\psi_{\epsilon 8}^{(3)}$	-0.00003	0.00000
$\lambda_{32}^{(2)}$	0.00148	0.00007	$\psi_{\delta}^{(3)}$	-0.00003	0.00000
$\lambda_{42}^{(2)}$	0.00196	0.00005	$\phi_{11}^{(3)}$	-0.00262	-0.00009
$\lambda_{63}^{(2)}$	0.00051	0.00003	$\phi_{12}^{(3)}$	-0.00028	-0.00006
$\lambda_{73}^{(2)}$	0.00054	0.00002	$\phi_{22}^{(3)}$	-0.00279	-0.00010
$\lambda_{83}^{(2)}$	0.00077	0.00001			
$\gamma_1^{(2)}$	0.00404	0.00696			
$\gamma_2^{(2)}$	-0.00307	0.00547			

Table 12: Changes in Bayesian Estimates of Parameters for the First New Prior Inputs

Parameter	ΔMean	ΔSE	Parameter	ΔMean	ΔSE
$\mu_1^{(1)}$	0.00016	0.00000	$\psi_{\epsilon 1}^{(2)}$	0.00001	0.00000
$\mu_2^{(1)}$	0.00027	0.00002	$\psi_{\epsilon 2}^{(2)}$	-0.00008	0.00000
$\mu_3^{(1)}$	0.00027	0.00002	$\psi_{\epsilon 3}^{(2)}$	-0.00012	-0.00001
$\mu_4^{(1)}$	0.00024	0.00001	$\psi_{\epsilon 4}^{(2)}$	0.00026	0.00001
$\mu_5^{(1)}$	0.00031	0.00001	$\psi_{\epsilon 5}^{(2)}$	-0.00005	0.00000
$\mu_6^{(1)}$	0.00032	0.00001	$\psi_{\epsilon 6}^{(2)}$	-0.00009	0.00000
$\mu_7^{(1)}$	0.00030	0.00001	$\psi_{\epsilon 7}^{(2)}$	-0.00007	0.00000
$\mu_8^{(1)}$	0.00029	0.00001	$\psi_{\epsilon 8}^{(2)}$	0.00009	0.00001
$\lambda_{32}^{(1)}$	-0.00032	-0.00002	$\psi_{\delta}^{(2)}$	0.00001	0.00000
$\lambda_{42}^{(1)}$	-0.00049	-0.00001	$\phi_{11}^{(2)}$	0.00160	0.00006
$\lambda_{63}^{(1)}$	-0.00015	0.00000	$\phi_{12}^{(2)}$	0.00001	0.00003
$\lambda_{73}^{(1)}$	-0.00018	-0.00001	$\phi_{22}^{(2)}$	0.00129	0.00005
$\lambda_{83}^{(1)}$	-0.00023	0.00000	$\mu_1^{(3)}$	0.00000	0.00001
$\gamma_1^{(1)}$	-0.00241	-0.00064	$\mu_2^{(3)}$	0.00000	0.00001
$\gamma_2^{(1)}$	0.00197	-0.00055	$\mu_3^{(3)}$	0.00000	0.00001
$\psi_{\epsilon 1}^{(1)}$	0.00002	0.00000	$\mu_4^{(3)}$	0.00000	0.00000
$\psi_{\epsilon 2}^{(1)}$	-0.00008	-0.00001	$\mu_5^{(3)}$	0.00000	0.00004
$\psi_{\epsilon 3}^{(1)}$	-0.00011	-0.00001	$\mu_6^{(3)}$	0.00000	0.00002
$\psi_{\epsilon 4}^{(1)}$	0.00017	0.00001	$\mu_7^{(3)}$	0.00000	0.00002
$\psi_{\epsilon 5}^{(1)}$	-0.00004	0.00000	$\mu_8^{(3)}$	0.00000	0.00002
$\psi_{\epsilon 6}^{(1)}$	-0.00010	0.00000	$\lambda_{32}^{(3)}$	-0.00027	-0.00002
$\psi_{\epsilon 7}^{(1)}$	-0.00004	0.00000	$\lambda_{42}^{(3)}$	-0.00034	0.00000
$\psi_{\epsilon 8}^{(1)}$	0.00007	0.00000	$\lambda_{63}^{(3)}$	-0.00080	-0.00002
$\psi_{\delta}^{(1)}$	0.00003	0.00000	$\lambda_{73}^{(3)}$	-0.00077	-0.00002
$\phi_{11}^{(1)}$	0.00122	0.00004	$\lambda_{83}^{(3)}$	-0.00092	-0.00002
$\phi_{12}^{(1)}$	-0.00001	0.00002	$\gamma_1^{(3)}$	-0.00061	-0.00063
$\phi_{22}^{(1)}$	0.00102	0.00004	$\gamma_2^{(3)}$	0.00062	-0.00103
$\mu_1^{(2)}$	0.00028	0.00000	$\psi_{\epsilon 1}^{(3)}$	0.00001	0.00000
$\mu_2^{(2)}$	0.00047	0.00002	$\psi_{\epsilon 2}^{(3)}$	-0.00010	-0.00001
$\mu_3^{(2)}$	0.00046	0.00001	$\psi_{\epsilon 3}^{(3)}$	-0.00007	0.00000
$\mu_4^{(2)}$	0.00044	0.00000	$\psi_{\epsilon 4}^{(3)}$	0.00010	0.00000
$\mu_5^{(2)}$	0.00059	0.00001	$\psi_{\epsilon 5}^{(3)}$	-0.00002	0.00000
$\mu_6^{(2)}$	0.00058	0.00001	$\psi_{\epsilon 6}^{(3)}$	-0.00003	0.00000
$\mu_7^{(2)}$	0.00058	0.00001	$\psi_{\epsilon 7}^{(3)}$	-0.00004	0.00000
$\mu_8^{(2)}$	0.00057	0.00001	$\psi_{\epsilon 8}^{(3)}$	0.00002	0.00000
$\lambda_{32}^{(2)}$	-0.00072	-0.00002	$\psi_{\delta}^{(3)}$	0.00001	0.00000
$\lambda_{42}^{(2)}$	-0.00118	-0.00002	$\phi_{11}^{(3)}$	0.00126	0.00004
$\lambda_{63}^{(2)}$	-0.00030	-0.00001	$\phi_{12}^{(3)}$	0.00014	0.00003
$\lambda_{73}^{(2)}$	-0.00031	-0.00001	$\phi_{22}^{(3)}$	0.00134	0.00005
$\lambda_{83}^{(2)}$	-0.00042	0.00000			
$\gamma_1^{(2)}$	-0.00209	-0.00281			
$\gamma_2^{(2)}$	0.00151	-0.00221			

Table 13: Changes in Bayesian Estimates of Parameters for the Second New Prior Inputs

```
#R Code with the use of R2WinBUGS

setwd("C:/Users/Benjamin Chan/Desktop/CUHK/Stat/5020/Project")
#Load packages
library(mvtnorm)               #Load mvtnorm package
library(R2WinBUGS)             #Load R2WinBUGS package
library(mcmcplots)             #Load mcmcplots package
library(plyr)                  #Load plyr package

#Data Processing
#The data found on http://archive.ics.uci.edu/ml/machine-learning-
databases/abalone/abalone.data
#are saved in "Abalone.txt" first.
