The Bayesian Method of Estimation for the Number of Latent Classes in Growth Mixture Models

Chittanun Sitthisan

Follow this and additional works at: http://digscholarship.unco.edu/dissertations

Recommended Citation
http://digscholarship.unco.edu/dissertations/389
UNIVERSITY OF NORTHERN COLORADO

Greeley, Colorado

The Graduate School

THE BAYESIAN METHOD OF ESTIMATION FOR THE
NUMBER OF LATENT CLASSES IN GROWTH
MIXTURE MODELS

A Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

Chittanun Sitthisan

College of Education and Behavioral Sciences
Department of Applied Statistics and Research Methods

December, 2016
This dissertation by: Chittanun Sitthisan

Entitled: *The Bayesian Method of Estimation for the Number of Latent Classes in Growth Mixture Models*

has been approved as meeting the requirement for the Degree of Doctor of Philosophy in College of Education and Behavioral Sciences in Department of Applied Statistics and Research Methods

Accepted by the Doctoral Committee

__________________________________
Khalil Shafie, Ph.D., Research Advisor

__________________________________
Susan R. Hutchinson, Ph.D., Committee Member

__________________________________
Jay R. Schaffer, Ph.D., Committee Member

__________________________________
Trent L. Lalonde, Ph.D., Committee Member

__________________________________
Mehrgan Mostowfi, Ph.D., Faculty Representative

Date of Dissertation Defense

__________________________________
Accepted by the Graduate School

__________________________________
Linda L. Black, Ed.D.
Associate Provost and Dean
Graduate School and International Admissions
ABSTRACT

Sitthisan, Chittanun. The Bayesian Method of Estimation for the Number of Latent Classes in Growth Mixture Models Published Doctor of Philosophy dissertation, University of Northern Colorado, 2016.

It is widely accepted that blindly specifying an incorrect number of latent classes may result in misidentifying the class membership of observations and in inconsistently estimated parameters. The current dissertation examined the Bayesian method to estimate the number of latent classes in growth mixture models. The procedure for estimating the number of latent classes was developed via Markov chain Monte Carlo using the Metropolis-Hastings algorithm. The key idea was to construct the likelihood function and then specify the prior information toward the number of latent classes ($K$), and then calculate the posterior distribution for $K$. Simulated observations were generated by the Metropolis-Hastings sampling technique from the posterior distribution. The average value of $K$ was used under Bayesian method to estimate the number of latent classes. Other growth parameters were produced as by-products. The properties and merits of the proposed procedure were illustrated by means of a simulation study through a written R program.

It was found that the Bayesian performance of estimation depended on the informative prior toward the number of latent classes only through the complexity
of the growth mixture model. Additionally, the Bayesian method was optimal for both small and large sample sizes. It performs much better when the model consists of many latent classes with larger values of the unknown parameters. These properties could be useful in applied research. However, the number of time points had less influence on the latent class estimation. In conclusion, it can be said that the accuracy of the estimation of the number of components on GMM underperformed for a less complex model and a small sample size.

Based on the results of this dissertation, it is suggested that covariates be added when performing sampling of posterior distribution using the Metropolis-Hastings method on the basis of Markov chain Monte Carlo. Procedures relying on the Bayesian approach should be avoided when the mixture of subpopulation is less than three groups. This is mainly because the performance of such estimation techniques is generally poor. Another technique such as reversible jump Markov chain Monte Carlo can be conducted on unconditional growth mixture models under the Bayesian framework.
ACKNOWLEDGEMENTS

First of all, I would like to express my special appreciation and thanks to my advisor, Dr. Khalil Shafie. His help and supports extend so far beyond this dissertation than I can formulate words to properly express my gratitude for his guidance. He has been more involved and attentive than I could expect from an advisor. Through the dissertation process, he encouraged me to learn and showed me that there was no limit of gaining knowledge. He is mindful of training me to be an independent researcher, and is also my model professor since he has never been once too busy to help guide me through research challenges. Now that my time here has come to a close, and I realize how many opportunities Dr. Khalil Shafie has provided for me with his patience, motivation, and immense knowledge. Now, I am strong enough and can walk out of University of Northern Colorado (UNC) proudly with doors open in my life and my future career has truly changed. For these reasons, it is suitable for me to fully thank Dr. Khalil Shafie for what he has done for me.

In addition, I would like to thank Dr. Susan Hutchinson for her solid guidance and suggestions that helped shape this document and its contents. From the very first time I met her, I could see what a caring and devoted professor she was. Later, I learned that she exceeded my initial impression. Along the way, I drifted away from time to time from my routine study because of my frustrations
caused by both academic and vocational difficulties. Very fortunately, it was her generous patience and warmest encouragement that brought me back on my track. I am eternally thankful to her for providing me with perspectives that strengthened not only my dissertation but also me as a researcher.

My sincere thanks also go to Dr. Jay Shafer who has provided me with invaluable chances to obtain hands-on research experiences in the computer lab and accepted me as a teaching assistant. Without his precious supports and understandings, it would not be possible for me to achieve my goal. I also sincerely appreciate Dr. Trent Lalonde for his useful and practical advices and encouragements throughout this difficult project. Besides, I would also like to thank my friends and colleagues at UNC and ASRM. To mention just only a few, a special thank goes to Zabedah, Jay, and Jamie for their friendship in the lab.

Furthermore, I would like to express my particular gratitude to Chandrakasem Rajabhat University for granting me a scholarship to further my study for a Ph.D. at UNC. Without this very important initial assistance, my dream can never become true. At the same time, I would like to thank my Thai friends at UNC: Sarinda, Alisa, Raveema, Udon, and Weeramol. They were always there when I needed help during my study at UNC.

Last but not least, I would like to thank my parents, Mr. Kovit and Mrs. Umphan Sitthisan, for their supports, encouragements, understandings, and unconditional love. No one could have asked for more encouraging and enthusiastic parents, and I can never thank them enough for everything they have done for me leading me up to this moment. I would also like to thank my sister, Prapawan, and
my brother, Thamanoon, for supporting me spiritually throughout the writing
period of this dissertation and my life in general. Generally speaking, I consider
myself a truly lucky person to have such a supportive family, and cannot wait to see
a greater future soon.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Latent Growth Curve Models</td>
<td>4</td>
</tr>
<tr>
<td>Growth Mixture Models</td>
<td>8</td>
</tr>
<tr>
<td>Specifying Number of Latent Classes</td>
<td>14</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>15</td>
</tr>
<tr>
<td>Purpose of the Study</td>
<td>19</td>
</tr>
<tr>
<td>Research Questions</td>
<td>20</td>
</tr>
<tr>
<td>Chapter Summary</td>
<td>20</td>
</tr>
<tr>
<td>II REVIEW OF LITERATURE</td>
<td>22</td>
</tr>
<tr>
<td>Overview of Growth Mixture Modeling</td>
<td>23</td>
</tr>
<tr>
<td>Factor Analysis</td>
<td>26</td>
</tr>
<tr>
<td>Structural Equation Modeling</td>
<td>27</td>
</tr>
<tr>
<td>Model of Change</td>
<td>32</td>
</tr>
<tr>
<td>Growth Curve Models</td>
<td>33</td>
</tr>
<tr>
<td>Mixture Modeling</td>
<td>43</td>
</tr>
<tr>
<td>Growth Mixture Modeling</td>
<td>51</td>
</tr>
</tbody>
</table>
Studies Examining Unknown

Number of Latent Classes .................................. 57
Bayesian Methods ........................................... 59
Chapter Summary ............................................ 78

III METHODOLOGY ........................................... 80
Growth Mixture Models ...................................... 81
The Unconditional Growth Mixture Model .................. 82
Bayesian Estimation for the
Growth Mixture Models ...................................... 87
Markov Chain Monte Carlo Simulation ..................... 95
Verifying the Validity of a Simulation ....................... 100
Analysis of Simulated Data .................................. 107

IV RESULTS ...................................................... 109
Model Convergence .......................................... 111
The Performance of the Estimation ......................... 125
Summary of the Estimation .................................. 150
Parameter Estimates Using
Mplus and R Program ........................................ 154
Chapter Summary ............................................ 178

V CONCLUSIONS ............................................. 180
Choosing the Proposed and Prior Distribution ............. 180
LIST OF TABLES

1  Class size specification in growth mixture models .................. 102
2  Design Summary .......................................................... 103
3  The estimated Posterior Variance (\(\hat{\sigma}^2\)) of the Number of Latent Classes (K) for Informative Prior (Poisson with \(\lambda = 3\)) .................. 112
4  The Estimated Posterior Variance (\(\hat{\sigma}^2\)) of the Number of Latent Classes (K) for Noninformative Prior (Discrete Uniform with parameter 3) ....................................................... 113
5  The Estimated Posterior Variance (\(\hat{\sigma}^2\)) of the Number of Latent Classes (K) for Informative Prior (Poisson with \(\lambda = 4\)) .................. 114
6  The Estimated Posterior Variance (\(\hat{\sigma}^2\)) of the Number of Latent Classes (K) for Noninformative Prior (Discrete Uniform with parameter 4) ....................................................... 115
7  The Estimated Posterior Variance (\(\hat{\sigma}^2\)) of the Number of Latent Classes (K) for Informative Prior (Poisson with \(\lambda = 5\)) .................. 116
8  The Estimated Posterior Variance (\(\hat{\sigma}^2\)) of the Number of Latent Classes (K) for Noninformative Prior (Discrete Uniform with parameter 5) ....................................................... 117
The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Informative Prior (Poisson with \( \lambda = 3 \)) .......................... 119

The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Noninformative Prior (Discrete Uniform with parameter 3) .......................... 120

The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Informative Prior (Poisson with \( \lambda = 4 \)) .......................... 121

The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Noninformative Prior (Discrete Uniform with parameter 4) .......................... 122

The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Informative Prior (Poisson with \( \lambda = 5 \)) .......................... 123

The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Noninformative Prior (Discrete Uniform with parameter 5) .......................... 124

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with \( N = 15 \) for Informative Prior (Poisson with \( \lambda = 3, 4, 5 \)) ...................................................... 131

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with \( N = 15 \) for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5) ...................................................... 132
Parameter Estimates, 95% Confidence Intervals, Empirical Standard Error, and Convergence for Number of Latent Classes (K) on 2-class GMM with \( N = 15 \) for Informative Prior

(Poisson with \( \lambda = 3, 4, 5 \)) ......................................................... 133

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Error, and Convergence for Number of Latent Classes (K) on 2-class GMM with \( N = 15 \) for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5) ......................................................... 134

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with \( N = 15 \) for Informative Prior

(Poisson with \( \lambda = 3, 4, 5 \)) ......................................................... 135

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with \( N = 15 \) for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5) ......................................................... 136

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with \( N = 50 \) for Informative Prior

(Poisson with \( \lambda = 3, 4, 5 \)) ......................................................... 137
22 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with N = 50 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 138

23 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 3-class GMM with N = 50 for Informative Prior (Poisson with \( \lambda = 3, 4, 5 \)) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139

24 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 3-class GMM with N = 50 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 140

25 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 4-class GMM with N = 50 for Informative Prior (Poisson with \( \lambda = 3, 4, 5 \)) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 141

26 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 4-class GMM N = 50 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142

27 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence for Number of Latent Classes (K) on 2-class with N = 200 for Informative Prior (Poisson with \( \lambda = 3, 4, 5 \)) . . . . . . 143
28 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence for Number of Latent Classes (K) on 2-class GMM N = 200 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5) .......................... 144

29 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, Convergence for Number of Latent Classes (K) on 3-class GMM with N = 200 for Informative Prior (Poisson with λ = 3, 4, 5) .......................... 145

30 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence for Number of Latent Classes (K) on 3-class GMM with N = 200 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5) .......................... 146

31 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence for Number of Latent Classes (K) on 4-class GMM with N = 200 for Informative Prior (Poisson with λ = 3, 4, 5) .......................... 147

32 Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence for Number of Latent Classes (K) on 4-class GMM with N = 200 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5) .......................... 148

33 Summary of Estimated Number of Latent Classes (K) for the True Number of Latent Classes = 2 with Different Priors .......................... 152

34 Summary of Estimated Number of Latent Classes (K) for the True Number of Latent Classes = 3 with Different Priors .......................... 153
35 Summary of Estimated Number of Latent Classes (K) for the True
Number of Latent Classes = 4 with Different Priors ............... 154
36 Model Fit Statistics for Different Numbers of Latent Classes ....... 156
37 Estimates of Parameters from Simulated Data Set for the
Growth Mixture Model with 5 Time Points and
3-class Model Using Mplus ........................................... 160
38 Estimates of Parameters from Simulated Data Set for the Growth Mix-
ture Model with 4 Time Points and 4-class Model
Using Mplus ............................................................. 161
39 Estimates of Parameters from Simulated Data Set for the Growth Mix-
ture Model with 4 Time Points and 4-class Model
Using Mplus (continued) ............................................... 162
40 Parameter Estimates for Growth Mixture Models with Poisson Priors
(λ = 4) at Sample size 15, 5 Time Points and 4-class model Using R
Program .............................................................. 165
41 Parameter Estimates for Growth Mixture Models with Poisson Priors
(λ = 4) at Sample size 15, 5 Time Points and 4-class model Using R
Program (continued) ............................................... 166
42 Parameter Estimates for Growth Mixture Models with Poisson Priors
(λ = 5) at Sample size 15, 3 Time Points and 4-class model Using R
Program .............................................................. 167
Parameter Estimates for Growth Mixture Models with Poisson Priors
\((\lambda = 5)\) at Sample size 15, 3 Time Points and 4-class model Using R Program (continued) .......................................................... 168

Parameter Estimates for Growth Mixture Models with Poisson Priors
\((\lambda = 5)\) at Sample size 50, 4 Time Points and 4-class model Using R Program .......................................................... 169

Parameter Estimates for Growth Mixture Models with Poisson Priors
\((\lambda = 5)\) at Sample size 50, 4 Time Points and 4-class model Using R Program (continued) .......................................................... 170

Parameter Estimates for Growth Mixture Models with Poisson Priors
\((\lambda = 5)\) at Sample size 50, 5 Time Points and 4-class model Using R Program .......................................................... 171

Parameter Estimates for Growth Mixture Models with Poisson Priors
\((\lambda = 5)\) at Sample size 50, 5 Time Points and 4-class model Using R Program (continued) .......................................................... 172

Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors \((N = 3)\) at Sample size 15, 5 Time Points and 3-class model Using R Program .......................................................... 173

Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors \((N = 3)\) at Sample size 15, 5 Time Points and 3-class model Using R Program (continued) .......................................................... 174
Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors ($N = 4$) at Sample size 15, 5 Time Points and 3-class model Using R Program ........................................... 175

Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors ($N = 3$) at Sample size 15, 5 Time Points and 3-class model Using R Program (continued) .................................. 176

Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors ($N = 3$) at Sample size 200, 3 Time Points and 3-class model R Program .............................................................. 177

Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors ($N = 3$) at Sample size 200, 3 Time Points and 3-class model Using R Program (continued) ............................ 178
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example of confirmatory factory analysis.</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>Hypothesized full structural equation model, ( \delta ) and ( \varepsilon = \text{error} ).</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>Latent growth curve model.</td>
<td>42</td>
</tr>
<tr>
<td>4</td>
<td>General Modeling Framework.</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>Growth mixture model without covariates.</td>
<td>82</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

What is change? Presumably a simple question, but the consensus of the answer is different in empirical research depending on the discipline. The investigation of change appears in various fields of studies. To a physician, a change might be a decline of the cholesterol levels in the patients after taking new medicine. To a scholar, a change might be increasing students’ achievement in their areas of expertise when approaching recent innovations. To a medical researcher, a change might consist of symptom relief among subjects in an experimental group who receive a discovery drug treatment. To psychologists and therapists, a change might focus on behavioral development in patients. In addition, the fields of biology, agriculture, economics, and marketing are committed to determining if change has occurred due to technological innovations.

However, the investigation of change is a fundamental question of many researchers in almost every discipline. It is therefore important to study how change is measured. As a result, the body of knowledge concerning change is discussed to provide a conceptual and mathematical framework for the measurement of change. Two main types of change are group changes and individual changes. The analysis of group differences and individual differences are discussed, as there are two ways of
statistical thinking about the analysis of change. These two types of differences need different statistical procedures to approach them. Some researchers apply statistical tools in terms of modeling individual observations either for analyzing change or for assessing the strength and direction of a relationship. Researchers who are typically more interested in group differences prefer the analysis of variance (ANOVA) which compares between-group differences and within-group differences, while the analysis of change focusing on individual change over time could be achieved using hierarchical linear modeling (Goldstein, 2011; Raudenbush & Bryk, 2002).

However, there exist some characteristics of individual change that cannot be adequately described by observable variables. Studying variables that cannot be directly observed leads to more complex models in terms of measurement of change. This type of model requires an efficient method of measurement. Researchers in the social sciences are interested in applications that examine abstract variables such as attitudes, feelings, motives, and expectations to measure causes or consequences of observed behaviors (K. Bollen, 1989). Moreover, abstract variables could serve as the representative of psychological constructs that are impossible to measure directly. Alternative names of these conceptual variables are latent variables, unobserved variables, unmeasured variables, consturcts, and factors (K. Bollen, 1989).

These latent variables or factors would usually be hypothesized to have a direct effect on observed variables and could be inferred from patterns of association among sets of observed variables and one or more factors. The patterns of association could be shown in covariance matrices. The primary statistical tool for
analyzing relations between observed and latent variables based on covariance matrices, is structural equation modeling (SEM).

Structural equation modeling for continuous latent variables is the most well-known SEM methodology (Kline, 2011). Structural equation modeling, known as covariance structure analysis or latent variable modeling, is a general modeling framework that allows for specifying and testing hypothesized patterns of relationships among sets of observed and unobserved variables. Observed variables can be categorical, ordinal, or continuous in SEM (Kline, 2011). Structural models, as developed by Jöreskog (1973) can be decomposed into two parts: the measurement and the structural models. The measurement part specifies relations between observed variables and latent variables. In contrast, the structural part defines relationship among latent variables to each other. The basic statistic of SEM is the covariance aimed at two main goals of SEM namely “(a) to understand patterns of covariances among a set of observed variables and, (b) to explain as much of their variance as possible with the researcher’s model” (Kline, 2011, p. 10).

The fundamental covariance structure hypothesis tested in SEM is whether the population covariance matrix is close enough to the covariance matrix derived from the hypothesized model. The covariance structure hypothesis can be written in the form \( \sum = \sum (\theta) \), where \( \sum \) is the population covariance matrix of observed variables and \( \sum (\theta) \) is the covariance matrix of structural parameters derived from a vector of hypothesized model parameters, \( \theta \) (Bollen, 1989). The parameter vector contains all parameters in SEM.
Parameter estimates in SEM can be obtained by fitting the model, often applying a maximum likelihood (ML) method which is the most widely used fitting method in SEM (Kline, 2011). Like other statistical techniques, SEM requires several assumptions to be met in order to ensure reliable results in evaluating estimation and evaluation of model fit. According to Jöreskog (1973), the required assumptions are as follows: (a) observations are independent, continuous in nature, and drawn from a multivariate normal distribution; and (b) the hypothesized model is appropriate. Since Jöreskog’s early development of SEM in the early 1970s, there have been extensive applications of SEM to many different disciplines for evaluating researchers’ theories in such areas as behavioral sciences (Shimizu & Ishikawa, 2011), health sciences (Price, Laird, Fox, & Ingham, 2009), education (Sarnacchiaro & D’Ambra, 2012) and sports analysis (Baghal, 2012). When applied researchers not only pay attention to testing hypotheses about an underlying model within a set of observed variables but also focus on within-individual changes in the response variable across time; latent growth curve (LGC) models could be considered.

**Latent Growth Curve Models**

The latent growth curve (LGC) model, represented as a special case of structural equation modeling, includes both components of covariance structure and the estimation of latent means. The LGC model is a type of data analysis used to understand individual differences in both rate and baseline of change. In the LGC model, the same observed variable is repeatedly measured on each individual at the same waves of time (i.e., time points) in the study; this type of data analysis is based on longitudinal data. A longitudinal study is the research design that collects
data from samples at different time points in order to study changes or continuity in the characteristics of samples (Gall, Gall, & Borg, 2007). The unique feature of longitudinal studies is that the same individuals are measured repeatedly through the duration of the study (Fitzmaurice, Laird, & Ware, 2004). In order to assess within-individual change across time, a precise estimate of change could be obtained from longitudinal studies because they have the potential capability to eliminate extraneous variables, which are the factors that influence the response regardless of whether they are measured or not. By comparing individuals’ responses on two or more occasions, irrelevant sources of variability among individuals can be removed from the model where individuals serve as their own control (Fitzmaurice et al., 2004).