X<-read.table("Abalone.txt",sep=",")    #Read in data
colnames(X)<-
c("Sex","Length","Diameter","Height","Whole.weight","Shucked.weight","Vis
cera.weight","Shell.weight","Rings")
head(X)

D<-X[,c(1,9,2:8)]              #Rearrange columns
head(D)                        #Display some data
#Output the data (with column rearrangement) in csv
write.csv(D,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/Abalone.csv",row.names=FALSE)

(Freqcount_Rings<-count(D, 'Rings'))
write.csv(Freqcount_Rings,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/Freqcount_Rings.csv")
hist(D$Rings,main="Histogram of
Rings",xlab="Rings",breaks=seq(1,29,by=1))

(N<-nrow(D))                   #Total sample size

D1<-D[D$Sex=="M",2:9]          #Select sex M
(N1<-nrow(D1))                 #Number of observations for sex M
D2<-D[D$Sex=="F",2:9]          #Select sex F
(N2<-nrow(D2))                 #Number of observations for sex F
D3<-D[D$Sex=="I",2:9]          #Select sex I
(N3<-nrow(D3))                 #Number of observations for sex I

Dall<-D[,2:9]                  #Save all data, only remove sex attribute
(original data order)
Dall<-as.matrix(Dall)

(Cor<-cor(Dall))               #Correlation matrix for all sexes
write.csv(Cor,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/Correlation_matrix.csv")

(p<-ncol(Dall))                #Number of observed variables (sex not
included)
(Mean<-apply(Dall,2,mean))     #Compute column mean for all sexes
(Sd<-apply(Dall,2,sd))         #Compute column sd for all sexes

(mean_M<-apply(D1,2,mean))     #Compute unstandardized mean vector for sex
M
(mean_F<-apply(D2,2,mean))     #Compute unstandardized mean vector for sex
F
(mean_I<-apply(D3,2,mean))     #Compute unstandardized mean vector for sex
I
```

```
mean_Overall<-Mean            #Compute unstandardized mean vector for all
sexes

mean_compare<-rbind(mean_M,mean_F,mean_I,mean_Overall)
write.csv(mean_compare,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/mean_compare.csv")

Vall<-rbind(D1,D2,D3)         #Initialize matrices to store standardized
observations (in sex order: M,F,I)
Vall<-as.matrix(Vall)

for (i in 1:N) {
   for (j in 1:p){
      Vall[i,j]=(Vall[i,j]-Mean[j])/Sd[j]
   }
}

head(Vall)                        #Display some standardized observations

V1<-Vall[1:N1,]                   #Save standardized observations for sex
M
V2<-Vall[(N1+1):(N1+N2),]         #Save standardized observations for sex
F
V3<-Vall[(N1+N2+1):(N1+N2+N3),]   #Save standardized observations for sex
I

V1mean<-apply(V1,2,mean)          #Standardized mean vector for each sex
V2mean<-apply(V2,2,mean)
V3mean<-apply(V3,2,mean)
(mean_compare_standard<-rbind(V1mean,V2mean,V3mean))
write.csv(mean_compare_standard,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/mean_compare_standard.csv")

(V1sd<-apply(V1,2,sd))            #Standard deviation vector for each sex
(V2sd<-apply(V2,2,sd))
(V3sd<-apply(V3,2,sd))

#Linear multisample model estimation (also act as auxiliary Bayesian
estimation)
parameters=c("mu.y1", "lam1", "gam1", "sgm1", "sgd1", "phx1",
             "mu.y2", "lam2", "gam2", "sgm2", "sgd2", "phx2",
             "mu.y3", "lam3", "gam3", "sgm3", "sgd3", "phx3")

init1_lin=list(mu.y1=rep(0,8), lam1=rep(0,5), gam1=rep(0,2),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y2=rep(0,8), lam2=rep(0,5), gam2=rep(0,2),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y3=rep(0,8), lam3=rep(0,5), gam3=rep(0,2),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_lin=list(mu.y1=c(0,rep(1,7)), lam1=rep(1,5), gam1=rep(1,2),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y2=c(0,rep(1,7)), lam2=rep(1,5), gam2=rep(1,2),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y3=c(0,rep(1,7)), lam3=rep(1,5), gam3=rep(1,2),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_lin=list(init1_lin, init2_lin)

R=matrix(c(8, 0, 0, 8), nrow=2)
```

```
data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_linear<-bugs(data, inits_lin, parameters,
                   model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/linearmodel.txt",
                   n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                   working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_linear<-model_linear$summary
write.csv(defaultsummary_linear,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_linear.csv")

#Convergence plots (linear model)
rmeanplot(model_linear,parms="mu.y1",ylim=c(0.2,0.4))
rmeanplot(model_linear,parms="lam1")
rmeanplot(model_linear,parms="gam1")
rmeanplot(model_linear,parms="sgm1")
rmeanplot(model_linear,parms="sgd1")
rmeanplot(model_linear,parms="phx1")

rmeanplot(model_linear,parms="mu.y2",ylim=c(0.3,0.5))
rmeanplot(model_linear,parms="lam2")
rmeanplot(model_linear,parms="gam2")
rmeanplot(model_linear,parms="sgm2")
rmeanplot(model_linear,parms="sgd2")
rmeanplot(model_linear,parms="phx2")

rmeanplot(model_linear,parms="mu.y3",ylim=c(-0.85,-0.55))
rmeanplot(model_linear,parms="lam3")
rmeanplot(model_linear,parms="gam3")
rmeanplot(model_linear,parms="sgm3")
rmeanplot(model_linear,parms="sgd3")
rmeanplot(model_linear,parms="phx3")

#Sensitivity analysis without the aid of auxiliary Bayesian estimation
# New prior 1
parameters=c("mu.y1", "lam1", "gam1", "sgm1", "sgd1", "phx1",
             "mu.y2", "lam2", "gam2", "sgm2", "sgd2", "phx2",
             "mu.y3", "lam3", "gam3", "sgm3", "sgd3", "phx3")

init1_lin=list(mu.y1=rep(0,8), lam1=rep(0,5), gam1=rep(0,2),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y2=rep(0,8), lam2=rep(0,5), gam2=rep(0,2),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y3=rep(0,8), lam3=rep(0,5), gam3=rep(0,2),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_lin=list(mu.y1=c(0,rep(1,7)), lam1=rep(1,5), gam1=rep(1,2),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y2=c(0,rep(1,7)), lam2=rep(1,5), gam2=rep(1,2),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y3=c(0,rep(1,7)), lam3=rep(1,5), gam3=rep(1,2),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_lin=list(init1_lin, init2_lin)

R=matrix(c(6, 0, 0, 6), nrow=2)
```

```
data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_newprior1<-bugs(data, inits_lin, parameters,
                      model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/newprior1model.txt",
                      n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                      working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_newprior1<-model_newprior1$summary
write.csv(defaultsummary_newprior1,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_newprior1.csv")

# New prior 2
parameters=c("mu.y1", "lam1", "gam1", "sgm1", "sgd1", "phx1",
             "mu.y2", "lam2", "gam2", "sgm2", "sgd2", "phx2",
             "mu.y3", "lam3", "gam3", "sgm3", "sgd3", "phx3")

init1_lin=list(mu.y1=rep(0,8), lam1=rep(0,5), gam1=rep(0,2),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y2=rep(0,8), lam2=rep(0,5), gam2=rep(0,2),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y3=rep(0,8), lam3=rep(0,5), gam3=rep(0,2),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_lin=list(mu.y1=c(0,rep(1,7)), lam1=rep(1,5), gam1=rep(1,2),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y2=c(0,rep(1,7)), lam2=rep(1,5), gam2=rep(1,2),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y3=c(0,rep(1,7)), lam3=rep(1,5), gam3=rep(1,2),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_lin=list(init1_lin, init2_lin)

R=matrix(c(9, 0, 0, 9), nrow=2)

data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_newprior2<-bugs(data, inits_lin, parameters,
                      model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/newprior2model.txt",
                      n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                      working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_newprior2<-model_newprior2$summary
write.csv(defaultsummary_newprior2,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_newprior2.csv")

# Change in mean and sd for new prior 1
change_mean_newprior1<-defaultsummary_newprior1[,1]-
defaultsummary_linear[,1]
change_sd_newprior1<-defaultsummary_newprior1[,2]-
defaultsummary_linear[,2]

# Change in mean and sd for new prior 2
```

```
change_mean_newprior2<-defaultsummary_newprior2[,1]-
defaultsummary_linear[,1]
change_sd_newprior2<-defaultsummary_newprior2[,2]-
defaultsummary_linear[,2]

#Summary of Sensitivity analysis
change_mean_sd_newpriors<-
cbind(change_mean_newprior1,change_sd_newprior1,change_mean_newprior2,cha
nge_sd_newprior2)
head(change_mean_sd_newpriors)           #Display some results
write.csv(change_mean_sd_newpriors,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/change_mean_sd_newpriors.csv")


#Linear multisample model estimation with the aid of auxiliary Bayesian
estimation (previous results)
#Hyperparameters are set by auxiliary Bayesian estimation
Ephx1<-(4-2-1)*model_linear$mean$phx1; R1<-solve(Ephx1)
Ephx2<-(4-2-1)*model_linear$mean$phx2; R2<-solve(Ephx2)
Ephx3<-(4-2-1)*model_linear$mean$phx3; R3<-solve(Ephx3)

(mu1prior=as.vector(model_linear$mean$mu.y1))
(mu2prior=as.vector(model_linear$mean$mu.y2))
(mu3prior=as.vector(model_linear$mean$mu.y3))
(lam1prior=as.vector(model_linear$mean$lam1))
(lam2prior=as.vector(model_linear$mean$lam2))
(lam3prior=as.vector(model_linear$mean$lam3))
(gam1prior=as.vector(model_linear$mean$gam1))
(gam2prior=as.vector(model_linear$mean$gam2))
(gam3prior=as.vector(model_linear$mean$gam3))

parameters=c("mu.y1", "lam1", "gam1", "sgm1", "sgd1", "phx1",
             "mu.y2", "lam2", "gam2", "sgm2", "sgd2", "phx2",
             "mu.y3", "lam3", "gam3", "sgm3", "sgd3", "phx3")

init1_lin=list(mu.y1=rep(0,8), lam1=rep(0,5), gam1=rep(0,2),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y2=rep(0,8), lam2=rep(0,5), gam2=rep(0,2),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y3=rep(0,8), lam3=rep(0,5), gam3=rep(0,2),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_lin=list(mu.y1=c(0,rep(1,7)), lam1=rep(1,5), gam1=rep(1,2),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y2=c(0,rep(1,7)), lam2=rep(1,5), gam2=rep(1,2),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y3=c(0,rep(1,7)), lam3=rep(1,5), gam3=rep(1,2),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_lin=list(init1_lin, init2_lin)

data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R1=R1, R2=R2, R3=R3, y1=V1,
y2=V2, y3=V3)

model_lin_withaux<-bugs(data, inits_lin, parameters,
                        model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/linearwithauxmodel.txt",
                        n.chains=2, n.iter=10000, n.burnin=5000,
n.thin=1, bugs.directory="C:/Program Files/WinBUGS14/",
```

```
                        working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_lin_withaux<-model_lin_withaux$summary
write.csv(defaultsummary_lin_withaux,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_lin_withaux.csv")

# Compare results with and without auxiliary Bayesian estimation
summary_linear<-c(defaultsummary_linear[,1],DIC=model_linear$DIC)
summary_lin_withaux<-
c(defaultsummary_lin_withaux[,1],DIC=model_lin_withaux$DIC)
summary_lin_without_with_aux<-cbind(summary_linear,summary_lin_withaux)
write.csv(summary_lin_without_with_aux,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/summary_lin_without_with_aux.csv")