Latent variables in the LGC model represent amount and form of change or growth trajectories in the observed variables. Two aspects of change are specified as factors in the basic univariate linear LGC model: a latent intercept and a latent slope. The intercept factor represents an initial level of an outcome usually measured at time zero whereas the slope factor represents the linear rate of change in the measured outcome (Duncan, Duncan, & Strycker, 2006; Preacher, Wichman, & Briggs, 2008). The general LGC model is also part of the structural equation modeling framework with the same structure as in SEM that consists of the measurement and structural model. The measurement part in LGC model links the repeated measures of observed variables to latent growth factors (i.e., latent intercept and latent slope), while the structural part links growth factors to one another (Kaplan, 2002).
Latent growth curve models have been used to model longitudinal data where repeated univariate response measures, over a number of occasions, are observed from a homogenous population. With a single linear growth trajectory, the LGC model allows intercept and slope growth factors, which are random effects, to describe longitudinal changes in the subjects’ outcome variable scores. The intercept and slope growth factors are considered random because they represent a random selection from a population. Mean and variance of the two random effects, i.e., continuous latent growth factors or latent variables, are estimated (Peugh & Fan, 2012). Other random effects estimated in the LGC model are variation in the mean intercept and slope estimates.

As previously mentioned, parameter estimates in the random intercept and random slope model under the LGC model framework, for example, are latent means (i.e., mean intercept and mean slope), latent variances (i.e., intercept variance and slope variance), and covariance between latent variables (i.e., intercept/slope covariance). A set of parameter estimates in the mean vector with parameter $\theta$, $\mu(\theta)$, describe the average initial level parameters and rates of growth for the linear LGC model, which are the mean intercept and mean slope, respectively. The covariance matrix as a function of $\theta$, $\Sigma(\theta)$, contains variances and covariances of the latent factors as well as variances and covariances of the observed variables. The variances of the latent factors reflect the intra-individual variation (or between-person variance) in the initial status and rate of growth (Preacher et al., 2008). The covariance between the latent mean intercept and mean slope reflects the relationship between the initial status (i.e., intercept) and the rate
of growth (i.e., slope) at a given occasion. The LGC model with maximum likelihood estimation requires certain conditions or assumptions to allow the proper interpretation of longitudinal change. The specific assumptions associated with LGC models mostly involve the distribution of latent variables (Preacher et al., 2008).

One weakness in the methodological literature of psychometric modeling with latent variables is that the model has been analyzed as if the data are obtained from a single population; however, it is often possible that individuals in the sample were drawn from a mixture of populations. In many applied studies, such as those conducted on educational achievement, attitudes, opinions, alcohol or drug use, and mental health, the homogeneity assumption of standard measurement models is unrealistic across subpopulations since subsets of the sample may have different backgrounds (B. O. Muthén, 1989). For instance, in adolescent delinquency research, data have been found to come from a mixture of six trajectories: rare offenders, moderate-level chronics, high-level chronics with varying instructional background including poor academic achievement, unsupportive family environments, life events, and substance users (Wiesner & Windle, 2004).

When researchers search for the appropriate technique in order to elaborate more details when analyzing a mixture of populations with regular covariance structure models, conventional growth curve modeling is not suitable for this problem even though it is a powerful tool to investigate differences of individuals over time including both intraindividual or within-person change over time and interindividual or between-person variability in such change. Moreover, the latent growth curve model assumes that growth trajectories are captured from a single
population of individuals. If a mixture of populations exists, but one trajectory is
specified, this may yield results that lack power to detect the relationships held
within any one of the groups. Moreover, the estimated growth trajectory may not
represent all trajectory classes (Eye & Bergman, 2003; Jedidi, Jagpal, & DeDarbo,
1997; Sterba, Prinstein, & Cox, 2007). Therefore, it is reasonable that if the
existence of latent sources of heterogeneity in trajectories (i.e., growth factors) is
assumed, then growth mixture modeling (GMM) should be employed.

**Growth Mixture Models**

The analytical basis of growth mixture modeling to examine the presence of
unobserved heterogeneity in the development between subjects is the finite mixture
model (Bauer & Curran, 2003; B. O. Muthén, 1989). Note that finite mixture
analysis is used to specify a model in which the data come from a mixture of
populations (Everitt & Hall, 1981; G. McLachlan & Peel, 2000). Thus, the basic
assumption of finite mixture models, is essentially a mixture of distributions of the
observed data. The weights of mixture distributions correspond to the relative size
of components. A mixture model for \(k\) classes is equivalent to a mixture distribution
with \(k\) classes; and individuals within the same class should be characterized by the
score representing the characteristics of that class. The focus of the model is
therefore, to examine the magnitude and the direction of the relationships between
a function of the class given that each latent class is defined by its own mean vector
and covariance matrix.

When looking for the change of the individual over time, not only in terms of
the shape (i.e., nature form) but also direction of change (i.e., trajectory),
researchers usually conduct analysis under a longitudinal basis framework using
growth mixture modeling. Growth mixture models are types of longitudinal mixture
models that represents unobserved heterogeneity of individuals in change over time
(B. O. Muthén, 1989; Nagin, 1999). To map hypotheses of development in
heterogeneous subpopulation using GMM, a latent categorical variable is defined as
a mixture of subgroups in the population whose membership is unknown and must
be inferred from the data (F. Li, Duncan, Duncan, & Acock, 2001). For example, in
the study of growth in adolescent alcohol use assessed in four time points
(1976-1979) by F. Li et al. (2001), three hypothesized growth mixture models (i.e.,
$K = 2$, $K = 3$, and $K = 4$) were compared both among themselves and to the LGC
model ($K = 1$) using information criteria-based indices. Based on an inspection of
the growth trajectories and examining model parameter estimates in the study as
well as information criteria-based indices, the three-class model was the best for
representing the pattern of growth in adolescents alcohol use which is a basis for
subsequent analyses.

To be more specific about the kind of population heterogeneity to be
discussed, consider the following related latent variable models to GMM. In the
study regarding regular covariance structure modeling, if population heterogeneity
is assumed and subpopulation groups such as gender are observed, multiple-group
LGC modeling could be applied (Peugh & Fan, 2012; Preacher et al., 2008).
However, if data are randomly selected from a heterogeneous population containing
unobserved subpopulation groups, modeling population heterogeneity could be
conducted in two ways: a latent class growth analysis and growth mixture modeling
Latent class growth analysis is a fixed effect analysis model for which no intercept or slope is estimated, which implies that unobserved latent subpopulations differ only in their growth trajectory means.

Regarding the covariance matrix and random error in latent class growth analysis within each of the growth trajectories, the off-diagonal covariance matrix elements are considered zero, and the variation of all individuals about the growth trajectories mean is considered as random error. The growth mixture model, on the other hand, is a random effects analysis model. The off-diagonal covariance matrix elements are not restricted to zero; they are freely estimated. Moreover, the variation of any individual about the mean growth trajectories is not random error and is one of the parameters estimated in the model.

When considering model complexity, studies by Verbeke and Lesaffre (1996) presented model invariance over groups. The aspects of the model that were held invariant in Verbeke and Lesaffre (1996) study were factor loadings ($\Lambda_k = \Lambda$), a covariance matrix of growth factor ($\psi_k = \psi$), and covariances of the repeated measures ($\Theta_k = \Theta$). Mean levels of growth were only a group difference in such a study. Another approach by Nagin (1999) as well as White et al. (2000) stated that the residual variability in growth was captured by mean growth trajectories for each class (i.e., fixed effect) by constraining the covariance matrix of growth factors to zero (i.e., $\psi_k = \mathbf{0}$).

A theoretical concern in growth mixture modeling is specification of an unobserved grouping variable. Generalizing the multiple-group framework to the case in which group membership is completely unobserved, the grouping variable in
GMM is replaced by the probability of class membership (Muthén & Muthén, 1998). Because the group membership variable in each group is unobserved, the proportion of subjects in each class is unknown. Therefore, the individual probabilities of group membership need to be estimated along with other parameters in the model (Bauer & Curran, 2003).

To introduce an unobserved latent variable in GMM, a latent categorical variable, $z_{ik}$, is considered to represent the unobserved subpopulation membership for individual $i$, $i = 1, 2, \ldots, N$; $k = 1, 2, \ldots, K$. The $K$ components here refer to the number of latent classes, a trajectory class variable, or a latent component. The $K$ latent classes of individuals may be different with respect to their growth factor means, growth factor variances, or both mean and variance components (B. O. Muthén, 2004). Because population heterogeneity is unobserved and the number of latent components must be specified, not estimated, key questions of how many subpopulation growth trajectories, $K$, should be specified, arose among researchers, before data from heterogeneous populations have been analyzed accurately. Fitting several models with different numbers of components was a common way to select the proper number of $K$ latent growth trajectories (Bauer & Curran, 2003). Fit statistics used to identify the appropriate number of latent growth classes included Akaike’s information criterion (AIC), the Bayesian information criterion (BIC), or the consistent AIC (CAIC). Once the numbers of components have been specified, then the growth mixture model could be estimated by maximum likelihood using an EM (Expectation-Maximization) algorithm (B. O. Muthén, 2004). Probability of class membership for each individual as well
as the individuals’ score on the growth factors, such as intercepts and slopes, could be estimated thereafter.

Like SEM and LGC models, growth mixture models require a number of assumptions to be met for the formulation and fitting of the model. In enumeration by Bauer (2007), the following assumptions are needed to be met:

1. The repeated measures have normal distributions conditional on random effects within components and are conditional on any exogenous predictors; the mean, variances, and covariances within components were specified following the literature review or the theory relating to the topic,

2. Exogenous predictors have linear relationships with individual trajectory parameters,

3. Sample observations are independent and are equal in probabilities of selecting sample individuals.

Note that the assumption of missing data patterns which are missing at random is not made in the current dissertation project since no missing data were assumed.

Growth mixture modeling has gained an advantage over other models based on the SEM framework for its capability of exploring and specifying different group-based growth curves in longitudinal data. However, in order to use GMM effectively, researchers should also be aware of theoretical correspondence regarding the following issues as have been shown by some authors (Grimm & Ram, 2009; Grimm, Ram, & Estabrook, 2010). Growth mixture models can be sensitive to
starting values, the possible parameter values specified from the range of start values by the researcher.

1. A reasonable range of possible values for each parameter of interest is assumed to have been defined from guidance in the literature. Different starting values will provide different results for estimating parameters which reflect the sensitivity of GMM (Hipp & Bauer, 2006; G. McLachlan & Peel, 2000; B. O. Muthén, 2004). Therefore, GMM will only yield accurate results if researchers carefully select appropriate starting values recommended by the literature.

2. How and when to include covariates in the model still is an active topic to explore (Bauer & Curran, 2003; Huang, Brcht, Hara, & Hser, 2010; Lubke & Muthén, 2007; B. O. Muthén & Muthén, 2000).

3. The nature of the model requires more exploratory analyses for establishing generalizability and confidence of results.

4. Non-normal distributions in the outcome measure lead to the incorrect interpretation of results obtained by GMM (Bauer & Curran, 2003).

5. There has been some empirical research to determine appropriate numbers of latent class e.g., B. O. Muthén (2004), Nylund et al. (2007), and Tofighi and Enders (2008), but there is still room for additional research to further understand criteria for determining number of classes.

The issue of specifying number of latent class is the focus of the current study.
Specifying Number of Latent Classes

As stated previously, growth mixture models assume that the populations in a longitudinal panel being studied are heterogeneous in the shape of their growth trajectories i.e., each class is described by a distinct set of growth model parameter values (Tofghi & Enders, 2008). A fundamental concern in the application of GMM is to determine the number of components (i.e., latent classes), since generally the true model is not known a priori and the correct number of latent classes ought to be specified before fitting the model. The importance of the number of latent classes to growth mixture models is due to the following: (a) the group identification may not represent the true group if the number of latent classes or components is identified incorrectly (Bauer & Curran, 2003), and (b) the consistently estimated parameters are derived from selecting the correct number of components even if the condition of normality within components is not met (Jedidi et al., 1997).

When conducting analysis of growth mixture models, two major steps are specifying and evaluating model fit. The first step is to specify the number of latent classes. The ways to determine the number of latent classes are inspecting the smallest Bayesian Information Criteria (BIC) among various competing specifications, examining the posterior probability that is assigned to a particular class, and using prior knowledge (L. K. Muthén & Muthén, 2012). The second step is to fit the hypothesized model to define a growth mixture model.

As discussed, growth mixture models are types of applied data analysis procedures used to identify unobserved heterogeneity in a variety of developmental
processes in a population. Although the existence of latent sources of heterogeneity is assumed in GMM, this issue has not been theoretically resolved in the context of extracting the correct number of latent classes when the researcher has no prior knowledge about the exact number of latent classes. Most simulation studies have recommended fitting and evaluating the performance of different fit indices to enumerate the number of latent classes in a GMM analysis. The main theme throughout the current study is to assist applied researchers in providing an applicable method for determining the number of classes for GMM.

**Problem Statement**

As growth mixture model becomes more popular, it becomes increasingly critical for researchers to clarify the enumeration of latent groups (classes). In fact, correctly enumerating the number of latent components is a critical issue in a growth mixture modeling analysis, since classes play an important role for interpreting results and making inferences about growth parameters. The question arises, as to how many unobserved groups exhibit distinct growth trajectories across time? This question presents a particularly challenging problem.

Few studies have applied a set of model selection criteria to guide the decision on the number of latent components in mixture modeling. Study by Nylund et al. (2007) used criteria that rely heavily on likelihood-based indices. Statistical information criteria based on likelihood includes traditional naive chi-square difference (NCS), Lo-Mendell-Rubin (LMR) (Lo, Mendell, & Rubin, 2001), bootstrapping likelihood ratio (BLRT) (G. J. McLachlan, 1987), Akaike’s information criterion (AIC) (Akaike, 1987), Bayesian information criterion (BIC)
(Schwarz, 1978), sample adjusted BIC, consistent version of AIC (CAIC) (Bozdogan, 1987), sample size adjusted CAIC (SACAIC) (Tofighi & Enders, 2008), Draper’s BIC (DBIC) (Draper, 1995), Hannan and Quinn’s information criteria (HQ) (Hanan & Quinn, 1979), and Hurvich and Tsai’s AIC (HT-AIC) (Hurvich & Tsais, 1989). Also, another study, Tofighi and Enders (2008), empirically examined enumeration of the number of latent components in the mixture modeling context. Tofighi and Enders’ study used simulations to examine the performance of fit indices to determine the number of latent classes. Five factors manipulated in Tofighi and Enders’ study were the number of repeated measures including sample size, separation of the latent classes, the mixing percentages, and within-class distribution shape; while Nylund et al. (2007) considered sample size, the number of items in each factor, class probabilities, and number of classes in the population as conditions in their study. Sample size adjusted BIC (SABIC) appeared to perform more consistently than other statistical tests in correctly identifying the number of latent classes under the study of Nylund et al. (2007) and Tofighi and Enders (2008). In studies of finite mixture models, BIC showed the most consistency in selecting the correct model (Jedidi et al., 1997).

Currently, comparison of the performance of model selection criteria is recommended to assist applied researchers in determining the number of latent classes in a GMM context. Based on the results of these commonly used model selection criteria across all modeling settings, identifying the number of classes correctly had previously been somewhat inconsistent, except for BIC in the finite mixture structure model (Jedidi et al., 1997). Jedidi et al. (1997)’s study is an
extension of theoretical research on the finite mixture model by “allowing the mixing proportions to depend on prior information and/or subject-specific variables” (p.57). Interestingly, expanding theoretical research to correctly identify the number of latent classes in finite mixture models corresponds to the concept of the Bayesian method. Although Lee and Song (2003) had incorporated a Bayesian model selection approach for testing the number of latent components for finite mixtures of SEMs, they did not use this method for growth mixture models.

Although using various fit indices is generally reasonable, there has not been common acceptance of the optimum criteria for identifying the number of classes in GMM. Moreover, limitations and pitfalls using those criteria of model fit still exist, since the approaches used in fit indices is a likelihood-based method which is based on asymptotic or large sample theory. This means as more data are obtained from the same underlying process, the distribution of the parameter of interest approaches normality (Gelman et al., 2014). This theory is not robust when sample size is small. More importantly, when conducting statistical inferences under SEM framework; the sample size must be large to ensure accurately estimated parameters (Scheines, Hoijtink, & Boomsma, 1999). Another weakness in the methodological literature results from lack of using previous information to make the best decision on the number of latent classes. Also, applied researchers have to carry out several analyses on different numbers of latent classes and apply the fit indices recommended by the literature to decide the number of classes in GMM analyses. The problem of specifying numbers of latent classes in GMM still persists.
The question as to “how many components should be applied for these data,” still needs to be answered for applied researchers.

Finally, studies on the enumeration of number of components of mixture models had expanded the knowledge of Bayesian analysis through simulation. A study by S. Y. Lee and Song (2003) proposed a Bayesian approach on mixtures of structural equation models with an unknown number of components. The problem was formulated as a problem of model selection by choosing the mixture sturctural equation models with different numbers of components. The Bayes factor was computed applying the idea of data augmentation (Tanner & Wong, 1987). Simulated observations in Lee and Songs study were generated by Gibbs sampling, the sampling technique algorithm under Markov chain Monte Carlo, from the posterior distribution.

To better understand this challenge, techniques based on the Bayesian framework to determine the number of latent classes for mixture models are briefly discussed. For example, reversible jump Markov chain Monte Carlo methods are applied in traditional mixture models (Richardson & Green, 1997), whereas path sampling (Gelman & Meng, 1998) and Bayes factor (S. Y. Lee & Song, 2003) are used in mixtures of structural equation models. As can be seen from these studies, attempts to specify the numbers of components have been made based on other types of mixture models rather than growth mixture models. Determining the number of latent classes in growth mixture models is mostly done empirically, regardless of previous information, unlike in Bayesian methods, which do include prior information. However, the estimation of the number of components on growth
mixture models applying Bayesian methods has not been investigated. There was no empirical evidence of using prior information to help guide applied researchers in correctly identifying the number of latent classes in GMM. The overall purpose of this dissertation is to assist applied researchers in answering this question with added confidence.

**Purpose of the Study**

The current study used a Bayesian approach for specifying the numbers of components in growth mixture models. In this study I aimed to develop a new method for applied researchers to employ when attempting to determine the appropriate number of latent classes in growth mixture model analysis involving heterogeneity in subpopulations in both magnitude and direction. The recommendations are based on different levels of numbers of waves of data collected, overall sample size, and number of latent classes in the population. Markov chain Monte Carlo simulation methods using Metropolis Hastings algorithm were used to develop these guidelines.

Even though incorporating covariates may play an important role in estimating growth factors and provide a better understanding of the data, the role of covairates was beyond the scope of the current study. Recommendations based on findings from the current study may be helpful for GMM class enumeration and estimation on the number of latent classes. As such, from a methodological standpoint, the results from this study could provide some guidance on the estimation of the number of latent classes in a GMM analysis to applied researchers as they attempt to apply GMM to answer their research questions.
Research Questions

The question, “How will the Bayesian method be designed for estimating the number of components on growth mixture models?”, was investigated in this dissertation through the following research questions.

Q1 Could the developed R program using the Bayesian method support researchers to estimate the number of latent classes on growth mixture models?

Q2 How can researchers select the candidate distribution in estimating the number of latent classes on growth mixture models?

Q3 What is the informative prior performance for different values of parameters on estimation for the number of latent classes in growth mixture models?

Lastly, this study provides an alternative method (i.e., Bayesian estimation) with regard to GMM to specify the number of components. A Bayesian method allows inclusion of knowledge of the observed data within the analysis through the prior distribution of unknown parameters. The prior distribution in this dissertation consisted of two types based on whether or not applied researchers possess the information concerning the parameters of interest prior to conducting the research. The product of the prior distribution and the sampling distribution from the data yield the posterior distribution which was the distribution of interest. Then the mean of this distribution was used to estimate the number of latent variables.

Chapter Summary

Chapter I established the background for the current dissertation of change under SEM framework, the advantage of GMM compared with traditional latent
growth curve models, the need for the study, and briefly described the method used for the investigating the number of latent classes in GMM.
CHAPTER II

REVIEW OF LITERATURE

There are established statistical procedures for measuring changes between and within groups over time, but how does one measure the unobserved heterogeneous changes in individuals over time? To answer this question, longitudinal data are used in the context of growth mixture modeling. Growth mixture modeling is one option researchers can apply in order to detect development in each of their subjects. One of the difficulties in a growth mixture modeling analysis is deciding upon the number of latent classes (e.g., unobserved subgroups) that need to be incorporated. The purpose of this dissertation was to assist applied growth mixture modeling researchers in their decision making regarding the numbers of mixture components in growth mixture modeling. This review of literature provides the fundamental concepts, the relevant theoretical background, and the basic understanding of the methods used to support the need and the purpose of the current dissertation. Chapter II begins with an overview of growth mixture modeling, factor analysis, structural equation modeling, model of change, and growth curve modeling.