#Convergence plots (linear model with auxiliary Bayesian estimation)
rmeanplot(model_lin_withaux,parms="mu.y1",ylim=c(0.2,0.4))
rmeanplot(model_lin_withaux,parms="lam1")
rmeanplot(model_lin_withaux,parms="gam1")
rmeanplot(model_lin_withaux,parms="sgm1")
rmeanplot(model_lin_withaux,parms="sgd1")
rmeanplot(model_lin_withaux,parms="phx1")

rmeanplot(model_lin_withaux,parms="mu.y2",ylim=c(0.3,0.5))
rmeanplot(model_lin_withaux,parms="lam2")
rmeanplot(model_lin_withaux,parms="gam2")
rmeanplot(model_lin_withaux,parms="sgm2")
rmeanplot(model_lin_withaux,parms="sgd2")
rmeanplot(model_lin_withaux,parms="phx2")

rmeanplot(model_lin_withaux,parms="mu.y3",ylim=c(-0.85,-0.55))
rmeanplot(model_lin_withaux,parms="lam3")
rmeanplot(model_lin_withaux,parms="gam3")
rmeanplot(model_lin_withaux,parms="sgm3")
rmeanplot(model_lin_withaux,parms="sgd3")
rmeanplot(model_lin_withaux,parms="phx3")

#Model comparison
#Interaction multisample model (with main and interaction effects)
estimation without the aid of auxiliary Bayesian estimation
#Note that gam1, gam2 and gam3 are now 3x1 vectors instead of 2x1 vectors
parameters=c("mu.y1", "lam1", "gam1", "sgm1", "sgd1", "phx1",
             "mu.y2", "lam2", "gam2", "sgm2", "sgd2", "phx2",
             "mu.y3", "lam3", "gam3", "sgm3", "sgd3", "phx3")

init1_interact=list(mu.y1=rep(0,8), lam1=rep(0,5), gam1=rep(0,3),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
                    mu.y2=rep(0,8), lam2=rep(0,5), gam2=rep(0,3),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
                    mu.y3=rep(0,8), lam3=rep(0,5), gam3=rep(0,3),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_interact=list(mu.y1=c(0,rep(1,7)), lam1=rep(1,5), gam1=rep(1,3),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
                    mu.y2=c(0,rep(1,7)), lam2=rep(1,5), gam2=rep(1,3),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
                    mu.y3=c(0,rep(1,7)), lam3=rep(1,5), gam3=rep(1,3),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_interact=list(init1_interact, init2_interact)
```

```
R=matrix(c(8, 0, 0, 8), nrow=2)

data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_interact<-bugs(data, inits_interact, parameters,
                     model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/interactionmodel.txt",
                     n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                     working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_interact<-model_interact$summary
write.csv(defaultsummary_interact,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_interact.csv")

#Quadratic multisample model (without interaction term) estimation
without the aid of auxiliary Bayesian estimation
#Note that gam1, gam2 and gam3 are now 4x1 vectors instead of 2x1 vectors
parameters=c("mu.y1", "lam1", "gam1", "sgm1", "sgd1", "phx1",
             "mu.y2", "lam2", "gam2", "sgm2", "sgd2", "phx2",
             "mu.y3", "lam3", "gam3", "sgm3", "sgd3", "phx3")

init1_qua=list(mu.y1=rep(0,8), lam1=rep(0,5), gam1=rep(0,4),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y2=rep(0,8), lam2=rep(0,5), gam2=rep(0,4),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y3=rep(0,8), lam3=rep(0,5), gam3=rep(0,4),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_qua=list(mu.y1=c(0,rep(1,7)), lam1=rep(1,5), gam1=rep(1,4),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y2=c(0,rep(1,7)), lam2=rep(1,5), gam2=rep(1,4),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y3=c(0,rep(1,7)), lam3=rep(1,5), gam3=rep(1,4),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_qua=list(init1_qua, init2_qua)

R=matrix(c(8, 0, 0, 8), nrow=2)

data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_quadratic<-bugs(data, inits_qua, parameters,
                     model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/quadraticmodel.txt",
                     n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                     working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_quadratic<-model_quadratic$summary
write.csv(defaultsummary_quadratic,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_quadratic.csv")

#Second order multisample model (with interaction and quadratic terms)
estimation without the aid of auxiliary Bayesian estimation
#Note that gam1, gam2 and gam3 are now 5x1 vectors instead of 2x1 vectors
parameters=c("mu.y1", "lam1", "gam1", "sgm1", "sgd1", "phx1",
```

```
             "mu.y2", "lam2", "gam2", "sgm2", "sgd2", "phx2",
             "mu.y3", "lam3", "gam3", "sgm3", "sgd3", "phx3")

init1_sec=list(mu.y1=rep(0,8), lam1=rep(0,5), gam1=rep(0,5),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y2=rep(0,8), lam2=rep(0,5), gam2=rep(0,5),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
               mu.y3=rep(0,8), lam3=rep(0,5), gam3=rep(0,5),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_sec=list(mu.y1=c(0,rep(1,7)), lam1=rep(1,5), gam1=rep(1,5),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y2=c(0,rep(1,7)), lam2=rep(1,5), gam2=rep(1,5),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
               mu.y3=c(0,rep(1,7)), lam3=rep(1,5), gam3=rep(1,5),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_sec=list(init1_sec, init2_sec)

R=matrix(c(8, 0, 0, 8), nrow=2)

data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_secorder<-bugs(data, inits_sec, parameters,
                     model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/secordermodel.txt",
                     n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                     working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_secorder<-model_secorder$summary
write.csv(defaultsummary_secorder,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_secorder.csv")

#Model comparison based on DIC
(linear_DIC=model_linear$DIC)           #Main effects only
(interaction_DIC=model_interact$DIC)    #Main and interaction effects
(quadratic_DIC=model_quadratic$DIC)     #Main and quadratic effects
(without interaction)
(secorder_DIC=model_secorder$DIC)       #Full second order model

(DIC_models<-
rbind(linear_DIC,interaction_DIC,quadratic_DIC,secorder_DIC))
write.csv(DIC_models,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/DIC_models.csv")

#Conclusion: Linear multisample model is selected for subsequent use
because of its simplicity and interpretability while
#its DIC value is comparable with others.

#Testing constraints as model comparison (under linear models)
#Under equal intercepts constraints
#Note that mu.y1, mu.y2 and mu.y3 are replaced by mu.y.
parameters=c("mu.y", "lam1", "gam1", "sgm1", "sgd1", "phx1",
                     "lam2", "gam2", "sgm2", "sgd2", "phx2",
                     "lam3", "gam3", "sgm3", "sgd3", "phx3")

init1_eqint=list(mu.y=rep(0,8), lam1=rep(0,5), gam1=rep(0,2),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
```

```
                                lam2=rep(0,5), gam2=rep(0,2),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
                                lam3=rep(0,5), gam3=rep(0,2),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_eqint=list(mu.y=c(0,rep(1,7)), lam1=rep(1,5), gam1=rep(1,2),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
                                    lam2=rep(1,5), gam2=rep(1,2),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
                                    lam3=rep(1,5), gam3=rep(1,2),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_eqint=list(init1_eqint, init2_eqint)

R=matrix(c(8, 0, 0, 8), nrow=2)

data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_eqint<-bugs(data, inits_eqint, parameters,
                      model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/equalintmodel.txt",
                      n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                      working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_eqint<-model_eqint$summary
write.csv(defaultsummary_eqint,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_eqint.csv")

#Under equal lambdas constraints
#Note that lam1, lam2 and lam3 are replaced by lam.
parameters=c("mu.y1", "lam", "gam1", "sgm1", "sgd1", "phx1",
             "mu.y2",        "gam2", "sgm2", "sgd2", "phx2",
             "mu.y3",        "gam3", "sgm3", "sgd3", "phx3")

init1_eqlam=list(mu.y1=rep(0,8), lam=rep(0,5), gam1=rep(0,2),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
                 mu.y2=rep(0,8),               gam2=rep(0,2),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
                 mu.y3=rep(0,8),               gam3=rep(0,2),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_eqlam=list(mu.y1=c(0,rep(1,7)), lam=rep(1,5), gam1=rep(1,2),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
                 mu.y2=c(0,rep(1,7)),               gam2=rep(1,2),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
                 mu.y3=c(0,rep(1,7)),               gam3=rep(1,2),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_eqlam=list(init1_eqlam, init2_eqlam)

R=matrix(c(8, 0, 0, 8), nrow=2)

data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_eqlam<-bugs(data, inits_eqlam, parameters,
                   model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/equallammodel.txt",
```

```
                  n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                  working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_eqlam<-model_eqlam$summary
write.csv(defaultsummary_eqlam,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_eqlam.csv")

#Under equal gammas constraints
#Note that gam1, gam2 and gam3 are replaced by gam.
parameters=c("mu.y1", "lam1", "gam", "sgm1", "sgd1", "phx1",
             "mu.y2", "lam2",        "sgm2", "sgd2", "phx2",
             "mu.y3", "lam3",        "sgm3", "sgd3", "phx3")

init1_eqgam=list(mu.y1=rep(0,8), lam1=rep(0,5), gam=rep(0,2),
psi1=c(9999,rep(1,7)), psd1=1, phi1=matrix(c(1, 0, 0, 1), nrow=2),
                 mu.y2=rep(0,8), lam2=rep(0,5),
psi2=c(9999,rep(1,7)), psd2=1, phi2=matrix(c(1, 0, 0, 1), nrow=2),
                 mu.y3=rep(0,8), lam3=rep(0,5),
psi3=c(9999,rep(1,7)), psd3=1, phi3=matrix(c(1, 0, 0, 1), nrow=2))

init2_eqgam=list(mu.y1=c(0,rep(1,7)), lam1=rep(1,5), gam=rep(1,2),
psi1=c(9999,rep(2,7)), psd1=2, phi1=matrix(c(2, 0, 0, 2), nrow=2),
                 mu.y2=c(0,rep(1,7)), lam2=rep(1,5),
psi2=c(9999,rep(2,7)), psd2=2, phi2=matrix(c(2, 0, 0, 2), nrow=2),
                 mu.y3=c(0,rep(1,7)), lam3=rep(1,5),
psi3=c(9999,rep(2,7)), psd3=2, phi3=matrix(c(2, 0, 0, 2), nrow=2))

inits_eqgam=list(init1_eqgam, init2_eqgam)

R=matrix(c(8, 0, 0, 8), nrow=2)

data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_eqgam<-bugs(data, inits_eqgam, parameters,
                  model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/equalgammodel.txt",
                  n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                  working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_eqgam<-model_eqgam$summary
write.csv(defaultsummary_eqgam,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_eqgam.csv")

#Under equal covariance matrices of the two explanatory latent variables
constraints
#Note that phi1, phi2 and phi3 are replaced by phi while phx1, phx2 and
phx3 are replaced by phx.