There are established statistical procedures for measuring changes between and within groups over time, but how does one measure the unobserved
heterogeneous changes in individuals over time? To answer this question, longitudinal data are used in the context of growth mixture modeling. Growth mixture modeling is one option researchers can apply in order to detect development in each of their subjects. One of the difficulties in a growth mixture modeling analysis is deciding upon the number of latent classes (e.g., unobserved subgroups) that need to be incorporated. The purpose of this dissertation was to assist applied growth mixture modeling researchers in their decision making regarding the numbers of mixture components in growth mixture modeling. This review of literature provides the fundamental concepts, the relevant theoretical background, and the basic understanding of the methods used to support the need and the purpose of the current dissertation. Chapter II begins with an overview of growth mixture modeling, factor analysis, structural equation modeling, model of change, and growth curve modeling.

**Overview of Growth Mixture Modeling**

The contemporary approach for modeling change has focused on latent growth curve modeling that permits examination of both intraindividual (within-person) change over time and interindividual (between-person) variability in such change (Preacher et al., 2008). Latent growth curve model approaches assume a single growth trajectory for the entire population. Traditional latent growth curve modeling is a technique used to analyze longitudinal data in which the observed outcome variable is related to time or a time-related variable such as age (B. O. Muthén & Muthén, 2000). Various areas of research such as developmental psychology, cognitive and language development, as well as, health and aging
commonly investigate different subgroups of individuals that follow qualitatively different developmental trajectories over time (Hipp & Bauer, 2006). Other studies have been conducted on GMM. For example, Elliott et al. (2005) identified three distinct classes in the study of patterns of recovery following treatment for chronic post-traumatic stress disorder (PTSD). The groups varied significantly in terms of their characteristics, symptom severity, and improvement over time. Moreover, two and three classes were suggested in an initial exploratory model in the field of academic achievement (Espy, Fang, Charak, Minich, & Taylor, 2009). Results from these empirical studies confirm the theoretical contentions that there exists heterogeneity of growth trajectories in the larger population. Therefore, when samples from a longitudinal study are drawn from a finite mixture of populations, researchers may consider employing GMM. In GMM, the latent class variable captures heterogeneity (i.e., unobserved heterogeneity) in the growth model parameters: intercept and slope.

B. O. Muthén and Muthén (2000) categorized analysis of relationship of characteristics of interest into two types. One type of analysis, known as variable-centered approach, focuses on describing the relationship among variables. The first approach includes regression, factor analysis, and structural equation modeling. The goal is to identify significant predictors of outcomes and describe the relationship between dependent and independent variables. The second analysis, known as person-centered approach, focuses on the relationships among individuals. The second approach includes cluster analysis, latent class analysis, and finite mixture modeling, aimed to classify individuals into different groups based on the
pattern of individual responses which are more similar to each other (homogeneous) in the same group and are different (heterogeneous) from other groups. When latent classes do exist, researchers attempt to identify them using mixture modeling. Mixture models are applied data analyses used to detect unobserved heterogeneity in groups of people when their responses to measured variables are similar (Nylund et al., 2007). As such, the conventional growth modeling approach may not be an appropriate method to study heterogeneous patterns of change.

Growth mixture modeling is a more flexible, alternative approach to traditional growth curve modeling in that GMM relaxes the assumption of a single population and allows parameters to be different across unobserved subpopulations. Growth mixture modeling assumes that individuals are sampled from a heterogeneous population. Thus, GMM can be incorporated into a more general latent variable framework for many reasons (Bilir, Binici, & Kamata, 2008). The first reason is that GMM applies categorical latent variables (i.e., same as latent trajectory classes or latent class indicators) to account for unobserved heterogeneity in the population. Growth mixture modeling is accomplished by considering various classes of individuals around different mean growth curves instead of considering variability of individuals around a single mean growth curve. The second reason is that model parameters, such as continuous latent variable intercepts (i.e., initial status) and slopes, are allowed to vary across latent classes. The third reason is that GMM can be used to investigate the association between parameters of latent classes and other variables, including the relationship between parameters of latent classes and class-specific covariate effects. These accompanying benefits are
assumptions that allow for proper interpretation of longitudinal change within the GMM framework. According to Bauer (2007) the specification and fitting of growth mixture models require the following assumptions: “(a) within-class conditional normality (p. 765), (b) properly specified mean and covariance structure (p. 767), (c) effect of exogenous predictors are linear (p. 768), (d) missing data are missing at random (p. 771), and (e) sampled individuals are independent and self-weighting (p. 773)”. In order to establish the foundation for conducting research under growth mixture modeling, concepts related to growth mixture models such as factor analysis, structural equation modeling, model of change, and growth curve modeling are briefly discussed. A detailed description of growth mixture modeling is presented by extending these theoretical concepts to demonstrate the models utilized and procedures undertaken to estimate all unknown parameters on growth mixture model.

**Factor Analysis**

Researchers in practically every scientific discipline, including biologists, educational researchers, market researchers, medical researchers, psychologists, and social scientists have increasingly focused their interest on studying theoretical constructs that cannot be observed directly. Examples of unobserved variables include self-concept and motivation in psychology as well as verbal ability and teacher expectancy in education. These unobserved variables are represented by measured indicator variables and are termed latent variables or factors (Byrne, 1998). The investigation of relations between sets of observed variables is a fundamental concern to researchers. These observed variables are also called
indicators or manifest variables. Two statistical techniques that can be used to examine covariance relationships among observed and latent variables are exploratory factor analysis (EFA) and confirmatory factor analysis (CFA) (Byrne, 1998).

Exploratory factor analysis is associated with theory development designed to explore the covariance relationship between observed and latent variables. Theoretically, all observed variables within a certain group or factor are at least moderately correlated among themselves but show weak correlations with other variables in a different factor (Byrne, 1998; Johnson & Wichern, 2007). Ideally, these relationships result in exposition of high pattern or structure coefficients on one factor and low on the other factors. Note that pattern or structure coefficients are sometimes referred to by applied researchers as factor loadings. Pattern and structure coefficients reflect the strength of relationship between observed and latent variables. Confirmatory factor analysis, on the other hand, is associated with theory testing and is employed when the number of latent variables, along with corresponding indicators is specified by the analyst (Kline, 2011).

**Structural Equation Modeling**

As an extension of CFA, structural equation modeling is a more general modeling framework that unites factor analysis with simultaneous equation modeling (i.e., path analysis) that allows researchers to investigate complex relationships among factors (Kaplan, 2000). More specifically, researchers across a variety of disciplines use SEM to test the plausibility of a theory regarding causal relationships among latent factors.
Most statistical techniques, including multiple regression and analysis of variance (ANOVA), for instance, apply fundamental statistical concepts in terms of modeling the individual cases. While these standard statistical techniques only analyze observed variables, SEM is used to analyze both observed and latent variables and models covariances rather than individual observations (K. Bollen, 1989; Kline, 2011). Another advantage of SEM corresponds to the explicit incorporation of measurement error in the model, whereas other methods assume measurement without errors in all variables, which is unrealistic in most situations. Error variance is therefore estimated in all SEM analyses. The above features distinguish SEM from most standard statistical techniques.

The term structural equation modeling refers to a family of related procedures extended from regression equations. In SEM, a series of regression equations are simultaneously developed to estimate the relationships among two or more variables based on the analysis of variance-covariance matrices. Since SEM procedures emphasize covariances rather than cases, such modeling techniques are known by various names including analysis of covariance structures, covariance structure analysis, or covariance structure modeling (Bollen, 1989; Kline, 2011). Therefore, SEM researchers have two main goals: (a) to comprehend the pattern of relationships among observed variables in the model and (b) to explain their variability in a hypothesized model (Kline, 2011). Moreover, to ensure accurate estimation and inference, as with all parametric statistical methodology, SEM requires certain underlying assumptions to be satisfied. The major assumptions associated with structural equation modeling include multivariate normality of
random or completely random missing data, sufficiently large sample size, and correct model specification (Kaplan, 2000).

As outlined by (Jöreskog, 1973), structural equation models are comprised of two components: (a) a measurement model and (b) a structural model. The measurement model corresponds to confirmatory factor analysis and identifies the relationships between the latent variables and their observed variables. The structural model corresponds to path analysis and specifies the directional relationships among the latent variables themselves (K. Bollen, 1989; Kaplan, 2000; Kline, 2011; Tabachnick, 2007). The best way to introduce the measurement model (i.e., CFA) is through an illustration with a diagram modified from Duncan, Harmer, Acock, and Stoolmiller (1998). Figure 1 presents a CFA for one latent variable (i.e., a single-factor latent construct) $\xi_1$. The latent variable in this example is hypothesized to have four indicators. The theoretical relations shown in Figure 1 could be represented through the following set of equations in matrix form in Equation 1

$$ y = \Lambda_y \xi + \delta, $$

where $y$ is a vector of random observed variable, $\Lambda$ is a matrix of factor regression weights (loadings) to represent relationships between latent and observed variables, $\xi$ is vector of latent factors, and $\delta$ is a vector of measurement errors (K. Bollen, 1989; Kaplan, 2000).

Consider separately, the set of structural models often presented in the form of a path diagram, especially in the initial stages of model specification. A path
Figure 1. Example of confirmatory factory analysis.

diagram is a schematic representation of causal linear relationships among a number of variables represented by straight arrows (Loehlin, 1987). An example from Bagizzi and Yi (1988) is a model describing the effect of expectancy-value attitude (EV) and past behavior (PB) on attitudes toward an act (AACT) as well as the effect of attitude toward an act on behavioral intentions (BI) as shown in Figure 2.

Figure 2. Hypothesized full structural equation model, $\delta$ and $\varepsilon = \text{error}$. 
For the structural model shown in Figure 2 the full structural model in matrix form is

$$\eta = \beta \eta + \Gamma \xi + \zeta. \quad (2)$$

where \( \eta \) is a vector of latent dependent variables (i.e., endogenous), \( \xi \) is a vector of latent independent variables (i.e., exogenous), \( \beta \) is a matrix of regression coefficients relating the endogenous latent variables to each other, \( \Gamma \) is a matrix of regression coefficients relating latent endogenous variables, similar to dependent variables, to latent exogenous variables, similar to independent variables, and \( \zeta \) is vector of residuals. The hypothesized SEM shown in Figure 2 is

$$x = \Lambda_y \xi + \delta \quad (3)$$

$$y = \Lambda_x \eta + \epsilon. \quad (4)$$

where \( y_1, \ldots, y_4 \) are different items measuring attitude toward an act (AACT), \( y_5 \) is the behavioral intention to give blood (BI), \( x_1 \) is an item measuring the effect of expectancy-value attitude (EV), \( x_2 \) and \( x_3 \) are items measuring the past behavior (PB), \( \delta \) and \( \epsilon \) are vectors of residuals.

In summary, SEM uses a conventional approach to specify the connection between theory and the equations of the model, by taking advantage of the variance-covariance matrix to examine the hypothesized patterns of relationships among a set of observed and latent variables. Structural equation modeling includes both measurement and structural models. The utilities of SEM, as presented thus far, have been based on cross-sectional data which are the data obtained from measuring individuals at one point in time (Kaplan, 2000). A cross-sectional study
design allows researchers to assess between-individual differences, but it does not provide within-individual changes in the response (Fitzmaurice et al., 2004). A longitudinal study, on the other hand, can be used to describe within-individual changes in the response variable. Moreover, given the increased use of complex theoretical models that attempt to examine individuals’ behavior from both developmental and contextual perspectives, new analytical techniques have been identified to study such models. Therefore, the conceptual development for the measurement of change has become the subject of much discussion.

**Model of Change**

The assessment of measurement of change is motivated by questions such as “does change occur on pre-test and post-test scores following an intervention and, is there a consistent pattern of change across the groups?” Answers to questions about change may be obtained by conducting longitudinal studies. Longitudinal panel data consist of the repeated measurements of the response variable on the same individuals at several points in time (Fitzmaurice et al., 2004). The fundamental objective of a longitudinal analysis is to measure change within individuals on a given response variable, and to determine how individuals change throughout the period of the study. Precision and adequate psychometric properties of the measurement of individual change have been proposed and debated in the methodological literature of the behavioral sciences and education for many years (D. Rogosa, Brandt, & Zimowski, 1982).

Conventional approaches, such as the paired t-test, repeated measures analysis of variance (ANOVA), and multivariate analysis of variance (MANOVA)
have been proposed for analyzing change over time (Voelkle, 2007). However, these procedures focus on determining group changes (i.e., mean changes of the entire sample or of subgroups) as opposed to determining individual changes. In the paired $t$-test, repeated measures ANOVA, and MANOVA, variances are partitioned into two components: (a) between-person or interindividual differences which is the variation based on a different variable in the model that explains it, and (b) within-person or intraindividual differences or variation due to differences in individuals themselves for which variation is only partially explained by the model. Consequently, within-person variance is considered as error variance (Voelkle, 2007); however, this error variance may contain useful information about change. Attempts to explain the source of individual change, and variation between and within individuals were examined by Willett and Sayer (1994) by addressing questions such as: Is individual change related to environment, treatment, or training? Are psychological adjustments in children associated with health status, gender, or home background? Do the attributes of the academic programs of each student play a key role in the rates of change in a study? To gain a better understanding of how specific individuals change across time, the methods of growth curve modeling should be considered (Bock, 1979).

**Growth Curve Models**

Theoretical models that attempt to investigate behavior of individuals from both developmental and contextual perspectives have been referred to as growth curve models. The framework of growth curve modeling techniques provides a fundamental alternative regarding how researchers conceptualize and study change.
D. Rogosa et al. (1982) developed the idea of statistically testing individual time path models for the measurement of individual change. Rogosa et al. also presented a simple form of general growth models along with differential equation models for determining the rate of growth or change. Concurrently, D. R. Rogosa and Willett (1985) sketched out similar procedures to perform the correlates and predictors of individual change over time. D. R. Rogosa and Willett (1983), and Willett (1989) demonstrated the reliability of change measurement. Willett and Sayer (1994) also proposed covariance structure analysis to investigate interindividual differences in change. Moreover, Voelkle (2007) provided potential research questions when subjects are measured repeatedly across time; for example, what was the shape and direction of change (i.e., trajectory) for the sample? Can individual growth trajectories be predicted? Is there any variability of individual trajectories for change? If so, what is that variability? Applications of growth modeling applied to longitudinal research designs can be performed to answer these particular questions.

A systematic approach to testing individual differences in growth, developed by D. R. Rogosa and Willett (1985), was designed to investigate the analysis of change: (a) the individual growth curve and (b) rate of growth. The first step in specifying the formulation of change is the construction of the statistical model for individual growth. Using the covariance structure approach to answer research questions about individual change, the method described here was referred to as time-structured data presented by Bock (1979). In the methodology, the sample of individuals must be observed on three or more waves of data. In addition, each individual must be observed systematically over time, that is, spaced occasions
need not be equal, but the number of waves of assessments must be similar in all individuals.

Suppose panel data consist of \( n \) observations and \( t \) measurement occasions or time points. Let \( Y_{it} \) be the vector of an observation on the \( i \)th person \((i = 1, 2, \ldots, N)\) at time \( t \) \((t = 1, 2, \ldots, T)\) where \( t \) is the time of the \( t^{th} \) wave of observations and the time points are the same for all subjects. The psychometric theory that represents the relationship between true and observed score is presented by the simple additive model

\[
Y_{it} = \xi_i(t) + \varepsilon_{it}
\]

where \( \xi_i(t) \) is a mathematical function describing the true score of individual \( i \) at time \( t \) and \( \varepsilon_{it} \) is the random measurement error associated with the observation of individual \( i \) on occasion \( t \). Furthermore, \( \varepsilon_{it} \) is assumed to be independently and identically distributed from a distribution with zero mean and variance \( \sigma_{\varepsilon}^2 \) for all \( i \) and \( t \).

Subsequently, an algebraic function of time is selected to represent each person's actual growth trajectory, \( \xi_i(t) \), after accounting for measurement error, \( \varepsilon_{it} \). In much empirical modeling of individual growth trajectories, when the mechanism driving the growth is unknown, the class of polynomial functions is selected to represent the functional form of the actual growth model. Choice of an appropriate polynomial model is also an important factor. Even though the higher-order model is superior to the lower-order model in terms of goodness-of-fit, the practical significance of parameters is difficult to interpret (Willett, 1989). Thus, when the
portion of the life span is restricted (i.e., limited number of waves), the character of
individual trajectories of growth model is frequently a simple linear quadratic
function of time (Willett, 1989; Willett & Sayer, 1994).

Variants of the growth curve model have been developed under a variety of
names depending on types of research and characteristics of data: (a) hierarchical
linear models (HLMs) or multilevel models (Goldstein, 2011; Raudenbush & Bryk,
2002), (b) random coefficient regression models (De Leeuw & Kreft, 1986; Rovine &
Molenaar, 2000), and (c) latent growth curve models (Duncan et al., 2006).
Moreover, modeling individual growth is required for the measurement of individual
change. One of the alternative types of growth models listed above is the
hierarchical linear model (HLM). In HLMs, two levels of individual change, which
are the within-subjects or level-1 model, and the between-subjects or level-2 model,
can be examined. An important advantage of hierarchical linear modeling
procedures is the ability to include both time-invariant and time varying covariates
in the model. Thus, subjects in HLMs procedures need not be measured at the
same time points. Another feature making HLMs especially useful in longitudinal
research is that HLMs can estimate change on individuals across time whereas
traditional approaches estimate average change of a population across time. The
second statistical model of individual change over time is the regression model with
random coefficients, which is a special case of growth curve models. As pointed out
by Swamy (1970), interindividual differences observed in a cross-sectional sample
cannot be explained by a regression equation with a few independent variables
unless researchers treat the coefficient vector of a regression model as random.
These random coefficients could account for interindividual heterogeneity in the model. Given the same attributes as in HLMs, the random coefficient regression models allow parameters associated with an individual growth curve (e.g., intercept and slope) across subjects (level-1 unit) to vary randomly with no prediction of variation across subjects. In the random coefficient regression model, the variability in the regression coefficients of both intercept and slopes across between-subject units (level-2) are allowed to be estimated. Finally, when empirical researchers inquire about individual change over time including whether change differs within persons (intraindividual) and whether changes are different between persons (inerindindividual), the analysis of longitudinal data should be conducted. Further analysis within the SEM framework to assess the growth and change over time on latent variables has been proposed by many authors (Duncan & Duncan, 2004; Hancock et al., 2001; Preacher et al., 2008). The growth curve model including the latent variables in the model is called latent growth curve modeling.

**Latent Growth Curve Models**

Researchers have demonstrated the changes of individuals over time based on a longitudinal research design using LGC modeling. Latent growth curve modeling is considered an advanced statistical method, which enables scientists to perceive and understand the shape and direction of change in the latent variable (Preacher et al., 2008). According to Preacher et al. the shape and direction of change is also called a trajectory. The fundamental assumptions of LGC modeling are that (a) the pattern of change is systematically related to time, (b) at least three waves of data are required, (c) the subjects are measured at the same intervals of time, and (d)
the multivariate data are normal. In a basic LGC model, repeated measures, y, are observed for some outcome variable at a number of occasions. The latent variables represents pattern of change in y. These factors (latent variables) are referred to as the intercept factor and slope factor.

Suppose that there are N subjects and T measurement occasions or time points in a longitudinal study. Let \( y_i \) be a \( T \times 1 \) random vector \( y_i = y_{i1}, y_{i2}, \ldots, y_{iT} \) where \( y_{it} \) is the outcome or observation of individual \( i \) on occasion \( t \) (\( i = 1, 2, \ldots, N; t = 1,2, \ldots, T \)), and let \( \eta_i \) be a vector containing \( q \) continuous latent variables. A latent growth curve model could be written in the form of outcome \( y_i \) related to the latent \( \eta_i \) as

\[
y_i = \Lambda \eta_i + \varepsilon_i
\]

(6)

where \( \Lambda \) is a matrix containing latent factor scores (i.e., intercept and slope), and \( \varepsilon_i \) is a vector of residual or measurement errors. The matrix \( \Lambda \) and the vector \( \eta_i \) determine the growth trajectory of the model (K. A. Bollen & Curran, 2006). For example, when \( q = 2 \), and \( \eta_i = (I_i, S_i)' \), the corresponding model represents a linear growth model where \( I_i \) is the latent random intercept and \( S_i \) is the latent random slope for \( i^{th} \) subject, a matrix of factor landings can be expressed as

\[
\Lambda = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 0 & 1 & 2 & \ldots & (T - 1) \end{bmatrix} '
\]

(7)

The relationship among patterns of change and other variables can be specified by utilizing the full structural model specifications shown in Equation 2.
The vector $\eta_i$ in Equation 8 is then expressed as a function of a linear combination of endogenous and exogenous latent variables (Coffman & Millsap, 2006). This full structural model is sometimes referred to as the latent variable growth model, and can be employed to examine the individual components of the growth model. The LGC model without the vector of observed variables, $y_i$, is considered to contain only exogenous predictors (i.e., latent independent variables) and can be written as

$$\eta_i = \Gamma \xi_i + \zeta_i$$

where $\eta_i$ is the unobservable latent vector, $\Gamma$ is a matrix of factor loadings on the exogenous latent predictors, $\xi_i$, is the vector of exogenous latent predictors, and $\zeta_i$ is the vector of disturbance terms. Note that endogenous variables are not examined in the LGC model (Coffman & Millsap, 2006).