parameters=c("mu.y1", "lam1", "gam1", "sgm1", "sgd1", "phx",
             "mu.y2", "lam2", "gam2", "sgm2", "sgd2",
             "mu.y3", "lam3", "gam3", "sgm3", "sgd3")

init1_eqcov=list(mu.y1=rep(0,8), lam1=rep(0,5), gam1=rep(0,2),
psi1=c(9999,rep(1,7)), psd1=1, phi=matrix(c(1, 0, 0, 1), nrow=2),
                 mu.y2=rep(0,8), lam2=rep(0,5), gam2=rep(0,2),
psi2=c(9999,rep(1,7)), psd2=1,
```

```
                  mu.y3=rep(0,8), lam3=rep(0,5), gam3=rep(0,2),
psi3=c(9999,rep(1,7)), psd3=1)

init2_eqcov=list(mu.y1=c(0,rep(1,7)), lam1=rep(1,5), gam1=rep(1,2),
psi1=c(9999,rep(2,7)), psd1=2, phi=matrix(c(2, 0, 0, 2), nrow=2),
                  mu.y2=c(0,rep(1,7)), lam2=rep(1,5), gam2=rep(1,2),
psi2=c(9999,rep(2,7)), psd2=2,
                  mu.y3=c(0,rep(1,7)), lam3=rep(1,5), gam3=rep(1,2),
psi3=c(9999,rep(2,7)), psd3=2)

inits_eqcov=list(init1_eqcov, init2_eqcov)

R=matrix(c(8, 0, 0, 8), nrow=2)

data=list(N1=N1, N2=N2, N3=N3, zero=c(0,0), R=R, y1=V1, y2=V2, y3=V3)

model_eqcov<-bugs(data, inits_eqcov, parameters,
                   model.file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/equalcovmodel.txt",
                   n.chains=2, n.iter=10000, n.burnin=5000, n.thin=1,
bugs.directory="C:/Program Files/WinBUGS14/",
                   working.directory="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/")

defaultsummary_eqcov<-model_eqcov$summary
write.csv(defaultsummary_eqcov,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/defaultsummary_eqcov.csv")

#Testing constraints based on DIC
(noconstraints_DIC=model_linear$DIC)   #Linear Model without constraints
(equalint_DIC=model_eqint$DIC)         #Equal intercepts
(equallam_DIC=model_eqlam$DIC)         #Equal lambdas
(equalgam_DIC=model_eqgam$DIC)         #Equal gammas
(equalcov_DIC=model_eqcov$DIC)         #Equal covariance matrices of
explanatory latent variables

(DIC_testconstraints<-
rbind(noconstraints_DIC,equalint_DIC,equallam_DIC,equalgam_DIC,equalcov_D
IC))
write.csv(DIC_testconstraints,file="C:/Users/Benjamin
Chan/Desktop/CUHK/Stat/5020/Project/DIC_testconstraints.csv")

#Conclusion: Multisample model is useful in the sense that the DIC is
samllest without constraints for all linear models considered.
```

```
#Linear SEM with multisample data (linearmodel.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)

        nu3[i]<-gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]

        xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
```

```
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y1[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam1[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam1[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y2[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam2[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam2[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y3[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam3[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam3[i]~dnorm(0.6,0.4)}

    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Linear Multisample SEM with first new prior
(newprior1model.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)

        nu3[i]<-gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]
```

```
        xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y1[i]~dnorm(-1,0.4)}
    for(i in 1:5){lam1[i]~dnorm(0.2,0.4)}
    for(i in 1:2){gam1[i]~dnorm(0.4,0.4)}
    for(i in 1:8){mu.y2[i]~dnorm(1,0.4)}
    for(i in 1:5){lam2[i]~dnorm(0.2,0.4)}
    for(i in 1:2){gam2[i]~dnorm(0.4,0.4)}
    for(i in 1:8){mu.y3[i]~dnorm(0,0.4)}
    for(i in 1:5){lam3[i]~dnorm(0.4,0.4)}
    for(i in 1:2){gam3[i]~dnorm(0.7,0.4)}

    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Linear Multisample SEM with second new prior
(newprior2model.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)

        nu3[i]<-gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]
```

```
        xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y1[i]~dnorm(0.2,0.4)}
    for(i in 1:5){lam1[i]~dnorm(0.7,0.4)}
    for(i in 1:2){gam1[i]~dnorm(0.5,0.4)}
    for(i in 1:8){mu.y2[i]~dnorm(0.4,0.4)}
    for(i in 1:5){lam2[i]~dnorm(0.3,0.4)}
    for(i in 1:2){gam2[i]~dnorm(0.5,0.4)}
    for(i in 1:8){mu.y3[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam3[i]~dnorm(0.7,0.4)}
    for(i in 1:2){gam3[i]~dnorm(0.7,0.4)}

    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Linear Multisample SEM with auxiliary estimation
(linearwithauxmodel.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)

        nu3[i]<-gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]
```

```
    xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
} #end of i

#priors on loadings and coefficients
mu.y1[1]~dnorm(0.23776977,4)
mu.y1[2]~dnorm(0.30800997,4)
mu.y1[3]~dnorm(0.31311459,4)
mu.y1[4]~dnorm(0.28061309,4)
mu.y1[5]~dnorm(0.32791675,4)
mu.y1[6]~dnorm(0.32748821,4)
mu.y1[7]~dnorm(0.31516263,4)
mu.y1[8]~dnorm(0.30663321,4)

lam1[1]~dnorm(0.99121838,4)
lam1[2]~dnorm(0.85018533,4)
lam1[3]~dnorm(1.01047776,4)
lam1[4]~dnorm(0.9610022,4)
lam1[5]~dnorm(0.91471167,4)

gam1[1]~dnorm(0.53051649,4)
gam1[2]~dnorm(-0.086271204206,4)

mu.y2[1]~dnorm(0.3701532,4)
mu.y2[2]~dnorm(0.45751105,4)
mu.y2[3]~dnorm(0.4707897,4)
mu.y2[4]~dnorm(0.44102394,4)
mu.y2[5]~dnorm(0.44263698,4)
mu.y2[6]~dnorm(0.38971012,4)