To understand the matrix $\Lambda$ for quadratic growth curve models, the first partial derivatives of the hypothesized growth function with respect to each growth parameter can be derived as shown in Equation 9. For the growth function of time $t = 0, \ldots, T$ with parameter $\theta$, $f(t, \theta)$, the hypothesized function is

$$\hat{y}_{it} = \theta_1 + \theta_2 t_{it} + \theta_3 t_{it}^2$$

The first derivatives of the function represented in Equation 9 with respect to the first parameter ($\theta_1$) corresponds to the first column of $\Lambda$, which represents the intercept of the function $\frac{\partial \hat{y}_{it}}{\partial \theta_1} = 1$. The first derivatives of the same function but with respect to the second parameter ($\theta_2$) corresponds to the second column of $\Lambda$, which represents the linear function $\frac{\partial \hat{y}_{it}}{\partial \theta_2} = t$. The last of the first derivatives of this function with respect to the third parameter ($\theta_3$) corresponds to the third column of
\( \Lambda \), which represents the quadratic function \( \frac{\partial^2 y_{it}}{\partial \beta^2} = t^2 \). Therefore, a linear LGC model with four time points for each \( i^{th} \) individual is presented in expanded matrix notation as

\[
\begin{bmatrix}
y_{i1} \\
y_{i2} \\
y_{i3} \\
y_{i4}
\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} I_i \\ S_i \end{bmatrix} + \begin{bmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \varepsilon_{i3} \\ \varepsilon_{i4} \end{bmatrix}
\]

(10)

Moreover, when \( q = 3 \), \( \eta_i = (I_i, S_i, Q_i)' \), the corresponding model represents a quadratic growth model, where \( I_i \) is the latent random intercept, \( S_i \) is the random slope, and \( Q_i \) is a latent random quadratic term for the \( i^{th} \) individual, which can be seen as

\[
\Lambda = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 0 & 1 & 2 & \ldots & (T - 1) \\ 0 & 1 & 4 & \ldots & (T - 1)^2 \end{bmatrix}'
\]

(11)

A quadratic growth model with four time points for a latent random quadratic trajectory for individual \( i \) is presented in matrix notation in Equation 12 as

\[
\begin{bmatrix}
y_{i1} \\
y_{i2} \\
y_{i3} \\
y_{i4}
\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} I_i \\ S_i \\ Q_i \end{bmatrix} + \begin{bmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \varepsilon_{i3} \\ \varepsilon_{i4} \end{bmatrix}
\]

(12)
The observed variables $y_{i1}$, $y_{i2}$, $y_{i3}$, and $y_{i4}$ are measured at time points 1, 2, 3, and 4, respectively. These variables are treated as indicators of the growth trajectory (i.e., latent intercept and latent slope). The intercept in growth models represents subjects' threshold level on the outcome variable at the first wave of data collection. The factor loadings in the $\Lambda$ matrix specify the nature of the trajectory. The coefficients in the $\Lambda$ matrix shows the connection of intercept and slope factors to the repeated measurement of the outcome variable. The first column of $\Lambda$ is fixed to a value of 1 to show that there is the same influence of all observed measurements in each time point on the intercept. The remaining columns contain the values of the time metric; for example, the second column represents linear growth, and the third column specifies a quadratic component (Preacher et al., 2008). The factor loadings in LGC models are usually fixed and justified on the basis of theory and the time intervals of data collection, while factor loadings from standard CFA and SEM models are estimated (Duncan et al., 2006). In most applications of LGC models, an intercept factor and a slope factor, either linear or quadratic, are specified to account for the correlations among the measured variables in a LGC model (Coffman & Millsap, 2006). Basically, the estimated parameters in the LGC model are the mean of the latent variables (i.e., intercept and slope mean); the variance among latent variables (i.e., intercept and slope variance); the covariance among the latent variables (i.e., the covariance between intercept and slope); and the error variances of the measured variables (i.e., variances from the data) (Coffman & Millsap, 2006; Duncan & Duncan, 2004; Preacher et al., 2008).
The diagram notation corresponding to a latent growth curve model is presented in Figure 3. Four time points of data collection are examined in this example. As displayed in Figure 3, latent variables are represented as circles, and rectangles represent observed variables. All intercept loadings in Figure 3 are fixed to 1 and defined as an initial status factor. The latent slope is the fixed effect estimate of the slope of height between time points. Assuming the growth factor is linearly related to time, slope loadings start at 0 in the first time period with the other three slope loadings fixed to 1, 2, and 3. The possible observed measures of height collected at each age at time points 1, 2, 3, and 4, are $y_{i1}$, $y_{i2}$, $y_{i3}$, and $y_{i4}$, respectively. Single-headed arrows indicate regression weights and double-headed arrows indicated variances or covariances.

Figure 3. Latent growth curve model.
In conclusion, latent growth curve modeling is a versatile method capable of examining both intraindividual (within-person) change and interindivdual (between-person) variability in change over time, also known as rates of growth (Preacher et al., 2008). The conventional LGC model treats the observed growth trajectories as sampled from a single homogenous population of individuals characterized by a single rate of growth. However, it is possible that some samples, drawn from a finite mixture of populations, may have their own growth trajectories. For example, there may be very different classes of reading growth in a sample of children from certain populations. Some of the children may have had accelerated rates of growth in reading, some may have shown normal rates of growth in reading, while still others may have exhibited slow or below average rates of growth in reading. If heterogeneity of growth in populations is ignored, parameter estimates and estimates of growth could be biased (Duncan et al., 2006). Therefore, researchers may consider employing growth mixture modeling which relaxes the assumption of homogenous growth in the population (Kaplan, 2002).

**Mixture Modeling**

Latent variable modeling, as mentioned previously, can be discussed in the context of continuous and/or categorical variables, as well as cross-sectional analyses or longitudinal data analyses. The methodology requiring continuous latent variable in the analysis (LGC modeling), and the method using categorical latent variables in a latent structure model (latent class analysis) have been discussed. Additionally, the methodology that uses a combination of continuous and categorical latent
variables in the analysis of longitudinal data is \textit{growth mixture modeling}. The latent variable modeling general framework is summarized in Figure 4.

\begin{figure}[h]
\centering
\includegraphics[width=0.7\textwidth]{general_modeling_framework}
\caption{General Modeling Framework.}
\end{figure}

As displayed in Figure 4, the first circle, labeled A, represents the framework of continuous latent variables, \( \eta \), and variables in boxes, \( y_1, \ldots, y_4 \), indicates
continuous repeated measures over four time points of a univariate $y$ variable. For example, the observed variables of student achievement in Grades 9, 10, 11, and 12 are denoted as $y_1$, $y_2$, $y_3$, and $y_4$, respectively. The first framework includes exploratory and confirmatory factor analysis, structural equation modeling (SEM), and latent growth curve modeling. The second circle labeled $B$ represents a framework of categorical latent variables, $z$, and variables in boxes, $u_1, \ldots, u_4$, refer to binary outcomes of four observed variables obtained by considering whether or not the individual is involved. Since group membership is unobserved in the case of multiple groups in the conventional model (labeled A), using the modeling framework (labeled B) provides flexible modeling by allowing group differences in any of the parameters. Given that the latent class variable is categorical and unobserved, a person is classified into a particular class by the highest posterior probability of membership in the different latent classes (B. O. Muthén, 2001). The second framework includes latent class analysis. The third circle, labeled C, represents a framework of a combination of continuous and categorical latent variables. The third type of mixture model framework is commonly used in growth mixture modeling as well as factor mixture models. The bottom part of Figure 4, labeled D, shows a growth mixture modeling with four repeated measures, $y_1, \ldots, y_4$, representing the continuous repeated measure outcomes. The particular diagram of growth mixture model assumes linear growth; therefore, the model depicts two continuous latent growth factors, the intercept ($i$) and slope ($s$), which are considered the growth parameters. The latent class variable, $z$, is a categorical variable indicated by the growth parameters. The value of $z$ could be 0 or 1.
depending on whether or not it belongs to that class. The directional arrow from the latent trajectory classes to the growth factors indicates that the growth factor might vary across the latent classes $z$. The growth factor from the bottom part of the diagram means, in application, that each growth trajectory could be different across the classes of growth in the population. Lastly, the number of observed items and the number of classes are not specified in this general diagram.

**Latent Class Analysis**

Latent class analysis (LCA) is a measurement theory used to categorize classes of individuals often based on characteristics measured by a set of binary outcomes. The unobserved groups of individuals in LCA are referred to as latent classes (Clogg, 1995; Duncan et al., 2006; Kaplan, 2002). There may be other types of outcomes variables in LCA such as ordinal, nominal, count, continuous, or any combination thereof that needs particular types of statistical software for analyzing. The purpose of LCA is to (a) estimate the number and size of the latent classes in the mixture, (b) estimate the response probabilities for each indicator given the latent class, and (c) assign latent class membership to individuals in the population (Duncan et al., 2006). Latent class analysis applies the concept of conditional item probabilities corresponding to the item parameters to specify the probability of an individual in each class. The values of these probabilities are called posterior probabilities (B. O. Muthén & Muthén, 2000). Moreover, class probability parameters are used to identify the size of each class. With the categorical latent
variable \( z \) taking on \( K \) classes \( (z = k, k = 1, \ldots, K) \), the marginal probability for item \( u_j = 1 \) has the form

\[
f(u_j = 1) = \sum_{k=1}^{K} f(z = k) f(u_j = 1 | z = k)
\]

(13) 


Latent class analysis allows for inclusion of covariate information that helps explain the structure of the latent classes (B. O. Muthén, 2001; B. O. Muthén & Muthén, 2000). For example, covariates related to age, gender, and ethnicity were applied to antisocial behavior classes in the study of B. O. Muthén and Muthén (2000). To express the probability that individual \( i \) is assigned in class \( K \) of the latent class variable, a multivariate logistic regression approach can be applied. Therefore, analyzing LCA yields class probabilities that could be used to classify individuals into the latent classes (B. O. Muthén, 2001). The analysis that combines the mixture of Gaussian mixture models and latent class models is

*finite mixture modeling* (Pearson, 1894; G. McLachlan & Peel, 2000).

**Finite Mixture Modeling**

The group-based methodology for measuring and explaining individual-level heterogeneity in developmental trajectories has been called latent growth curve analysis (Meredith & Tisak, 1990; Willett & Sayer, 1994). The group-based method has also been called a person-based approach to capture heterogeneous grouping of developmental trajectories commonly found in longitudinal data sets (Nagin, 2005). In the past decade, group-based methodology has become a centerpiece of longitudinal data analysis to help developmental psychologists understand both
normal and pathological development (Cloninger, 1986; Holyoak & Spellman, 1993; Kochanska, 1997).

Therefore, the group-based statistical models have had two defining features: (a) the predicted trajectory of each group, and (b) the probability of membership of each such group. The goal of the analysis is to identify groups or clusters of individuals with similar trajectories. For example, a finite set of different polynomial functions of age or time could be considered to summarize individual differences in trajectories in the group-based method. The statistical analysis in a group-based framework is a powerful device for summarizing complex sets of longitudinal data (Anderson, 1980). By grounding the statistical analysis in a small number of groups, individual-level heterogeneity in developmental trajectories could be defined by its size and shape. The size of group is expressed in terms of its proportional representation in the population and the shape is presented in a graphical form. Technically, an application of statistical methods for a group-based trajectory is referred to as finite mixture modeling (Nagin, 2005). A commonly used theory to determine if subgroups exists within a population that follows distinct developmental trajectories (i.e., heterogeneous subpopulation) is finite mixture modeling. Historically, finite mixture models have been used to test hypotheses about population heterogeneity (Hosmer, 1973; G. McLachlan & Peel, 2000). The basic assumption of finite mixture modeling is a mixture distribution; for example, mixture of normal distributions, mixtures of exponential and other continuous distribution, and mixtures of discrete distributions (Everitt & Hall, 1981; G. McLachlan & Peel, 2000).
Mixture distributions are the mixing of proportions or weights of two or more component densities of mixture. Note that there is a mixing proportion (i.e., class probabilities or weight) for each latent class. In this context, $Y$ is being used to represent the entire sample and denotes as $Y=[Y'_1, Y'_2, ..., Y'_N]$, and $Y_1, ..., Y_N$ denotes a random sample of size $n$. In this context, $Y_i$ consists of the random variables corresponding to $p$ measurements of some characteristics under the study for individual $i$. To set up the basic finite mixture model, let $y=(y'_1, y'_2, ..., y'_N)$ denote an observed random sample, where $y_i$ is the observed value of the random vector $Y_i$ for $i = 1, ..., N$ cases. The probability density function of the $p$-dimensional random vector, $Y_i$, can be written in general form as

$$f(y_i) = \sum_{k=1}^{K} \pi_k f_k(y_i)$$  \hspace{1cm} (14)$$

where $f_k(y_i)$ is the probability density function of the random vector $Y_i$ under a $K$-component mixture model, the mixing proportions are denoted as $\pi_k$ satisfying $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$. As $f_1(y_i), \ldots, f_k(y_i)$ are densities, they are also called the component densities of the mixture (G. McLachlan & Peel, 2000). The number of components, $k$, in the finite mixture model for this formulation is considered fixed. However, in most applications, the number of components is unknown. The observational data along with the mixing proportions as well as the parameters specified in the model for component densities can be used to infer the number of components.
To illustrate the basic finite mixture model, a mixture of two univariate normal components with variance $\sigma^2$, means $\mu_1$ and $\mu_2$, and proportions $\pi_1$ and $\pi_2$ are displayed in Equation 15

$$f(y_i) = \pi_1\phi(y_i; \mu_1, \sigma^2) + \pi_2\phi(y_i; \mu_2, \sigma^2),$$

where $\phi(y_i; \mu_i, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{1}{2}\frac{(y_i - \mu_i)^2}{\sigma^2}}$ denotes the univariate normal density with parameters $\mu$ and $\sigma^2$. Some parametric family could be applied to specify the component densities. For example, in the mixture model, the component densities are specified as $f_k(y_i; \theta_k)$, where $f_k(y_i; \theta_k)$ is the existing form for the $k^{th}$ component in the mixture, and $\theta_k (k = 1, \ldots, K)$ is the vector of unknown parameters.

Conceptually, the formula for the mixture density or marginal density $f(y_i)$ is given by

$$f(y_i|\psi) = \sum_{k=1}^{K} \pi_k f_k(y_i|\theta_k),$$

and

$$\psi = (\pi_1, \pi_2, \ldots, \pi_{k-1}, \theta')',$n

where $\psi$ represents the vector containing all unknown parameters, and $\theta$ is the vector containing the $\theta_1, \ldots, \theta_K$ model parameters (G. McLachlan & Peel, 2000).

Additionally, in an SEM framework, there exists a model called growth mixture modeling (GMM) that combines conventional growth curve modeling with latent class analysis under the assumption that a finite population defined by unique trajectory classes exists (Kaplan, 2002; B. O. Muthén, 2001). However, in the GMM framework, different fixed-effect parameters and different random-effect parameters are added following a mixture of $k$ latent growth curve models.
Growth Mixture Modeling

Everitt and Hall (1981) and G. McLachlan and Peel (2000) approached mixture models from a finite distribution perspective whereas Clogg (1995) introduced mixture models in the context of latent class models that were comprised of a subset of the general class of latent structure models such as factor analysis models, covariance structure models, and latent trait models. Given the relevance of examining heterogeneity in longitudinal research, the growth mixture model referenced in this study has been proposed by B. O. Muthén (2001) who extended the LGC model by combining categorical and continuous latent variables in the same model.

A general growth mixture model captures unobserved heterogeneity in the sample on the basis of two important forms of heterogeneity: (a) different individuals which belong to different subpopulations can be accommodated by the conventional growth curve model, and (b) different growth trajectories can be captured by changing of growth factor means over the latent classes (i.e., class-varying random coefficient means) (Duncan et al., 2006). The extended form of GMM is the inclusion of a covariate in the model, since the values of latent growth parameters in each class could be influenced by a covariate. As a result, a function of a set of covariates can be further extended to estimate class membership probabilities in the model.
Model Specification

As mentioned previously, GMM allows different growth trajectories for different classes. The capability of GMM captures two important forms of heterogeneity: (a) heterogeneity in individual growth through the specification of the conventional growth curve model (i.e., LGC model) and (b) heterogeneity representing classes of growth trajectories (Kaplan, 2002). The specification of GMM is similar to the model given for a conventional growth curve except for allowing different growth trajectories for different classes in GMM. To reflect the presence of trajectory classes, the equations to specify heterogeneity in a LGC model can be seen as

\[ y_i = \Lambda_y \eta_i + \varepsilon_i, \]  
\[ \eta_i = \alpha + \beta \eta_i + \zeta_i, \]  

(18)

(19)

(F. Li et al., 2001), where \( y_i \) is a vector of repeated measure continuous outcomes for subject \( i (i = 1, \ldots, N) \) across time \( t (t = 1, \ldots, T) \). The \( i \) subscript indicates that the parameters are allowed to be different across individuals. The \( \Lambda_y \) term is a \( T \times q \) matrix of factor loadings containing \( T \) rows of number of time points and \( q \) columns of number of trajectory classes (i.e., number of latent factors, for example \( q = 2 \) represents a linear growth factor, \( q = 3 \) represents a quadratic growth factor). The \( \eta_i \) term is a \( q \times 1 \) vector of growth factors which is a vector of continuous latent variables that has \( q \) elements. The term \( \varepsilon_i \) is a vector of residuals. The term \( \alpha_k \) is the vector of mean parameters of growth factors, \( \beta \) is a matrix of coefficients
relating growth factors (i.e., latent variables) to one another. Finally, the term $\zeta_i$ is a vector of residuals.

Next in GMM is the specified latent class variable $Z_i$. Let $Z_i$ denotes a categorical random variable for $i = 1, \ldots, K$ with probabilities $\pi_1, \ldots, \pi_k$, respectively. The class membership $Z_i = k$ when $y_i$ come from the $k^{th}$ mixture component, and the conditional density of $y_i$ given $Z_i = k$ is denoted by $f_k(y_i)$, i.e.

$$f(y_i|Z_i = k) = f_k(y_i), \quad (20)$$

In the interpretation of a mixture model mentioned previously in Equation 23, if the density of $Y_i$ has the $k$-component mixture form, the marginal density or its unconditional density of $Y_i$ is given by $f(y_i)$.

The value $z_1, \ldots, z_n$ of $Z_1, \ldots, Z_n$, where $z_i = (z_{i1}, z_{i2}, \ldots, z_{ik})$ represents a vector of unknown parameters for the $k^{th}$ mixture class, can be viewed as the component level or a class membership indicator vector of $y_i$ for the $i^{th}$ individual. Each element of $z_i$ is theoretically assumed to follow a multinomial distribution with the value of either 0 or 1 for the $k^{th}$ element of $z_i$ depending on whether $y_i$ come from the $k^{th}$ class or not.

A multinomial distribution consisting of one draw on $k$ categories with probabilities $\pi_1, \ldots, \pi_K$ (G. McLachlan & Peel, 2000),

$$f(z_i) = \pi_1^{z_{i1}} \pi_2^{z_{i2}} \cdots \pi_K^{z_{ik}}, \quad (21)$$

and denoted as

$$z_i \sim \text{Mult}(1, \pi_1, \ldots, \pi_K).$$
Applications of the Model

Growth mixture modeling is widely used in different disciplines involving a variety of substantive issues (Nylund et al., 2007; Peugh & Fan, 2012). Many studies have presented application of GMM in applied studies while some studies have examined the issue relevant to GMM through simulation study.

**Applied studies in growth mixture modeling.** There is longitudinal research using growth mixture models in applied studies concerned with social behavior and problematic matters such as substance use and criminal behavior as well as job-loss. For example, Greenbaum and Dedrick (2007) analyzed changes in alcohol and marijuana use and the use of drug and alcohol treatment services with patients with serious emotional disturbance using growth mixture model. F. Li, Barrera, Hops, , and Fisher (2002) also examined heterogeneity in the developmental trajectories of alcohol use in adolescents and the accompanying trajectory-specific longitudinal influence of exposure to their deviant peers. Furthermore, L. C. Liu, Hedeker, Segawa, and Flay (2010) evaluated longitudinal preventive intervention trials on subgroups characterized by different types of growth trajectories. Liu et al. presented an application of GMM to ordinal-scale drug-use outcomes. Kreuter and Muthén (2008) also illustrated the analysis of criminal trajectory profiles using GMM.