mu.y2[7]~dnorm(0.45556052,4)
mu.y2[8]~dnorm(0.45253739,4)

lam2[1]~dnorm(0.98658896,4)
lam2[2]~dnorm(0.79141268,4)
lam2[3]~dnorm(0.97279875,4)
lam2[4]~dnorm(0.9680939,4)
lam2[5]~dnorm(0.94853086,4)

gam2[1]~dnorm(0.260710912323,4)
gam2[2]~dnorm(0.077336998146,4)

mu.y3[1]~dnorm(-0.63370484,4)
mu.y3[2]~dnorm(-0.80187014,4)
mu.y3[3]~dnorm(-0.8204269,4)
mu.y3[4]~dnorm(-0.75382003,4)
mu.y3[5]~dnorm(-0.81080713,4)
mu.y3[6]~dnorm(-0.75877274,4)
mu.y3[7]~dnorm(-0.8085693,4)
mu.y3[8]~dnorm(-0.79528537,4)

lam3[1]~dnorm(0.9782375,4)
lam3[2]~dnorm(0.7937561,4)
lam3[3]~dnorm(0.95368988,4)
lam3[4]~dnorm(0.94904175,4)
lam3[5]~dnorm(1.01238725,4)

gam3[1]~dnorm(0.3750694,4)
gam3[2]~dnorm(0.3938962559,4)

#priors on precisions
for (j in 1:8) {
```

```
        psi1[j]~dgamma(9,4)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(9,4)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R1[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(9,4)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(9,4)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R2[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(9,4)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(9,4)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R3[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Multisample SEM with interation term (interactionmodel.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-
gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]+gam1[3]*xi1[i,1]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-
gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]+gam2[3]*xi2[i,1]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)
```

```
        nu3[i]<-
gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]+gam3[3]*xi3[i,1]*xi3[i,2]

        xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y1[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam1[i]~dnorm(0.8,0.4)}
    for(i in 1:3){gam1[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y2[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam2[i]~dnorm(0.8,0.4)}
    for(i in 1:3){gam2[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y3[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam3[i]~dnorm(0.8,0.4)}
    for(i in 1:3){gam3[i]~dnorm(0.6,0.4)}

    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Multisample SEM with quadratic term (quadraticmodel.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-
gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]+gam1[3]*xi1[i,1]*xi1[i,1]+gam1[4]*xi1[i
,2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-
gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]+gam2[3]*xi2[i,1]*xi2[i,1]+gam2[4]*xi2[i
,2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)
```

```
        nu3[i]<-
gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]+gam3[3]*xi3[i,1]*xi3[i,1]+gam3[4]*xi3[i
,2]*xi3[i,2]

        xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y1[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam1[i]~dnorm(0.8,0.4)}
    for(i in 1:4){gam1[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y2[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam2[i]~dnorm(0.8,0.4)}
    for(i in 1:4){gam2[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y3[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam3[i]~dnorm(0.8,0.4)}
    for(i in 1:4){gam3[i]~dnorm(0.6,0.4)}

    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Multisample SEM with second order terms (secordermodel.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-
gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]+gam1[3]*xi1[i,1]*xi1[i,2]+gam1[4]*xi1[i
,1]*xi1[i,1]+gam1[5]*xi1[i,2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-
gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]+gam2[3]*xi2[i,1]*xi2[i,2]+gam2[4]*xi2[i
,1]*xi2[i,1]+gam2[5]*xi2[i,2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)
```

```
        nu3[i]<-
gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]+gam3[3]*xi3[i,1]*xi3[i,2]+gam3[4]*xi3[i
,1]*xi3[i,1]+gam3[5]*xi3[i,2]*xi3[i,2]

        xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y1[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam1[i]~dnorm(0.8,0.4)}
    for(i in 1:5){gam1[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y2[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam2[i]~dnorm(0.8,0.4)}
    for(i in 1:5){gam2[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y3[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam3[i]~dnorm(0.8,0.4)}
    for(i in 1:5){gam3[i]~dnorm(0.6,0.4)}

    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Linear Multisample SEM with equal intercepts
(equalintmodel.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y[1]+eta1[i]
        mu1[i,2]<-mu.y[2]+xi1[i,1]
        mu1[i,3]<-mu.y[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y[5]+xi1[i,2]
        mu1[i,6]<-mu.y[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y[1]+eta2[i]
        mu2[i,2]<-mu.y[2]+xi2[i,1]
        mu2[i,3]<-mu.y[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y[5]+xi2[i,2]
        mu2[i,6]<-mu.y[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y[1]+eta3[i]
        mu3[i,2]<-mu.y[2]+xi3[i,1]
        mu3[i,3]<-mu.y[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y[5]+xi3[i,2]
        mu3[i,6]<-mu.y[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)