Moreover, some researchers employing the GMM approach have conducted medical research to study the patterns of treatment response among patients. Elliott et al. (2005) attempted to differentiate groups of individuals who exhibited different patterns of recovery following treatment for chronic post-traumatic stress disorder (PTSD). Peer and Spaulding (2007) also utilized GMM to explore heterogeneity in longitudinal psychological recovery within an intensive psychiatric rehabilitation program for people with
schizophrenia and related severe mental illness. Also, application of latent class growth models and growth mixture models with the same longitudinal research were conducted by Reinecke and Seddig (2011). The authors specified a quadratic growth curve to account for non-linear change in self-reported delinquency behavior of adolescent from the German panel study. The sample was collected and analyzed from five waves from childhood to late adolescent (age 13-17). Comparisons of different classes of both latent class growth models and growth mixture models were performed using Bayesian information criterion (BIC). According to the smallest BIC, the best model for latent class models included six classes whereas the best model for growth mixture modeling had only four classes. Latent growth curve models and GMM are used by researchers to identify groups of similar individuals based on their growth trajectories and intercepts. Both types of models are used in an iterative procedure under the condition without variance parameters estimated in LGC model. For this procedure, LGC models with no variance parameters are estimated first and the resulting parameter values are entered as starting points for a growth mixture model in order to increase the likelihood that the growth mixture model could be successfully estimated. Fit indices are compared for these models. Comparison of the fit indices indicates that the estimation of individual variability, which is based on the variance parameters in GMM with two latent classes, are reported (i.e., higher psychosocial functioning [HPF] and lower psychosocial functioning [LPF]) and are used in the subsequent analysis. The intercept, slope, and quadratic parameters are also estimated for both latent classes. Using GMM analyses in the study by Reinecke and Seddig (2011) shows the heterogeneity in the type of change trajectory: linearity in the LPF group and nonlinearity in the HPF group.
To summarize, this subsection presented use of growth mixture models in applied research. A summary of relevant empirical literature to real data conditions highlighted what happened or how to perform GMM analysis especially when the numbers of latent classes were unknown. Researchers need to find out the most suitable numbers of components using various fit indices to help them making their decisions. Therefore, the present dissertation project focused on the estimation of the numbers of latent classes to fill the need of those applied researchers.

**Simulation studies in growth mixture model.** Currently, there are dozens of simulation studies on growth mixture models. Some researchers have conducted GMM studies using Monte Carlo simulation. Hipp and Bauer (2006) presented a small-scale Monte Carlo simulation to provide an initial indication of a model with too few or too many latent classes and to estimate GMM assuming within-class normality with ordinal data. The results revealed that models with more classes which also permitted random effects within classes converged less frequently. Similarly, some researchers explored the impact of ignoring a higher level nesting structure in multilevel GMM (MGMM) (Q. Chen, Kwok, Luo, & Willson, 2010). Several aspects of the results were considered in the study by Q. Chen et al. (2010), for instance, hit rate, fixed effect estimates, variance estimates, standard error estimates, and statistical power needed to detect significant variance. The authors concluded that higher level nesting structures were ignored in MGMM.

Assessing the effects of covariates was also discussed in GMM. In one study, the effects of covariates on performance of identifying distinctive trajectories of delinquent behavior during the middle adolescent years were examined (Wiesner & Windle, 2004). In another study, the influence of including a covariate on GMM was used to examine patterns of days of heroin use among heroin users (Huang, Brcht, Hara, & Hser, 2010).
The data used in both studies were longitudinal. In the first study, a 4-wave panel study was used with adjustment problems, poor academic achievement, negative life events, and unsupportive family environment as covariates. As for the second study, a 16-year period of data was applied and three subject characteristics such as ethnicity, early onset of heroin use, and early onset of alcohol use were examined as covariates. Moreover, there are a variety of simulation studies that examine the issue of identifying the correct number of latent classes in growth mixture models with different true latent classes assuming and the indexes used.

Studies Examining Unknown Number of Latent Classes

Growth mixture modeling is superior to the conventional LGC model in that GMM uncovers unobserved heterogeneity in a population to classify substantively profound groups of individuals (B. O. Muthén, 2004). However, implementing solutions to identify the correct number of classes, parameter estimation, model evaluation, and model selection in GMM is not straightforward (Preacher et al., 2008). One of the most challenging tasks in a GMM framework is to extract the correct number of classes. Moreover, it has been suggested by B. O. Muthén (2004) that the influence of including covariates on GMM may play an important role in determining the number of classes (Huang et al., 2010). There is a variety of research that explored the issue of deciding the number of classes in SEM, LGC modeling, and GMM using Bayesian and non-Bayesian approaches.

For non-Bayesian approaches, Nylund et al. (2007) conducted a Monte Carlo simulation to examine the performance of three likelihood-based tests (i.e., naive chi-square [NCS], Lo-Mendell-Rubin [LMR], and bootstrap likelihood ratio test [BLRT])
and the traditionally used information criteria such as Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC). These criteria are used for determining the number of classes in mixture modeling. Nylund et al.'s study focused on three types of mixture models: latent class analysis (LCA), a factor mixture model (FMA), and growth mixture modeling (GMM). Three levels of sample size were used in their study, specifically \( N = 200, 500, \) and \( 1,000 \). All model populations for this simulation study were defined by the number of items, item probabilities, or class means and the number of classes in the population. Nylund et al. reported that BIC was the most consistent fit index for identifying the number of latent classes, but the BIC showed sensitivity to small sample size regardless of the type of model. Tofghi and Enders (2008) investigated the number of latent classes in GMM using the same criteria as in Nylund et al.'s study including nested model likelihood ratio tests (i.e., LMR), goodness of fit measures (i.e., multivariate skewness test [MST] and multivariate kurtosis test [MKT]). Larger sample sizes such as 400, 700, 1000, and 2,000 were applied in Tofghi and Enders study. The number of repeated measures (i.e., time points or waves of data) took on the values of 2, 4, and 6 in Tofghi and Enders’ study. There is no evidence that prior research has included investigation of small sample sizes to estimate the number of latent classes in growth mixture modeling; this could be a gap that needs to be addressed in the area GMM.

The technique to specify numbers of components or latent classes on growth mixture model, mentioned previously, applied non-Bayesian techniques. However, some studies have proposed the method of using Bayesian estimation to determine the number of components (i.e., latent classes). Steele and Raftery (2009) addressed the aspect of priors for the component density parameters. The authors mentioned uniform Dirichlet and the Jeffrey’s prior on the mixing parameters while a Poisson prior has been used to
determine number of latent classes. Nobile and Fearnside (2005) proposed Bayesian analysis using a Markov chain Monte Carlo method for the analysis of finite mixtures with an unknown number of latent classes. The mix of the distributions considered in their study was normal mix, mixture of multivariate normal, and mixtures of uniform distributions. The technique used to sample from the joint posterior distribution of number of components (k) and the class membership indicator (z) were hybrid techniques which included both Gibbs sampling and the Metropolis-Hastings algorithm. The method presented by Nobile and Fearnside can be applied to any family of mixture of latent classes’ parameter estimates.

The conditions manipulated in the previous study (e.g., Nylund et al. (2007) and Tofighi and Enders (2008)) to determine the correct number of latent classes for both Bayesian and Non-Bayesian study were applied to the current project. The details of each condition such as sample size, the true latent classes assumed, and the values taken for the time points were expressed in the methodology of the current dissertation.

**Bayesian Methods**

Bayesian methods have become well established among both statisticians and empirical researchers (Poirier, 2006). Bayesian inference has been referred to as the process to fit a probability model to a set of data and summarize the result by a probability distribution on the parameters of the model (Gelman et al., 2014). A Bayesian approach provides the following advantages in comparisons to frequentist statistics: (a) analysis provides obvious and clear assumptions, (b) probability statements about the unknown parameters can be made or changed over time rather than fixed, (c) statistical models incorporate both the sample information and prior information on the parameter estimates, and (d) the analysis allows researchers to model a wide class of data types.
leading to the key point of Bayesian estimation in the analysis of new types of models (Gill, 2008; B. O. Muthén & Asparouhov, 2012).

Focusing on distributional inferences led to two key assumptions for Bayesian analysis. First, the sampling distribution or objective information describes the distribution of the data given a parameter value. Particularly, this is the basic set-up of a likelihood function. Second, since unknown parameters are treated as random variables rather than fixed, they are assumed to have their own distributions. The distribution is referred to as prior distribution. The proportion of the product of the prior distribution of the parameters and the likelihood function of the sample data provides the posterior distribution which could be used to obtain parameter estimates through Bayesian inference. The likelihood function, the prior distribution, and the posterior distribution are discussed briefly below.

**Why Bayesian Method?**

Most studies using GMM apply the maximum likelihood method to estimate parameters through classical likelihood procedures. It might be asked why should researchers be interested in the Bayesian method? In general, if sample size is large enough, maximum likelihood estimation provides all information about parameter estimates from the data. This operation performs the same statistical inferences as using asymptotic estimation theory (i.e., provided the normality assumptions are satisfied) (Gelman et al., 2014). Gelman et al. (2014) also described the difficulty in other statistical approaches to formulate reasonable parameter estimates with small-sample performance. For example, in the SEM statistical technique, the sampling distribution of parameter estimates is unknown for small sample size. Also, covariances and variances of parameter estimates are more likely to be incorrectly estimated in small sample studies (Scheines,
Hoijtink, & Boomsma, 1999). Using the Bayesian approach, statistical analysts can overcome the problem of small sample size. This technique provides the whole distribution, referred to as a posterior distribution over the parameter (Gill, 2008; B. O. Muthén & Asparouhov, 2012; Scheines et al., 1999).

Model complexity, defined as models with large numbers of parameters, is another reason to choose a Bayesian method. Since a complex model is a model with multiple dimensions, it cannot use a natural approach (i.e., numerical integration) to estimate parameters (Ansari et al., 2000; Dunson, 2000; Gill, 2008; Lynch, 2007; B. O. Muthén & Asparouhov, 2012; Scheines et al., 1999). Therefore, Bayesian analysis is considered to be a computational tool for the analysis of complicated statistical models with complex data structures.

Finally, a Bayesian approach could incorporate prior knowledge toward the parameters through the distribution applied to each parameter in the model following previous studies (Lynch, 2007) or theoretical recommendation. Using data to inform researchers about a parameter and incorporate the prior information accounts for the uncertainty in all parameter values which is the flexibility in the Bayesian analysis (Jiang & Mahadevan, 2009).

**Bayes’ Theorem**

The Bayesian statistical approaches depend upon the concept of probability (Hsu, 1999). Therefore, parameter estimates can be obtained through Bayesian inference, by calculating the probability of parameters conditionally on the data using the Bayes’ theorem. Let A and B be two possible outcomes and assume that $A = A_1 \cup \ldots \cup A_m$ for which $A_i \cap A_j = \emptyset$ for every $i \neq j$. Bayes theorem for the conditional probability for $A_j$ given B could be shown as
\[ P(A_j|B) = \frac{P(B|A_j)P(A_j)}{P(B)} = \frac{P(B|A_j)P(A_j)}{\sum_{i=0}^{m} P(B|A_i)P(A_i)}, \] (22)

(Bain & Engelhardt, 1992).

**Likelihood Function**

Bayes’ theorem is applied in Bayesian analysis to make inferential statements about the unknown parameter for any value of the variable of interest through the probability function called *likelihood function*. For example, let θ denote unobservable vector quantities or population parameters of interest and \( y \) denotes the observed data on each of a set of \( n \) objects or units \( y = (y_1, \ldots, y_n) \). The likelihood function (i.e., distribution of \( y \) for a given \( \theta \)) could be represented as \( f(y|\theta) \). To calculate the likelihood function, the density function of the latent class membership, \( z_i \), is specified. As previously mentioned, the latent class indicator, \( z_i \), is distributed as a multinomial distribution shown in Equation 23 as

\[ f(z_i) = \prod_{k=1}^{K} \pi_k^{z_{ik}}, \] (23)

and the complete likelihood function with the latent class membership \( z_i \) for individual \( i \) is

\[ L_i(\theta|y_i, z_i, \eta_i) = \prod_{k=1}^{K} [\pi_k f_k(y_i, | \eta_i)]^{z_{ik}} \] (24)

where \( f_k(y_i, \eta_i) \) is the \( k^{th} \) (\( k = 1, \ldots, K \)) component joint density of \( y_i \) and the latent effect \( \eta_i \), and \( \theta \) is the vector of unknown parameters.

Since unknown parameters are treated as random variables rather than fixed, these unknown parameters are assumed to have distributional qualities which are referred to as the prior distribution which is the distribution on the parameters of interest unconditional on the given data denoted by \( p(\theta) \). A compromise between the prior distribution and the data provides the posterior distribution which can be used to obtain
parameter estimates through Bayesian inference. The posterior distribution is also called the probability of parameters conditional on the data, denoted as $p(\theta|y)$. The issue regarding prior and posterior distribution is elaborated on in the following section.

**Prior Distributions**

A prior distribution is the available information about unknown parameters that describe degree of belief of relative likelihoods of events related to the parameter before the data are collected (Gill, 2008; B. O. Muthén & Asparouhov, 2012). The difference in beliefs is based on substantive theory, a researchers’ past experience, and previous research on the same populations. Applying prior knowledge toward parameters by using the prior distribution to account for uncertainty in all parameter values is one of the advantages of the Bayesian approach. Therefore, the prior distribution could be considered as the core of Bayesian analysis. To provide a prior distribution for unknown parameters has the same meaning as specifying Bayesian models (Gill, 2008). To use Bayesian methods, researchers need to specify priors for the parameters of interest in the study. The general categories of prior distributions used in Bayesian models are classified as noninformative priors, informative priors, and conjugate priors.

**Noninformative priors.** Priors with no information about unknown parameters are called noninformative (or diffuse) priors. When reliable prior information about parameters do not exist, the uniform distribution can be used (Gelman et al., 2014). This prior is an improper prior or weakly informative, but it is convenient for the purpose of illustration (Stephens, 2000).

**Informative priors.** Research in the Bayesian framework using informative priors is the most optimal and accurate on parameter estimation, specifically in a mixture confirmatory factor analysis (S. Depaoli, 2014). Depaoli demonstrated that the impact of
a prior distribution on the measurement model parameters was sensitive with small amounts of data being used for estimation. Informative priors may be elicited from previous empirical studies. Within the context of estimation of the number of latent class in growth mixture models, it appears that a certain prior distribution should be specified to assure the correct number of latent classes. A study regarding prior distribution on GMM by S. Depaoli (2014) concluded that using different priors on each unknown parameters would always provide different conclusions.

**Conjugate priors.** A conjugate prior is a prior distribution from the family of probability density functions when the posterior distribution follows similar functional forms to the prior. For example, the beta prior for the binomial distribution is an example of a conjugate prior and can be expressed symbolically as

\[
\text{(Beta prior)} \times \text{(Binomial likelihood)} = \text{Beta posterior}
\]  

(25)

For this example, the posterior distribution is the beta distribution. A conjugate prior provides an algebraic convenience to researchers in that it gives the closed-form expression for the posterior distribution. The posterior distribution is defined as the product of the likelihood function and prior distribution. For complicated models, including many dimensions, a conjugate prior distribution may be impossible to justify.

**Posterior Distributions**

In the Bayesian approach, the calculation of the probability of parameters conditionally on the data is needed, which is referred to as the posterior distribution. The posterior distribution represents knowledge after taking the data into account (Hamaker & Klugkist, 2011). Consider the basic Bayes theorem shown earlier in Equation 22. Substituting parameters, \( \theta \) for \( B \) and data, \( y \) for \( A \), statements about \( \theta \) given \( y \) can be
made following the joint probability distribution for $\theta$ and $y$, $p(\theta, y)$. The joint probability density function can be written as the product of the prior distribution, $p(\theta)$, representing the uncertainty before any data are observed and the sampling distribution or likelihood function divided by the marginal distribution of the data. Derived from the joint distribution through the Bayes theorem, the general notation of the posterior distribution is stated as

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y|\theta)}{p(y)}, \quad (26)$$

(Gelman et al., 2014), where $p(\theta)$ is the prior probability distribution, $p(\theta|y)$ is the posterior probability distribution, $p(y|\theta)$ is the likelihood function, and $p(y)$ is the marginal distribution of $y$. Since the marginal distribution of the data is the sum or integration over all possible values of $\theta$ which do not depend on $\theta$, the posterior is proportional to the product of the likelihood function and the prior distribution of the parameters. These expressions summarize the technical core of Bayesian inference in the form of unnormalized posterior density (Gelman et al., 2014) shown in Equation 27, that is

$$p(\theta|y) \propto p(y|\theta)p(\theta) \quad (27)$$

The Bayesian approach enables researchers to compute high-dimensional models yielding the posterior distribution which can be obtained with the Markov chain Monte Carlo (MCMC) algorithm (Gelman et al., 2014).

**Markov Chain Monte Carlo**

Obtaining the posterior distribution is one of the important steps in Bayesian computation. The steps to obtain the posterior distribution require more elaborate algorithms. If the posterior distribution has a closed form such as normal, gamma, beta, Poisson, and so forth, simulations can be performed directly using computer programming
routines. If the posterior distributions involve complicated or unusual or high dimensional models, approximation of the posterior distribution could be achieved by combining different algorithms for constructing and sampling from arbitrary posterior distributions. Markov chain Monte Carlo (MCMC) is a general method used to draw values of unknown parameters from approximate distributions and corrects the draws to perform a better approximation for the target posterior distribution (Gelman et al., 2014). As suggested by the name of the method, MCMC can be considered as having two components: a Monte Carlo component and a Markov chain component. The estimator from this method is given by the process of computing Monte Carlo integration using Markov chains. Monte Carlo simulation and MCMC have been briefly discussed as a simulation-based approach to the approximation of complex integrals.

As mentioned, two elements within MCMC are a Monte Carlo component and a Markov chain component. Monte Carlo integration is a process used to draw samples from the specified distribution, and then calculates sample averages to approximate expected values (Gilks et al., 1996). Markov chain Monte Carlo method is a generalization of the sampling method based on Markov chains introduced by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller (1953), and further extended by Hastings (1970). Markov chain Monte Carlo method is an established, suitable methodology used to sample from complicated distributions that are not feasible to sample directly. In the context of Bayesian inference, MCMC methods are used to draw samples from some target densities which are mostly non-standard or complex forms of distributions. The target density in Bayesian applications is the joint posterior, or the posterior density of the model parameters. There are two reasons why MCMC is used as an alternative method in the area of statistics (O’Neill, 2002). First, MCMC allows a huge amount of model flexibility.
That is, it does not need to specify a convenient distributional form when evaluating the high-dimensional integration over all the possible unknown parameters. Second, MCMC enables analysis of all parameters or functions of parameters on Bayesian application through a posterior distribution. Next, the posterior summaries for individual parameters or for joint distributions of parameters such as means, medians, variances, and

Markov chain Monte Carlo generates a sequence of $\theta^{(1)}$, $\theta^{(2)}$, $\ldots$ random variables of some set $T$. The key idea of MCMC is that at each time $t \geq 0$, the next state $\theta^{(t+1)}$ is sampled from the conditional distribution of $\theta^{(t+1)}$ given $\theta^{(1)}$, $\theta^{(2)}$, $\ldots$, $\theta^{(t)}$ depending only on the current state of the chain, $\theta^{(t)}$ (Gelman et al., 2014; Gilks et al., 1996). The process of sampling implies that at any time $t$ in the process, the conditional probability of making a transition kernel or transition function to a new state depends only on the latest state of the process. Therefore, the unknown parameter at time $t$, $\theta^{(t)}$, is conditionally independent of the previous values and can be written in Equation 28 as

$$
P(\theta^{(t+1)}|\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(t)}) = P(\theta^{(t+1)}|\theta^{(t)})$$

(28)

The process in Equation 28 shows that the random variable at time $t+1$, $\theta^{(t+1)}$, does not depend further on $\theta^{(0)}$, $\theta^{(1)}$, $\ldots$, $\theta^{(t-1)}$. A distribution $P(X_{t+1} = \theta^{(t+1)}| X_t = \theta^{(t)})$ which depends only on the current state of the chain $X_t = \theta^{(t)}$ is called the transition kernel of the chain (Gilks et al. 1996). There are three basic sampling methods of constructing the chains within MCMC.

The first sampling method, referred to as Gibbs sampling, was originally introduced by Geman and Geman (1984). Simulations following their scenario use the Gibbs sampler to generate realizations from a given Markov random field. Geman and Geman’s study focused on image-processing models (Casella & George, 1992). The principal theoretical contribution of their study was to investigate the Markov random
field by sampling and computing the mode of the posterior distribution (Geman & Geman, 1984). Applications of the Gibbs sampler algorithm to sample a complicated model, defined as a model with various unknown parameters or high dimensional integration were conducted by Smith and Robert (1993), Lu, Zhang, and Lubke (2011), as well as Zhang, Hamagami, Wang, Nesselroade, and Grimm (2007), in Bayesian inference literature reviews.

The second sampling method serves as a basis of all other sampling methods in which a modified Monte Carlo scheme is used in the generalization of the original study by Metropolis et al. (1953). Hastings (1970) reworked Metropolis et al.’s algorithm to relax the assumption of a symmetric proposal distribution. The adapted Metropolis algorithm by Hastings is referred to as the Metropolis-Hastings algorithm. Some researchers have applied the Metropolis-Hastings algorithm for numerical problems in statistical analysis. O’Neill (2002) reviewed the applications of MCMC using a Metropolis-Hastings algorithm on various data sets.

The third sampling method proposed by Green (1995) is called reversible jump Markov chain Monte Carlo (RJMCMC). The RJMCMC sampler can be viewed as an extension of the Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm is used to find the distribution of parameters in the model with a constant number of the dimension of the parameter space, whereas RJMCMC can be used to draw samplers from the varying dimension of the parameter space. An example of using the method of reversible jump sampling includes finite mixture models in which the number of mixture components is allowed on spaces of varying dimension.
**Gibbs Sampling**

Gibbs sampling is a MCMC algorithm for generating a Markov chain of random variables from a joint probability distribution. When the joint distribution is complex (i.e., high-dimensional modeling) or unknown but the conditional distribution of each variable is known, Gibbs sampling is useful. Gibbs sampling computes random variables from a marginal distribution indirectly (Casella & George, 1992). Instead of computing high-dimensional integration, which requires difficult calculations, replacing such integration by unidimensional random variable generation is a straightforward approach to computing random variables using Gibbs sampling (Casella & George, 1992). The key feature of Gibbs sampling is that samples are drawn from the full conditional distributions (Smith & Robert, 1993). The full conditional distribution is the distribution of the parameter of interest conditional on the known information of all the other parameters. It can be said that the transition kernel in the Gibbs sampling method is formed by the full conditional distributions (Gamernan & Lopes, 2006).

Suppose that the distribution of interest is $\pi(\theta)$, where the parameter $\theta$ consists of $q$ components or subvectors, $\theta = \theta_1, \theta_2, \ldots, \theta_q$. Each component $\theta_i$ can be considered as a scalar, a vector, or a matrix (Gamernan & Lopes, 2006). Let $\pi(\theta) = \pi(\theta_1, \ldots, \theta_q)$ denote the joint density, and let $\pi_i(\theta_i) = \pi(\theta_i | \theta_{-i})$, $i = 1, \ldots, q$ denote the full conditional densities for each of the components $\theta_i$, given all the components of $\theta$, except for $\theta_i$ at the current values. Consider that the full conditional densities are known, Gibbs sampling provide an alternative scheme to draw samples directly from a known marginal distribution. This technique samples one parameter at a time. For each iteration of the Gibbs sampler, the value of each component cycles through the subvectors of $\theta$. At iteration $t$, each subset $\theta^t_i$ is sampled individually from the conditional distribution given
all the other components of $\theta$, $\pi(\theta^t_i | \theta^{t-1}_{i-1})$. Thus each subvector $\theta_i$ is updated by repeatedly replacing the value of each component with a value sampled from its distribution conditional on the latest values of the other components of $\theta$. There are, thus, $q$ steps in iteration $t$. The Gibbs sampling algorithm is defined by the following iterations

1. Set initial values of the chain $j = 0; \theta^{(0)} = \theta^{(0)}_1, \theta^{(0)}_2, ..., \theta^{(0)}_q$;

2. obtain a new value $\theta^{(t)} = \theta^{(t)}_1, \theta^{(t)}_2, ..., \theta^{(t)}_q$ from successive random drawings from the full conditional distributions $\pi(\theta^t_i | \theta^{t-1}_{i-1}, x)$, $i = 1, \ldots, q$; $t = 1, \ldots, k$ as follows:
   - sample $\theta^{(t)}_1 \sim \theta^{(t-1)}_1, \ldots, \theta^{(t-1)}_q$,
   - sample $\theta^{(t)}_2 \sim \theta^{(t-1)}_2, \theta^{(t-1)}_1, \theta^{(t-1)}_3, \ldots, \theta^{(t-1)}_q$,
   - sample $\theta^{(t)}_3 \sim \theta^{(t-1)}_3, \theta^{(t-1)}_1, \theta^{(t-1)}_2, \theta^{(t-1)}_4, \ldots, \theta^{(t-1)}_q$,
   - 
   - sample $\theta^{(t)}_q \sim \theta^{(t-1)}_q, \theta^{(t-1)}_1, \ldots, \theta^{(t-1)}_q$;

3. change iteration $t = t + 1$ and return to step 2 until the chain converges to a stationary distribution

(Gamernan & Lopes, 2006). When convergence is reached, this means $\theta^{(t)}$ is sampled from $\pi$. Since the distribution of the next values $\theta^{(t+1)}$ given the values up to the present depend only on the latest value, the components of $\theta$ in this version of the algorithm move as a Markov chain. As a result of the properties of Markov chains, the process completes a transition from $\theta^{(t)}$ to $\theta^{(t+1)}$ given by 29 as

$$K(\theta^{(t)}, \theta^{(t+1)}) = \prod_{l=1}^{k} \pi(\theta^{(t+1)}_l | \theta^t_m, m > l, \theta^{(t+1)}_m, m < l)$$ (29)

(Smith & Robert, 1993).
**Metropolis-Hastings Algorithms**

The Metropolis-Hasting algorithm is a MCMC method that can be used for sampling from the specified target distribution which is a posterior distribution in Bayesian analysis (Gelman et al., 2014). Each iteration of the Metropolis-Hastings algorithm is divided into three steps: (a) generate a line by sampling from a candidate, proposal, or a jumping distribution \( q(\theta^*|\theta^{(t-1)}) \); (b) propose a new state through the line; and (c) accept or reject the proposed state according to the Metropolis-Hastings probability; or, keep the current state. To obtain a sequence of random variables \( \theta^{(1)}, \theta^{(2)}, \ldots \), these samples must be drawn via Markov chain from a distribution \( \pi \) with respect to \( \theta, \pi(\theta) \). The process of the Metropolis-Hastings algorithm starts with constructing a transition kernel or a transition probability, \( q(\phi|\theta) \), from the current state \( \theta^{(t-1)} = \theta \) to the next realized state \( \theta^* = \phi \). The candidate density assumes the property \( \sum_\phi q(\phi|\theta) = 1 \). It appears in the chain that the process moves from \( \theta \) to \( \phi \) more often than from \( \phi \) to \( \theta \) (Chib & Greenberg, 1995). It turns out, as shown in Equation 30, the multiplication of density \( q(\phi|\theta) \) and the target distribution \( \pi(\theta) \) is

\[
\pi(\theta)q(\phi|\theta) > \pi(\phi)q(\theta|\phi)
\]  

(30)

However, Equation 30 should be balanced to meet a sufficient condition in order to show that it is the equilibrium of the chain (Gamerman & Lopes, 2006). Introducing the probability of moving, \( \alpha(\phi|\theta) \), into the left hand of Equation 30 could reduce the number of moves from \( \theta \) to \( \phi \) and make the Equation 30 balanced as shown in Equation 31

\[
\pi(\theta)q(\phi|\theta)\alpha(\phi|\theta) = \pi(\phi)q(\theta|\phi),
\]

(31)

and consequently

\[
\alpha(\phi|\theta) = \frac{\pi(\phi)q(\theta|\phi)}{\pi(\theta)q(\phi|\theta)} = \frac{\pi(\phi)/q(\phi|\theta)}{\pi(\theta)/q(\theta|\phi)}
\]

(32)
(P. M. Lee, 2012). Once a transitional probability function or an acceptance probability is designed, then a general formula of the Metropolis-Hastings algorithm due to Hastings (1970) is presented

\[ \alpha(\phi|\theta) = \min \left( \frac{\pi(\phi)q(\theta|\phi)}{\pi(\theta)q(\phi|\theta)}, 1 \right) \]  \hspace{1cm} (33)

(Chib & Greenberg, 1995; Gamerman & Lopes, 2006; Mengersen & Tweedie, 1996; Roberts & Smith, 1994; Tierney, 1994).

Finally, the decision of moving the state can be made referring to the probability of the move, \( \alpha(\phi|\theta) \). If the chain is currently at a point \( \theta \), then it generates a candidate value \( \phi \) for the next step. If the candidate point is accepted, the next state becomes \( \phi \), so the probability of going from state \( \theta \) (i.e., \( \theta^{t-1} \)) to state \( \phi(\theta^*) \) is shown in Equation 34 as

\[
p^*(\phi|\theta) = \begin{cases} 
q(\phi|\theta)\alpha(\phi|\theta) & \text{if } \theta \neq \phi \\
0 & \text{if } \theta = \phi,
\end{cases}
\]  \hspace{1cm} (34)

which is also defined as the off-diagonal density of a Metropolis kernel (Lee, 2012; Tierney, 1994). If the candidate point is rejected, the chain remains in the present state \( \theta \). The probability when the algorithm remains at \( \theta \) is set by Equation 35 as

\[
r(\theta) = 1 - \sum_{\phi} q(\phi|\theta)\alpha(\phi|\theta)
\]  \hspace{1cm} (35)

(Lee, 2012).

The simulation of a draw from a target distribution or a posterior distribution can be summarized as follows:

1. Draw a starting point \( \theta^{(0)} \) from a starting distribution \( p_0(\theta) \). Note the starting point can be set by an arbitrary initial value \( \theta^{(0)} \).
2. For state 2

(a) Sample a proposal or a candidate point $\theta^*$ from a proposal distribution $q(\theta^*|\theta^{t-1})$ at time $t$,

(b) Calculate the ratio $\alpha(\theta^*|\theta^{t-1}) = \frac{\pi(\theta^*|y)/q(\theta^*|\theta^{t-1})}{\pi(\theta^{t-1}|y)/q(\theta^{t-1}|\theta^*)}$

3. Generate $U$ from an independent Uniform distribution on $(0, 1)$.

4. Compare $U$ with $\alpha(\theta^*|\theta^{t-1})$,
   
   if $U \leq \alpha(\theta^*|\theta^{t-1})$ the move is accepted and define $\theta^t = \theta^*$,
   
   if $U > \alpha(\theta^*|\theta^{t-1})$ the move is accepted and define $\theta^t = \theta^{t-1}$,

5. Change the time $t$ to $t+1$ and return to step 2 to get the sequence of random variable $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(t)}$,

   (Chib & Greenberg, 1995; Gamerman & Lopes, 2006; P. M. Lee, 2012).

The alternative MCMC algorithm within the Bayesian framework described above (i.e., Gibbs sampling and Metropolis-Hastings algorithm) is both theoretically straightforward. Both Gibbs sampling and Metropolis-Hastings algorithms are applied to simulation of a posterior distribution (i.e., target distribution) on spaces of fixed dimension. For simulation from a Markov chain whose state is a vector when the dimension is not fixed, a number of solutions within the Bayesian framework are Bayes factors (Richardson & Green, 1997) and reversible jump MCMC (Richardson & Green, 1997). Both are not of interest in the current project.

**Applications Using Metropolis-Hastings**

Markov chain methods for sampling from complicated models that include large numbers of parameters include Gibbs sampling and the Metropolis-Hastings algorithm.
The related theoretical frameworks and the practical issues regarding these techniques need to be considered. Several studies have examined the performance of Gibbs sampling on Bayesian inference. The straightforward routine Bayesian analyses in a range of important applications are presented by Smith and Robert (1993). Also, Lu et al. (2011) applied the growth mixture models with latent class dependent missing data in which data depend on latent random class membership using Bayesian estimation. The data augmentation method was utilized to obtain the likelihood function Lu et al.’s (2011) study. The conjugate priors were adopted for the class-specific growth curve parameters. For example, a multivariate normal-inverse Wishart distribution prior was used for the fixed effect in the model ($\beta_k$) and the scale variance-covariance matrix ($\Psi_k$); an inverse Gamma distribution prior was used for $\phi_k$; and a multivariate normal distribution prior was used for probit parameters ($\varphi_t$) and $\gamma_t$. Given the likelihood function and the priors, the joint posterior distribution of the unknown parameters was calculated. The full conditional distribution for the parameters was obtained rather than integrating out the marginal posterior distributions. Subsequently, the Gibbs sampling method was applied to generate Markov chain for each parameter. Lastly, the Bayesian inference obtained from those samples was applied.

Similarly, S. A. Depaoli (2012) studied an indication to detect model misspecification in the context of the number of mixed classes specified within Bayesian growth mixture modeling. The procedure in the Bayesian framework, namely the posterior predictive checking, was used. Trajectory shape, mixture of class proportions, sample size, and estimator were the conditions manipulated in Depaoli’s work. The conjugate priors for each parameter were specified. The conjugate prior for the class proportion parameter was a Dirichlet distribution. The next set of priors to specify were
factor loadings and factor means; both were set to be the normal distribution. Then the observed variable variances were assumed to follow an inverse gamma distribution. The last prior distribution to be specified in Depaoli’s study was the matrix of factor variances and covariances which was the Inverse Wishart distribution.

Many other applications of Markov chain simulation have been presented in the recent applied statistical literature. Eidsvik and Tjelmeland (2006) introduced a new Metropolis-Hastings algorithm using directional updates. By using two classes of directional updates which were a point auxiliary variable and an auxiliary direction vector and allowing the proposal distribution along the chain depending on the density of the auxiliary variable, the results availed the advantage of large moves in the Markov chain. Their findings resulted in the small autocorrelation in the length of samples. Gelman et al. (1996) provided the optimal symmetric jumping kernel for simulating a normal target distribution using the Metropolis algorithm. One of the important findings showed that, for a d-dimensional multivariate normal problem, the rate of acceptance associated with the Metropolis algorithm was approximately 44% for $d = 1$ and the acceptance rate declined to 23% when $d$ was close to infinity. Extensive theoretical works on the Metropolis-Hastings algorithm were reviewed by Chib and Greenberg (1995). Chib and Greenberg also provided a brief review of the acceptance-rejection method of simulation. Although the method they used was non-Markov, since the successive observations were statistically independent, its concepts often appeared in the Metropolis-Hastings algorithm. Sufficient conditions for the algorithm within Markov chain theory to converge to a target distribution have also been studied; for example, Roberts and Smith (1994) studied simple conditions for the convergence of both the Gibbs sampler and the
Metropolis-Hastings algorithm; Mengersen and Tweedie (1996) studied a theoretical perspective on rates of convergence of the Hastings and Metropolis algorithm.

To summarize, the two main techniques referred to in the current study are Gibbs sampling and Metropolis-Hastings algorithms. The Metropolis-Hastings algorithm is more general than the Gibbs sampler in terms of approximating the target posterior distribution and is particularly helpful for sampling parameters that do not have a recognizable form for their full conditional distribution. A more advantageously elaborate application is to use the Metropolis-Hastings algorithm for models that are not conditionally conjugate (Gelman et al., 2014). In another algorithm, Gibbs sampling, which is the simplest of Markov chain simulation algorithms, could be used to sample from each conditional posterior distribution. When the number of latent classes $q$ is small, the Gibbs sampler is more efficient (Gilks et al., 1996). Additionally, Gibbs sampling works well with a univariate, fully conditional distribution. Among the limited contributions of Gibbs sampling, it is unclear if researchers should be abandoning Gibbs sampling in favor of the Metropolis-Hastings procedure, applied to the whole simulation in the complex multivariate distribution. In this situation, Gibbs sampling would be a sensible choice because the Metropolis-Hastings algorithm requires a reasonably efficient proposal distribution, which could be difficult to obtain in order to ensure convergence to the correct target distribution. By doing so, the Metropolis-Hastings algorithm seems to be more involved than the Gibbs sampler. However, the problem of autocorrelation from staying in the same stage for more than one iteration in the Metropolis-Hastings algorithm causes slower convergence to distribution. In at least one study, M. H. Chen and Schmeiser (1993) proposed that the method to deal with the slow convergence was to select a good transition probability. The current dissertation project focused on the
Bayesian model determination problem and considered the issue such as the estimation on the number of latent classes on GMM.

**Applied Bayesian Analysis**

Muthén and Asparouhov (2012) applied Bayesian analysis to factor analysis and structural equation modeling. An informative prior, namely, the Inverse Wishart distribution, was used for covariance matrices in their study. The posterior distribution of the covariance matrices obtained was also the Wishart family of distributions. Using the same model as in Muthén and Asparouhov (2012)’s study, Levy (2011) specified normal prior distributions for the intercepts and factor means and loadings. The variance was set to be an inverse-gamma distribution, Bayesian analysis was applied in order to investigate point estimates of standard deviations and interval estimates for the parameters on SEM again with an informative prior distribution over the parameters (Scheines et al., 1999). Other examples of the utility of the Bayesian approach for analyzing data under the SEM framework for other statistical analyses have received much attention such as Bayesian structural equation modeling for hierarchical modeling (Jiang & Mahadevan, 2009) and Bayesian latent structure models with covariates (Cai, Lawson, Hessain, & Choi, 2012). Moreover, analysis of different response data with missing responses under SEM using the Bayesian approach have been proposed by Kim, Das, Chen, and Warren (2009) as well as Li, Kano, Pan, and Song (2012).

Few studies have linked complex SEM to Bayesian theory. Lu et al. (2011) evaluated a Bayesian approach to estimating robust growth mixture models with non-ignorable missingness. A full Bayesian method was proposed to estimate the model through the data augmentation method. The conjugate priors for model parameters were adopted with the class membership and missing probability expressed and a probit link
function in the study. Priors for the model parameters were specified as follows: (a) a multivariate normal-inverse Wishart distribution prior, an inverse Gamma distribution prior, and a multivariate normal distribution prior were used for different growth curve parameters and (b) a multivariate normal distribution prior was used for probit parameters. Zhang et al. (2007) demonstrated Bayesian methods for analyzing longitudinal data in applied research using both noninformative and informative priors. Conjugate priors were used in all Bayesian analysis in the study. For example, the inverse Gamma distribution prior was used for the variance of measurement error; the inverse Wishart distribution prior was used for the covariance matrix of the random effects parameters and normal distribution priors were used for the coefficients. Another application of the Bayesian approach to growth mixture modeling was conducted by Menthen, Boelaert, and Lesaffre (2012). Their proposed procedures were discussed and illustrated by means of a small simulation study and a real data set using uniform priors over the relevant interval was also applied in their study.

Chapter Summary

To summarize, in GMM, the assumption of homogeneous growth within classes is relaxed (Kaplan, 2002). Additionally, this model is accomplished by considering not only continuous latent variables but also categorical latent variables. The significant idea of GMM is that the population is assumed to consist of a mixture of \( K \) homogeneous subgroups; each with its own unique developmental trajectory (Preacher et al., 2008). According to B. O. Muthén (2008), \( K \) homogeneous subgroups are also referred to as \( K \) classes of individuals, each of which is described by a distinct set of growth model parameter values.
The fundamental information on latent class analysis models and assumptions and mathematical framework of growth mixture models have been established. Bayesian computation and an introduction to the Markov chain Monte Carlo method with two different algorithms (i.e., Gibbs sampling and Metropolis-Hastings algorithm) were also presented. Latent variable models including the growth mixture model have been conducted in different contexts in previous research under Bayesian and non-Bayesian frameworks. One of the most important aspects of these previous studies is to assist researchers in determining the number of latent classes of the growth mixture model. For the non-Bayesian method, most analyses have emphasized estimation on the basis of a maximum likelihood approach, with different fixed number of latent classes (Nylund et al., 2007; Tofighi & Enders, 2008). Fit indexes have been used to specify the model with the best fit and to identify the number of components (i.e., latent classes) corresponding to the criteria of the fit statistics applied (Nylund et al., 2007; Tofighi & Enders, 2008). The disadvantage of the combination of criteria to guide the decision making on the number of latent classes in growth mixture modeling is that the results have not been consistent except for the BIC fit index (Tofighi & Enders, 2008). Despite potential utility of the BIC, the findings of prior simulation studies indicate the BIC’s sensitivity to model fit in the presence of small sample size. Estimation of the number of latent classes in mixture models has also been conducted using Bayesian methods (Steele & Raftery, 2009). Informative prior such as the Poisson distribution has been specified on the estimation of unknown parameters in Steele and Raftery’s (2009) study.
CHAPTER III

METHODOLOGY

To estimate the number of latent classes in growth mixture models, a Bayesian estimation method was introduced in the current dissertation using Markov chain Monte Carlo simulation methods implemented via Metropolis-Hastings algorithms. Simulated data sets were used to investigate the performance of using the Bayesian analysis to estimate the number of components on growth mixture models. All data sets came from a growth mixture model with three main factors: sample size, number of waves of data (i.e., time points), and number of latent classes. To use the Bayesian method to estimate the value of unknown parameters, previous knowledge about the parameters needs to be placed on model parameters in terms of the distributions. The distributions of these parameters are called prior distribution and which can take on different levels of information. The prior information concerning the unknown parameters was set based on the review of literature. In the current study, I focused on two types of prior information (i.e., informative and noninformative priors). The estimation of growth mixture model parameters related to the number of latent classes was performed within a Bayesian framework, based on Markov chain Monte Carlo using the Metropolis-Hastings algorithm. The procedure drew samples from the posterior distribution. Finally, statistical inference was conducted based on the distribution created from the generated Markov chain.
Growth Mixture Models

A growth mixture model comprised of the latent growth curve model and a latent
categorical variable $z$ is shown in Figure 5. The latent growth curve model is exhibited
first as a part of growth mixture modeling, which was the focus of the current project. In
the diagram, the large square represents an LGC model, the small square represents
measured or observed variables, the circle represents the latent variables, and the triangle
represents a constant. The single-headed arrow from the triangle to each of the latent
variables, $\eta_i$, indicates the linking of the constant on the path coefficients. In this general
diagram, the number of items (or, observed variables) in each latent variable and the
number of latent classes are not specified.