        nu3[i]<-gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]
```

```
        xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam1[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam1[i]~dnorm(0.6,0.4)}
    for(i in 1:5){lam2[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam2[i]~dnorm(0.6,0.4)}
    for(i in 1:5){lam3[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam3[i]~dnorm(0.6,0.4)}

    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Linear Multisample SEM with equal lambda (equallammodel.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)

        nu3[i]<-gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]

        xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
```

```
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y1[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam1[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y2[i]~dnorm(0.0,0.4)}
    for(i in 1:2){gam2[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y3[i]~dnorm(0.0,0.4)}
    for(i in 1:2){gam3[i]~dnorm(0.6,0.4)}

    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Linear Multisample SEM with equal gamma (equalgammodel.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-gam[1]*xi1[i,1]+gam[2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi1[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-gam[1]*xi2[i,1]+gam[2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi2[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)

        nu3[i]<-gam[1]*xi3[i,1]+gam[2]*xi3[i,2]

        xi3[i,1:2]~dmnorm(zero[1:2], phi3[1:2,1:2])
```

```
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y1[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam1[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y2[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam2[i]~dnorm(0.8,0.4)}
    for(i in 1:8){mu.y3[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam3[i]~dnorm(0.8,0.4)}


    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi1[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx1[1:2,1:2]<-inverse(phi1[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2
    phi2[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx2[1:2,1:2]<-inverse(phi2[1:2,1:2])

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
    phi3[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx3[1:2,1:2]<-inverse(phi3[1:2,1:2])
}#end of model
```

```
#WinBUGS: Linear Multisample SEM with equal covariance matrix of latent
var (equalcovmodel.txt)
model {
    #Group one
    for (i in 1:N1) {
        for (j in 1:8) { y1[i,j]~dnorm(mu1[i,j], psi1[j]) }
        mu1[i,1]<-mu.y1[1]+eta1[i]
        mu1[i,2]<-mu.y1[2]+xi1[i,1]
        mu1[i,3]<-mu.y1[3]+lam1[1]*xi1[i,1]
        mu1[i,4]<-mu.y1[4]+lam1[2]*xi1[i,1]
        mu1[i,5]<-mu.y1[5]+xi1[i,2]
        mu1[i,6]<-mu.y1[6]+lam1[3]*xi1[i,2]
        mu1[i,7]<-mu.y1[7]+lam1[4]*xi1[i,2]
        mu1[i,8]<-mu.y1[8]+lam1[5]*xi1[i,2]

        #structural equation
        eta1[i]~dnorm(nu1[i], psd1)

        nu1[i]<-gam1[1]*xi1[i,1]+gam1[2]*xi1[i,2]

        xi1[i,1:2]~dmnorm(zero[1:2], phi[1:2,1:2])
    } #end of i

    #Group two
    for (i in 1:N2) {
        for (j in 1:8) { y2[i,j]~dnorm(mu2[i,j], psi2[j]) }
        mu2[i,1]<-mu.y2[1]+eta2[i]
        mu2[i,2]<-mu.y2[2]+xi2[i,1]
        mu2[i,3]<-mu.y2[3]+lam2[1]*xi2[i,1]
        mu2[i,4]<-mu.y2[4]+lam2[2]*xi2[i,1]
        mu2[i,5]<-mu.y2[5]+xi2[i,2]
        mu2[i,6]<-mu.y2[6]+lam2[3]*xi2[i,2]
        mu2[i,7]<-mu.y2[7]+lam2[4]*xi2[i,2]
        mu2[i,8]<-mu.y2[8]+lam2[5]*xi2[i,2]

        #structural equation
        eta2[i]~dnorm(nu2[i], psd2)

        nu2[i]<-gam2[1]*xi2[i,1]+gam2[2]*xi2[i,2]

        xi2[i,1:2]~dmnorm(zero[1:2], phi[1:2,1:2])
    } #end of i

    #Group three
    for (i in 1:N3) {
        for (j in 1:8) { y3[i,j]~dnorm(mu3[i,j], psi3[j]) }
        mu3[i,1]<-mu.y3[1]+eta3[i]
        mu3[i,2]<-mu.y3[2]+xi3[i,1]
        mu3[i,3]<-mu.y3[3]+lam3[1]*xi3[i,1]
        mu3[i,4]<-mu.y3[4]+lam3[2]*xi3[i,1]
        mu3[i,5]<-mu.y3[5]+xi3[i,2]
        mu3[i,6]<-mu.y3[6]+lam3[3]*xi3[i,2]
        mu3[i,7]<-mu.y3[7]+lam3[4]*xi3[i,2]
        mu3[i,8]<-mu.y3[8]+lam3[5]*xi3[i,2]

        #structural equation
        eta3[i]~dnorm(nu3[i], psd3)

        nu3[i]<-gam3[1]*xi3[i,1]+gam3[2]*xi3[i,2]
```

```
        xi3[i,1:2]~dmnorm(zero[1:2], phi[1:2,1:2])
    } #end of i

    #priors on loadings and coefficients
    for(i in 1:8){mu.y1[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam1[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam1[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y2[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam2[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam2[i]~dnorm(0.6,0.4)}
    for(i in 1:8){mu.y3[i]~dnorm(0.0,0.4)}
    for(i in 1:5){lam3[i]~dnorm(0.8,0.4)}
    for(i in 1:2){gam3[i]~dnorm(0.6,0.4)}

    #priors on precisions
    for (j in 1:8) {
        psi1[j]~dgamma(6,10)
      sgm1[j]<-1/psi1[j]
    }

    psd1~dgamma(6,10)
    sgd1<-1/psd1
    phi[1:2,1:2]~dwish(R[1:2,1:2], 4)
    phx[1:2,1:2]<-inverse(phi[1:2,1:2])

    for (j in 1:8) {
        psi2[j]~dgamma(6,10)
      sgm2[j]<-1/psi2[j]
    }

    psd2~dgamma(6,10)
    sgd2<-1/psd2

    for (j in 1:8) {
        psi3[j]~dgamma(6,10)
      sgm3[j]<-1/psi3[j]
    }

    psd3~dgamma(6,10)
    sgd3<-1/psd3
}#end of model
```