The basic mathematical model including the distribution of parameters related to
GMM is briefly reviewed. Suppose the observed values of individual $i$, $y_i$, were generated
independently from a mixture of $K$ underlying population distributions with unknown
proportion, $\pi_1, \ldots, \pi_k$. The subscript $k$ represents the number of latent classes in the
growth mixture model. The K-component growth mixture model for a random vector of
observed repeated measured for individual $i$, $y_i$, is defined as in Figure 5.

$$
f(y_i) = \sum_{k=1}^{K} \pi_k f_k(y_i | \theta_k),
$$

(36) (Lubke & Neale, 2008; G. McLachlan & Peel, 2000). The probability density function for
$y_i$ given that the observation is from the $k^{th}$ latent class shown in Equation 36 is
parameterized by all unknown parameters, $\theta_k$.

The growth mixture model assumes that $y_i$ follows a combination of $K$
distributions; each latent class distribution represents a trajectory of each class in the
latent curve model. The number of latent classes, $K$, was the unknown parameter of
The Unconditional Growth Mixture Model

A traditional growth mixture model without covariates (i.e., unconditional growth model) was simulated in the current study. Following the diagram in Figure 5, suppose that in a longitudinal study, $N$ subjects are measured on $T$ measurement occasions or time points. Let $\mathbf{y}_i = \text{be a } T \times 1 \text{ random vector for individual } i (i = 1, 2, \ldots, N)$, which can
also be considered as repeated measure outcomes for person $i$ across $T$ measurement occasions, where $y_{it}$ is the observed or outcome variable of person $i$ at time $t$ ($t = 1, 2, \ldots, T$), and let $\eta_i$ be a $q \times 1$ random vector of continuous latent variables (i.e., growth parameters) of $i^{th}$ individual that has $q$ elements. For example, when $q = 2$ the model corresponds to a linear growth mixture model; the latent variable in this case includes the latent random intercept ($I_i$) followed by the latent random slope ($S_i$) for individual $i$.

Under the SEM framework, GMM specification includes both a measurement model and structural model. The measurement model referred to as a LGC model, that shows the relation of the outcome $y_i$ and the latent $\eta_i$ and can be written as

$$y_i = \Lambda \eta_i + \varepsilon_i,$$  (37)

where the subscript $i$ allows the parameters to vary across individuals, the $\Lambda$ term is a $T \times q$ matrix of factor loadings, and $\varepsilon_i$ is a $T \times 1$ vector of residuals of measurement errors, which are assumed to follow a multivariate normal distribution. It is shown next as

$$\varepsilon_i \sim MN_T(0, \Theta)$$  (38)

(Lu et al., 2011), where $\Theta$ is the matrix of error variances. Since invariant error variances over time are assumed, then the covariance matrix $\Theta = V(\varepsilon_i)$, is specified as $\Theta = I_T \phi$ (Lu et al., 2011), where $\phi$ is a scalar and $I_T$ is a $T \times T$ identity matrix.

The equation for the growth parameter portion of the growth mixture model across $k$ classes is

$$\eta_i = \alpha + \beta \eta_i + \zeta_i.$$  (39)

The growth parameter vector ($\eta_i$) is specified as before and is assumed to be a random vector in the growth mixture model, $\alpha$ is a vector of the growth factor means contained in the latent factor ($\eta_i$) of each class, $\beta$ is a matrix of coefficients relating growth factors to
one another, and $\zeta_i$ are $q \times 1$ vectors of errors. The $\zeta_i$ vector is assumed to follow a multivariate normal distribution with 0 mean and covariance matrix $\Psi = V(\zeta_i)$ specified through the following equation

$$\zeta_i \sim MN_q(0, \Psi)$$

(40)

(F. Li et al., 2001), where $\Psi$ is the $q \times q$ factor covariance matrix that can be allowed to vary across the $k$ classes. Note that the variability of individuals in each class in the trajectory parameter (i.e., growth factors or intercept and slope) is captured in $\Psi$. As can be seen, the observed model in Equation 37 and unobserved model in Equations 39 extend the previous growth mixture model shown in Equation 18 and 19 by including the latent trajectory classes, $K$, (where $K = 1, 2, \ldots, K$) which allows for different shapes of growth in the population. The growth mixture model presented in this study is characterized following a form of invariance presented by Duncan, Duncan, and Acock (2001). If the invariance is assumed, each class has the same set of matrices and vectors of parameters with the similar value of fixed hyperparameters and estimated parameters.

**Growth Mixture Model Considered**

The parameter vector of the mean structure model and covariance structure model within class is identified by taking the expected value of the conventional latent growth curve model as shown in Equation 37. It is assumed that the growth factor (i.e., same as residual $\zeta_i$) from the structural model and the time-specific residuals in the measurement model $\varepsilon_i$ are normally distributed and independent. The repeated measures $y_i$ are then multivariate normally distributed (Hipp & Bauer, 2006; Lu et al., 2011). The density of $y_i|\eta_i$ can be written in the notation form as

$$y_i|\eta_i \sim MN_q(\mu, \Sigma),$$

(41)
The mean and covariance structure of the repeated measure from Equation 37 for each latent class \( K \) are implied by taking the expected value of the latent growth curve model

\[
E(y_i) = E(\Lambda \eta_i + \epsilon_i)
\]

\[
\mu = \Lambda \eta_i
\]  

and

\[
\Sigma = \Lambda \Psi \Lambda' + \Theta.
\]

Further specified is the distribution of the latent variable \( \eta_i \) which follows a multivariate normal distribution with the mean, \( \mu_\eta \), and covariance, \( \Sigma_\eta \), which are derived from Equation 39 as

\[
\eta_i = \alpha + \beta \eta_i + \zeta_i
\]

\[
(I - \beta) \eta_i = \alpha + \zeta_i
\]

\[
\eta_i = (I - \beta)^{-1} \alpha + (I - k)^{-1} \zeta_i.
\]

Therefore,

\[
\mu_\eta = (I - \beta)^{-1} \alpha,
\]

and

\[
\Sigma_\eta = (I - \beta)^{-1} \Psi [(I - \beta)^{-1}]'.
\]

All the parameters previously derived are used to calculate the likelihood function which are discussed in the following section.

Another parameter specified here is a vector of categorical variables, \( z_i \), which corresponds to the class membership in GMM. Since the classification of a given
observation, \( y_i \), into a particular latent class is unobserved, the marginal distribution must be used for the observations. The mixture distribution with probabilities \( \pi_1, \ldots, \pi_k \) might be thought of as hyperparameters determining the probability in the mixture components or latent classes. The distribution of unobserved variables \( z_i \) is specified as a multinomial distribution as shown in Equation 21 with the density distribution of \( f(z_i) = \prod_{k=1}^{K} \pi_k^{z_{ik}} \).

The model described in Equations 37 through 46 present a traditional growth mixture model without covariates. The residuals \( \varepsilon_i \) and \( \zeta_i \) are assumed to follow a multivariate normal distribution, i.e., \( \varepsilon_i \sim MN_q (0, \Theta) \) and \( \zeta \sim MN_q (0, \Psi) \), and are independent. Recall that the general form of a multivariate normal probability distribution is

\[
 f(x) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \times exp \left( -\frac{1}{2}(x - \mu)'\Sigma^{-1}(x - \mu) \right) 
\]

(Gelman et al., 2014). Therefore, the specification of the probability densities in Equation 36, which are the combination of measurement and structural model in the previously mentioned context, is extended in the mixture components of a multivariate normal distribution of all parameters as

\[
 f(y_i | \eta_i) = \sum_{k=1}^{K} \pi_k f_k(y_i, \beta, \Lambda, \alpha_k, \Theta, \Psi_k | \eta_i), 
\]

Here, the number of latent classes, \( K \), is unknown and the main objective is to estimate the number of such unobserved latent classes. In fitting the growth mixture models shown in Equation 48, estimates of the growth factor means(\( \alpha_k \)), the residual variances (\( \Theta_k \), \( \Theta = I_T^T \phi \)), and the covariance of residuals (\( \Psi_k \)) including the class proportions (\( \pi_k \)) are also estimated. The design matrix of the class membership indicator (\( z_i \)) mentioned in Equation 21 as multinomial distribution can also be obtained.
Bayesian Estimation for the Growth Mixture Models

The procedures using a Bayesian method were developed in the current dissertation through R program for obtaining the posterior distribution of the number of components or latent classes in growth mixture modeling along with the posterior distribution of other unknown parameters in the model. Theoretically, the posterior distribution of parameters is the product of the likelihood function or sampling distribution and the prior distribution of unknown parameters divided by the sum or integral of the probability of the data with respect to all values of the parameter which is the marginal distribution of the data. Since the denominator does not involve any parameters, the posterior is proportional to the likelihood function multiplied by the prior distribution.

Likelihood Functions

The complete likelihood function with the latent class membership $z_i$ is specified in Equation 24 as $L_i(\theta | y_i, z_i, \eta_i) = \prod_{k=1}^{K} [\pi_k f_k(y_i | \eta_i)]^{z_{ik}}$. To obtain the likelihood, the observed data $y_i$ are augmented with the latent random effects, $\eta_i$, and the class membership indicator, $z_i$. If the latent variables are included in the growth mixture model (GMM), then the joint distribution function for individual $i$ and unobserved variable can be obtained by Equation 49

$$f_k(y_i, \eta_i | k, \beta, \Lambda, \alpha_k, \phi, \Psi_k)$$

$$= f_k(y_i | \eta_i, \Lambda_k, \phi, k) \times f_k(\eta_i | \beta, \alpha_k, \Psi_k)$$

$$= (2\pi)^{-\frac{q}{2}} |\Sigma_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(y_i - \Lambda \eta_i)'(\Sigma_k)^{-1}(y_i - \Lambda \eta_i)\right)$$

$$\times (2\pi)^{-\frac{g}{2}} |\Sigma_\eta|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\eta_i - (I - \beta)^{-1}\alpha_k)'(\Sigma_\eta)^{-1}(\eta_i - (I - \beta)^{-1}\alpha_k)\right)$$

(49)
where $\Sigma_k = \Lambda \Psi_k \Lambda^T + \Theta$ and $\Sigma_\eta = (I - \beta)^{-1} \Psi_k [(I - \beta)^{-1}]^T$ are previously derived. The matrix of coefficients $\beta$ and the matrix of factor loadings $\Lambda$ are fixed while fitting a growth mixture model, whereas the vector of growth factor means $\alpha_k$ and the matrix of $\Psi_k$ are class-specific growth parameters that can vary for different latent classes $K$. The scalar $\phi$ is also estimated but does not vary across different classes. The latent class membership $z_i$ is added in the function to obtain a complete likelihood function for individual $i$. Therefore, the complete likelihood of GMM for the $i^{th}$ individual becomes

$$L_i = L_i(y_i, \eta_i, z_i)$$

$$= \prod_{k=1}^{K} \left[ \pi_k f_k(y_i, \eta_i | \Lambda, \phi, \alpha_k, \beta, \Psi_k, k) \right]^{z_{ik}}$$

$$= \prod_{k=1}^{K} \left[ \pi_k f_k(y_i | \eta_i, \Lambda, \phi) \times f_k(\eta_i | \alpha_k, \beta, \Psi_k, k) \right]^{z_{ik}}$$

The likelihood function for the whole samples is obtained by

$$L(y, \eta, z) = \prod_{i=1}^{N} \prod_{k=1}^{K} \pi_k$$

$$\propto \prod_{i=1}^{N} \prod_{k=1}^{K} \left[ \pi_k \right]$$

$$\times (2\pi)^{-\frac{T}{2}} |\Sigma_k|^{-\frac{1}{2}} \exp \left( -\frac{1}{2} (y_i - \Lambda \eta_i)^T (\Sigma_k)^{-1} (y_i - \Lambda \eta_i) \right)$$

$$\times (2\pi)^{-\frac{T}{2}} |\Sigma_\eta|^{-\frac{1}{2}} \exp \left( -\frac{1}{2} (\eta_i - (I - \beta)^{-1} \alpha_k)^T (\Sigma_\eta)^{-1} (\eta_i - (I - \beta)^{-1} \alpha_k) \right)$$

where the covariance matrix of the observed variable ($\Sigma_k$) is defined as $\Sigma_k = \Lambda \Psi_k \Lambda^T + \Theta$, with the observed variable variance $\Theta$ with $\Theta = I_T \phi$ and the unobserved variable variance $\Sigma_\eta$ was defined as $\Sigma_\eta = (I - \beta)^{-1} \Psi_k [(I - \beta)^{-1}]^T$.

**Prior Distribution**

The model parameters that were estimated in the current dissertation project, as shown in the likelihood function for a growth mixture model in Equation 51, are the number of latent classes ($K$), the unobserved variables ($\eta_i$), the factor means ($\alpha_k$), the
class proportions ($\pi_k$), a scalar ($\phi$) for the covariance matrix ($\Theta$), the factor covariance ($\Psi_k$), and the class membership indicator ($z_i$) including the factor loadings ($A_k = \Lambda$) and the coefficients of growth factors ($\beta_k = \beta$).

In order to apply a Bayesian method to estimate the number of latent classes in growth mixture models, priors for the model parameters need to be specified and used to calculate the posterior distribution of each parameter. To make the accurate inference for the number of latent classes, $K$, in the growth mixture model, the priors should be selected carefully. The posterior distribution for $K$ could be highly related to the prior chosen for $K$ but also the priors chosen for the other unknown parameters of the growth mixture models.

Two forms of priors are informative and noninformative. It is common for researchers to use a noninformative prior distribution in the Bayesian framework; however, using informative priors is more optimal (S. Depaoli, 2014). As a result, in the current project I took into account both noninformative and informative priors for the estimation of number of latent classes in GMM.

The priors of unknown parameters in growth mixture models were chosen to be informative only, while the prior for the number of components parameter, $K$, were chosen to be both informative and noninformative priors. The noninformative or weakly informative priors on $K$ for the number of components were set to be a uniform distribution as suggested by Gill (2008). The uniform distribution is denoted as

$$K \sim DU(N).$$

The uniform priors were designed to take three different distributions, $DU(N)$, the values of $N$ taken on 3 to 5 to see the different boundaries of prior information toward the number of latent classes estimated. For the informative priors on $K$, Poisson priors were
examined as prior information to see how strong a role they play on the estimation of the number of latent classes in growth mixture models (Stephens, 2000). As demonstrated by Stephens (2000), high and low values of parameters from a Poisson based approach affect the inference for the number of latent classes, \( K \), in growth mixture models. Also, it has been recommended by Stephens that large value of parameter \( \lambda \) could result in better mixing over \( K \), but it was not clear how an optimal value among these parameters should be achieved. Therefore, the three levels of Poisson parameters (e.g., 3 through 5) were of interest in the current project. The Poisson distribution is denoted as

\[
K \sim \text{Poisson}(\lambda),
\]

where \( \lambda \) is the hyperparameter representing the mean and variance of the distribution.

The Poisson distribution has the density function

\[
f(k) = \frac{\lambda^k}{k!} \exp(-\lambda).
\]

The parameter for the informative prior which is a discrete uniform distribution was also varied (i.e., \( N = 3, 4, 5 \)). The discrete uniform has the density function

\[
f(k) = \frac{1}{N}.
\]

The prior distributions of other unknown parameters as shown in the likelihood function of this current project, are specified based on the literature review from previous studies (S. A. Depaoli, 2012; Lu et al., 2011).

The prior for the factor means \( \alpha_k \) is the multivariate normal distribution denoted as

\[
\alpha_k \sim N(\mu_\alpha, \sigma_\zeta^2)
\]

where \( \mu_\alpha \) is the hyperparameter representing the mean vector of the growth factors with the value of \( 0_k \) and \( \sigma_\zeta^2 \) is the hyperparameter representing covariance matrix of the growth.
factors defined as $I_k$. The dimension of both the mean vector and covariance depends on the number of latent classes. The multivariate normal distribution has the density function

$$f(\alpha_k) = (2\pi)^{-\frac{q}{2}} |\Sigma_\eta|^{-\frac{1}{2}} \exp \left(-\frac{1}{2} (\alpha_k - \mu_\alpha)'(\Sigma_\eta)^{-1}(\alpha_k - \mu_\alpha) \right)$$  \hspace{1cm} (57)

The prior for the class proportion parameter $\pi$ is the Dirichlet distribution denoted as

$$\pi_K \sim D[\delta_1, \ldots, \delta_K],$$ \hspace{1cm} (58)

where $\delta_1, \ldots, \delta_K$ are the hyperparameters that represents the proportion of components in the $K$ mixture components. The hyperparameter for proportion prior is set to be the vector of two, where the numbers of elements depends on the number of latent classes (Lu et al., 2011). The Dirichlet distribution has the density function

$$f(\pi_K) = \frac{\Gamma(\delta_1 + \ldots + \delta_K)}{\Gamma(\delta_1) \ldots \Gamma(\delta_K)} \pi_1^{\delta_1} \ldots \pi_K^{\delta_K-1}.$$ \hspace{1cm} (59)

The prior for $\phi_k$ is an inverse Gamma distribution denoted as

$$\phi_K \sim IG\left[\nu_0K, s_0K^{-1}\right],$$ \hspace{1cm} (60)

where $\nu_0k$ and $s_0k$ are known parameters specified as $\nu_0k = s_0k = 0.002$. The inverse Gamma distribution has the density function

$$f(\phi_K) \propto \phi_K^{-\frac{\nu_0K}{2}} \exp \left(-\frac{s_0K}{2\phi_K} \right).$$ \hspace{1cm} (61)

Note that the prior on scalar $\phi$, which is the matrix of residual variances, $\Theta = I_T\phi$, was adopted from the study by Lu et al. (2011).

The prior for the factor covariance ($\Psi_K$) is the inverse Wishart distribution denoted as

$$\Psi_K \sim IW_q[m_{K0}, V_{K0}],$$ \hspace{1cm} (62)

where $m_{K0}$ is a scalar hyperparameter and $V_{K0}$ is a $q \times q$ matrix. The value for $m_{K0}$ is 2
and $V_{K0}$ is a $2 \times 2$ identity matrix for the linear growth factor (Lu et al., 2011). The inverse Wishart distribution has the density function

$$f(\Psi_K) \propto |\Psi_K|^{-\frac{m_{K0}q+q+1}{2}} \exp(-\frac{1}{2}tr(V_{K0}\Psi_K)^{-1}).$$

(63)

The next step toward a more general growth mixture model is to assume an independent prior distribution for all unknown parameters in the growth mixture model. The subsequent discussion is the joint prior density which must have the product from all unknown parameters. Based on the probability density function discussed above, the joint prior density for noninformative priors (i.e., discrete uniform; $N = 3, 4, 5$) corresponds to

$$f(k, \alpha_K, \pi_K, \phi, \Psi_K)$$

$$= f(k) \times f(\alpha_k) \times f(\pi_k) \times f(\phi) \times f(\Psi_k)$$

$$= \frac{1}{N}$$

$$\times (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left(-\frac{1}{2} (\alpha_K - \mu_\alpha)'(\Sigma_{\alpha})^{-1}(\alpha_K - \mu_\alpha) \right)$$

(64)

$$\times \frac{\Gamma(\delta_1 + \ldots + \delta_k)}{\Gamma(\delta_1) \ldots \Gamma(\delta_k)} \frac{\pi_1^{\delta_1} \ldots \pi_K^{\delta_K - 1}}{\pi_1^{\delta_1} \ldots \pi_K^{\delta_K - 1}}$$

$$\times \phi^{-\frac{m_k}{2} - 1} \exp \left(-\frac{s_{0K}}{2\phi} \right)$$

$$\times |\Psi_K|^{-\frac{m_{K0}q+q+1}{2}} \exp(-\frac{1}{2}tr(V_{K0}\Psi_K)^{-1})$$

The joint prior density for informative prior (i.e., Poisson; $\lambda = 3, 4, 5$) is
\[ f(k, \alpha_K, \pi_K, \phi, \Psi_K) \]
\[ = f(k) \times f(\alpha_K) \times f(\pi_K) \times f(\phi) \times f(\Psi_K) \]
\[ = \frac{\lambda^K}{K!} \exp(-\lambda) \]
\[ \times (2\pi)^{-q/2} |\Sigma_\eta|^{-1/2} \exp\left(-\frac{1}{2} (\alpha_K - \mu_\eta)'(\Sigma_\eta)^{-1}(\alpha_K - \mu_\eta)\right) \] (65)
\[ \times \frac{\Gamma(\delta_1 + \ldots + \delta_K)}{\Gamma(\delta_1) \ldots \Gamma(\delta_K)} \pi^{\delta_1} \ldots \pi^{\delta_{K-1}} \]
\[ \times \phi^{-m_{Kq} - 1} \exp\left(-\frac{s_0K}{2\phi'}\right) \]
\[ \times |\Psi_K|^{-m_{Kq+q+1}/2} \exp(-\frac{1}{2} tr(V_{K0}\Psi_K)^{-1}) \]

The latent variable, \( \eta_i \), and class membership, \( z_i \), are considered as unknown parameters in the current dissertation. For \( \eta_i \), the conditional function is a multivariate normal density which is based on \( \mu_\eta \) and \( \Sigma_\eta \). The density function is denoted as
\[ \eta_i \sim \text{MN}_q(\mu_\eta, \Sigma_\eta). \] (66)

The multivariate normal has the density function
\[ f(\eta_i|\beta, \alpha_K, \Psi_K) = (2\pi)^{-q/2} |\Sigma_\eta|^{-1/2} \exp\left(-\frac{1}{2} (\eta_i - \mu_\eta)'(\Sigma_\eta)^{-1}(\eta_i - \mu_\eta)\right), \] (67)
where \( q \) is 2 corresponding to the linear growth factor (i.e., two columns for factor loadings) and
\[ \mu_\eta = (I - \beta)^{-1} \alpha_K, \] (68)
\[ \Sigma_\eta = (I - \beta)^{-1} \Psi_K [(I - \beta)^{-1}]' \] (69)

For \( z_{ik} \), the conditional distribution is a multinomial distribution given \( \pi \). The multinomial has the density function
\[ f(z_i|\pi) = \prod_{k=1}^{K} \pi_k^{z_{ik}}. \] (70)
These two unknown parameters are treated as the hierarchy representing priors, adding to the joint priors of all unknown parameters (Marrs, N.S.). The final form for the joint prior distribution is

\[
f(k, \alpha_K, \pi_K, \phi, \Psi_K, \eta_i, z_i)
\]

\[
= f(k) \times f(\alpha_K) \times f(\pi_K) \times f(\phi) \times f(\Psi_K) \times f(\eta_i | \beta, \alpha_K, \Psi_K) \times f(z_i | \pi_K)
\]

\[
= \frac{\lambda^k}{K!} \exp(-\lambda)
\]

\[
\times (2\pi)^{-\frac{q}{2}}|\Sigma_{\eta}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\alpha_K - \mu_\alpha)'(\Sigma_{\eta})^{-1}(\alpha_K - \mu_\alpha)\right)
\]

\[
\times \frac{\Gamma(\delta_1 + \ldots + \delta_K)}{\Gamma(\delta_1) \ldots \Gamma(\delta_K)} \pi_1^{\delta_1} \ldots \pi_K^{\delta_K-1}
\]

\[
\times \phi^{-\frac{\gamma}{2KK_0}} \exp\left(-\frac{\gamma_0 K}{2\phi_K}\right)
\]

\[
\times \left|\Psi_K\right|^\frac{m_KK_0+q+1}{2} \exp\left(-\frac{1}{2}tr(V_{K0} \Psi_K)^{-1}\right)
\]

\[
\times (2\pi)^{-\frac{q}{2}}|\Sigma_{\eta}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\eta_i - \mu_\eta)'(\Sigma_{\eta})^{-1}(\eta_i - \mu_\eta)\right)
\]

\[
\times \prod_{K=1}^{K} \pi_{iK}^{z_iK}.
\]

**Posterior Distributions**

The posterior distribution is proportional to the likelihood function multiplied by the prior distribution. The priors of unknown parameters in growth mixture models and the likelihood function were previously specified to use Bayesian methods to estimate the number of latent classes. The model parameters include the number of latent classes (K), the \(k^{th}\) mixing proportion (\(\pi_K\)), and the unobserved variables (\(\eta_i\)). The growth curve parameters \(\alpha_K, \phi, \Psi_K, (K = 1, 2, \ldots, K)\) are also estimated along with the number of components.

The marginal posterior distribution of the parameter of interest, which is the aim of Bayesian analyses, is hard to obtain because of high-dimensional integration in the
model discussed. Therefore, MCMC techniques were used for posterior computation of the proposed model in the current project. In the case where all the underlying priors are non-conjugate priors with no closed form of posterior distribution, the Metropolis-Hastings sampling method was utilized. More specifically, when the dimension of the parameter space in Markov chain simulation changes from one iteration to the next iteration, the Metropolis algorithm using the method of reversible jump sampling is suitable to perform (Gelman et al., 2014). Then the posterior distribution of unknown parameters is obtained.

Markov Chain Monte Carlo Simulation

The Metropolis-Hastings algorithm is implemented within Markov chain Monte Carlo to obtain the posterior distribution for all parameters conducted in the current project. The Metropolis-Hastings algorithm refers to the methods for sampling from the posterior when the full conditional distribution for each parameter cannot be obtained. The full conditional distribution is the distribution of the parameter conditional on all the others parameters with known information, or the full conditionals when they look like any known distribution. Metropolis-Hastings sampling produces a new sample for all dimensions at once rather than choosing a new sample for each dimension separately from one another. The Metropolis-Hastings technique is applied by generating a sequence of samples iteratively with the distribution of the next sample depending only on the sample from the present state of the chain (i.e., Markov chain).

Specially, one parameter is updated with respect to the acceptance ratio or acceptance probability specified in Equation 32. The minimum value between one and a value from the fraction of candidates and the most recently drawn value in the chain are compared. If the probability from the candidate is accepted, the candidate value is used in the next iteration. If the candidate is rejected, the current value is reused in the next
Different candidate distributions are used for each unknown parameter of the growth mixture models in the Metropolis-Hastings sampling procedure.

**Candidate Distributions**

The candidate or proposal distribution can be any distribution from which it is easy to simulate draws (Lynch & Western, 2004). The appropriate choice for a proposal distribution is an important step to sample candidate parameters. The appropriate choice for each unknown parameter is selected based on either the theory or the use from the previous study. The following proper candidate distributions were used for the sampling:

(a) truncated Poisson for the number of latent classes \((K)\) which corresponds to the characteristic of the parameter that was the count number and it cannot be zero; (b) multivariate normal distribution for latent variable \((\eta_i)\) (Lu et al., 2011); (c) multivariate normal for factor means \((\lambda_K)\); (d) Dirichlet distribution for draw proportion \((\pi_K)\); (e) gamma for the scalar specified in the residual variance of observed values \((\phi)\); (f) Wishart distribution for the covariance matrix \((\Psi_K)\) (Diaz-Garcia, Jáimez, & Mardia, 1997); and (g) multinomial for the class membership indicator \((z_i)\) (Evans, Hastings, & Peacock, 2000).

The truncated Poisson has the density function

\[
f(k) = \frac{\lambda^k \exp(-\lambda)}{k!(1 - \exp(-\lambda))},
\]

(Plackett, 1953; van der Heijden, Bustami, Cruyff, Engbersen, & van Houwelingen, 2003).

**Convergence and Summary Statistics**

The next step of the analysis in the current project was to assess convergence and calculate summary statistics. Demonstration of the application of the Bayesian method usually requires a burn-in or warm-up period before the estimated unknown parameters converge in the distribution to the true posterior. The burn-in period is the iteration at
which a run should be thrown away to avoid auto-correlation of the samples (Geyer, 1992). The chain converges with the stationary distribution when using a larger number of iterations within the algorithm (Leiby, Have, Lynch, & Sammel, 2012). Therefore, the burn-in period was set to 10,000 iterations and the last 10,000 iterations were used as post-burn-in iterations following Depaoli’s (2014) study in generating a Markov chain through the Metropolis-Hasting method. A total of 20,000 iterations were run for the convergence testing and data analysis. For any given parameter, the estimated posterior variance of the parameter, \( \hat{R} \), was used to assess convergence. The estimated posterior variance of the parameter was estimated by
\[
\hat{R} = \sqrt{\frac{\text{var}(\psi|y)}{W}}
\] (Gelman et al., 2014), where \( \psi \) was the simulated value, which was specified as \( \psi_{ij} (i = 1, \ldots, n; j = 1, \ldots, m) \). The subscripts \( i \) and \( j \) were specified after discarding the warm-up iterations. Then the post-burn-in iterations were split into the first and second half (i.e., \( m \) is the number of subgroups and \( n \) is the number of length of each chain). In the current project, the simulation of length 20,000 iterations was run two times (i.e., two chains), and then iterations 1 through 10,000 of each chain were discarded as warm-up. After discarding the burn-in period, the length of each chain was left with 10,000 iterations and each chain was split into two groups. Each group consisted of 5,000 iterations. The number of groups, \( m = 4 \), and the number of iterations, \( n = 5,000 \), were then specified to calculate \( \text{var}(\psi|y) \) in Equation 73. This posterior estimated variance consists of the between-sequence variances (B) and within-sequence variances (W). B and W can be computed from Equations 75 through 77.
\[
B = \frac{n}{m - 1} \sum_{j=1}^{m} (\bar{\psi}_j - \bar{\psi}_..)^2, \quad \text{where} \quad \bar{\psi}_j = \frac{1}{n} \sum_{i=1}^{n} (\psi_{ij}), \quad \text{and} \quad \bar{\psi}_.. = \frac{1}{m} \sum_{j=1}^{m} \bar{\psi}_j,
\] (74)
\[ W = \frac{1}{m} \sum_{j=1}^{m} s_j^2, \text{ where } s_j^2 = \frac{1}{n} \sum_{i=1}^{n} (\psi_{ij} - \bar{\psi}_j)^2, \quad (75) \]

and

\[ \text{var}(\psi | y) = \frac{n}{n} W + \frac{1}{n} B \quad (76) \]

(Gelman et al., 2014, p. 284), where \( \bar{\psi}_j \) is the within-sequence means, \( \bar{\psi}_. \) is the grand mean, and \( s_j^2 \) is the variance within chain. The convergence of the simulation is calculated from the unknown number of latent classes (k), and the constant value specified in the residual variance of observed data (\( \phi \)) which are scalar estimands. These two single parameters were the only two parameters that were used to calculate the convergence to ensure a precise estimation.

After passing the convergence tests, all parameters including the number of latent classes (K) were summarized based on four batches of each estimation. All simulations drawn were divided into small groups (i.e., batches) due to the recommendation made by Bayesian method.

The average estimate \( \bar{\theta}_j \) over four batches of converged simulation was

\[ \text{Average estimate} = \bar{\theta}_j = \frac{1}{4} \sum_{i=1}^{4} \hat{\theta}_{ij}, \quad (77) \]

where, \( \hat{\theta}_{ij} \) was the estimated parameter of \( j_{th} \) parameter, \( \theta_j \), in the \( i^{th} \) simulation the batch or simulation group.

Each unknown parameter was estimated based on four sets of data from converged simulation replications using Markov chain Monte Carlo simulation. The average estimate of each unknown parameter across this data set was obtained using Equation 77. The dissertation results were analyzed in terms of bias, standard error, and incredible (i.e., confidence) interval.
Empirical standard error. The standard deviation of the parameter estimates, also known as the empirical standard error of the number of latent class, $K$, parameter was obtained by

$$\text{Empirical standard error} = \sqrt{\frac{1}{3} \sum_{i=1}^{k} (\hat{\theta}_{ij} - \bar{\hat{\theta}}_j)^2}$$

(78)

Average lower and upper limits of the 95% confidence interval. The average lower and upper limits of the 95% confidence interval of the number of latent class, $K$, were defined as

$$\text{Lower limit} = \frac{\sum_{i=1}^{4} \hat{\theta}_{ij}^l}{4}, \text{ and } \text{Upper limit} = \frac{\sum_{i=1}^{4} \hat{\theta}_{ij}^u}{4}$$

(79)

where $\hat{\theta}_{ij}^l$ denotes the 95% lower limit of confidence interval for the $j^{th}$ parameter, and $\hat{\theta}_{ij}^u$ denotes the 95% upper limit of confidence interval for the $j^{th}$ parameter. The lower and upper limits of each parameter are calculated from the 95% highest posterior density credible interval.

As previously mentioned, the aim of this current dissertation is to estimate the number of latent classes in growth mixture models. The other six GMM parameters were also estimated simultaneously. When several parameters are estimated at the same time, it is not possible to obtain an approximately unbiased estimator (Gelman et al., 2014). This is because the information or knowledge of these parameters is relevant to the estimation of other parameters. The Bayesian estimates theoretically are expected to be biased (Gifford & Swaminathan, 1990). Gifford and Swaminathan (1990) conducted research to investigate the Bayesian and joint maximum likelihood estimators and found that both Bayesian and joint maximum likelihood had bias in the estimation. Study by Ho et al. (2011) was similar to Gifford and Swaminathan’s study in terms of the bias in the posterior mean of the parameters of interest in their study. As such, the bias diagnosis
across all model parameters are omitted in the summary statistics part in the current dissertation.

**Verifying the Validity of a Simulation**

A simulation in the current dissertation is presented to evaluate the validity of the MCMC method used to estimate the number of components (i.e., latent classes) on growth mixture models. The specifications of different scenarios and different priors on number of latent classes are verified to ascertain the validity of the Markov chain Monte Carlo sampling for estimating the parameters using the Bayesian method. To simplify the presentation, simulation design, data generation, and simulation implementation are presented. A linear GMM with different numbers of latent classes in the simulation was assumed.

**Simulation Design**

To verify the accuracy of the computer programming created to estimate the number of latent classes on growth mixture models, three main factors were considered: sample size, waves of data, and number of latent classes.

**Sample size.** Three sample size levels, $N = 15, 50, \text{ and } 200$, were studied in the current dissertation. These values of sample sizes were chosen based on a review of GMM in both applied and simulation studies (L. Li & Hser, 2011; Nylund et al., 2007; Tofghi & Enders, 2008). The sample size of 200 was chosen to represent the moderate sample sizes necessary to address the requirements of applied researchers who conduct a moderate sample size, or utilizing the data from an existing data base. Small sample sizes, $N = 15 \text{ and } 50$, were chosen in this dissertation to address the needs of researchers who conduct research based on smaller scale studies especially for investigating individual differences such as the study by Grimm (2007) of individual achievement on reading recognition using
the data from the National Longitudinal Survey of Youth (NLSY), conducted on 34 children in some subset.

**Waves of data.** D. Rogosa et al. (1982), Willett (1989), and Voelkle (2007) provided statistical models for the estimation of change corresponding to time paths. Specially, Rogosa et al. argued that collection of only two waves of data (e.g., pretest and protest) offers limited information about the question of change. The effective measurement of change should be based on collecting data from more than two data collection points, or waves of data. Willett explained that more waves of data were always more appropriate in order to improve the reliability of the different scores when change was measured. Furthermore, Willet demonstrated that applying three waves of data showed an approximately 250% increase in reliability of the measurement of change compared with two waves of data. Willet’s study found that there has been very little benefit of adding more than four waves of data in longitudinal analysis. Therefore, the number of waves of data used in the current project was selected based on values of \( T = 2, 3, \) and 4, which are found in most previous studies about change under LGC modeling and GMMs (B. O. Muthén & Curran, 1997; B. O. Muthén, 2004; Tofighi & Enders, 2008).

**The number of latent classes specified.** Three levels of number of latent classes, \( K = 2, 3 \) and 4, were specified to mimic the number of latent classes in growth mixture models that are likely to be used by researchers in practice (Nylund et al., 2007; Tofighi & Enders, 2008). The GMM with a true \( K = 2 \) class model is considered a simple structure model and set to linear trend model (L. Li & Hser, 2011; Lu et al., 2011; Nylund et al., 2007). The second level of latent classes, \( K = 3 \), was set to represent the model that has been most often used in GMM research in both simulation studies (S. Depaoli, 2014; Nylund et al., 2007; Peugh & Fan, 2012; Tofighi & Enders, 2008) and applied research
(Kaplan, 2002; F. Li et al., 2001; Wang, 2007; Wiesner & Windle, 2004). The $K = 4$ level of numbers of latent classes is representative of a more complex model including the models from both applied (Bilir et al., 2008; Huang et al., 2010) and simulation research (Nylund et al., 2007).

The class proportions selected in the current dissertation followed the study by Nylund et al. (2007) for a 2-class model and Tofighi and Enders (2008) for a 3-class model. The proportions in each class for the 4-class model were selected to be equal in this dissertation. The class size specifications are given in Table 1.

Table 1

<table>
<thead>
<tr>
<th>Class size specification in growth mixture models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class size</td>
</tr>
<tr>
<td>Growth mixture model</td>
</tr>
<tr>
<td>2-class model</td>
</tr>
<tr>
<td>3-class model</td>
</tr>
<tr>
<td>4-class model</td>
</tr>
</tbody>
</table>

Design Summary

To sum up, the first scheme represents the model which is a growth mixture model without covariates. Six different prior distributions of the number of latent classes were applied based on whether the information about the parameters was known or not. Along with three different sample sizes, three numbers of waves of data, and the three levels of number of latent classes specified were also crossed (see Table 2). With all conditions applied, analyses for the total of 162 simulation designs were conducted on growth mixture models in the current dissertation. The defined parameter values to generate
growth mixture modeling data are the values applied from Hipp and Bauer (2006). Even though the authors presented the parameter values for a latent growth curve model, a random value for each parameter for the GMM data in the current study was generated within the specified range of Hipp and Bauers study. Theoretically, the factor loadings of linear growth factors and quadratic growth factors are specified systematically according to the number of waves of data. Note that the matrix of each type of factor loading used in this dissertation was changed due to the time points of the study (e.g., factor loadings of 3, 4, and 5 time points).

Table 2

*Design Summary*

<table>
<thead>
<tr>
<th>Priors</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Nonformative</td>
<td>Uniform; DU(2)</td>
</tr>
<tr>
<td>2</td>
<td>Nonformative</td>
<td>Uniform; DU(3)</td>
</tr>
<tr>
<td>3</td>
<td>Nonformative</td>
<td>Uniform; DU(4)</td>
</tr>
<tr>
<td>4</td>
<td>Informative</td>
<td>Poisson; Poisson(3)</td>
</tr>
<tr>
<td>5</td>
<td>Informative</td>
<td>Poisson; Poisson(4)</td>
</tr>
<tr>
<td>6</td>
<td>Informative</td>
<td>Poisson; Poisson(5)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample size</th>
<th>15</th>
<th>50</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waves of data</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Number of component</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

*Data Generation*

Developing computer algorithms to generate growth mixture data in this dissertation was based on mathematical models previously specified to describe the
manner of the GMM population. Unobserved (i.e., latent) and observed variables were generated based on the parameters specified from previous research using the `mvtnorm` function in the R program version 3.3.1. A linear GMM with 2, 3, and 4 latent trajectory classes was the focus of the current project. The longitudinal data for the preliminary simulation study assumed the values of 3, 4, and 5 repeated measures. First, latent variable, $\eta_i$, was generated following Equations 39 through 69. As in Equation 44, the latent variable was specified as $\eta_i = (I - \beta)^{-1}\alpha_k + (I - \beta)^{-1}\zeta_i$.

As linear latent trajectory classes were assumed in the current study, two 2-dimensional matrices were set for $\beta$ and $I$ corresponding to the number of column for a linear growth mixture model. For example, it was assumed that $I$ is a 2-dimensional identity matrix and $\beta_k = \beta = \begin{bmatrix} 0.80 & 0.80 \\ 0.80 & 0.80 \end{bmatrix}$, (Hipp & Bauer, 2006). Therefore, the only two parameters that were allowed to vary with classes were the vector means of latent factors, $\alpha_k$, and the residual variance, $\zeta_i$.

The `runif` function in the R program was used to create a random number generator for alpha ($\alpha_k$). The residual variance for unobserved variables varied across the latent classes following a multivariate normal distribution as specified in Equation 40, $\zeta_i \sim MN_q(0, \Psi)$. Where the subscript $q$ corresponded to the type of growth trajectory in the model, for instance, when $q = 2$, the corresponding model represents a linear model which was the focus in the current dissertation. Furthermore, when $q = 3$, the corresponding model represented a quadratic growth curve model. The growth factor covariance matrices were assumed to follow the possible range of the parameter values from Hipp and Bauer (2006) and distributed using Equation 69 as $\Sigma_\eta = (I - \beta)^{-1}\Psi_k[(I - \beta)^{-1}]^\prime$.

The unobserved variables were generated first by using Equation 39 with the growth factors specified according to linear growth in each class. Subsequently, the
observed variables were generated from Equation 37, \( y_i = \Lambda \eta_i + \varepsilon_i \), by plugging the unobserved variables generated earlier into this observed variable equation. The step of generating the observed variable data involves a multivariate normal distribution. The defined distribution of the population to simulate the data was developed from the literature reviews of previous work in Bayesian growth mixture modeling (S. Depaoli, 2010; Lu et al., 2011). The corresponding vector means and the variance components were specified first and then applied in the model to generate the observed data for GMM. The means of the observed variables are given in Equation 42 as \( \mu = \Lambda \eta_i \) where \( \Lambda \) and \( \eta_i \) are specified earlier. The covariance matrix of the observed variable was calculated from Equation 43, \( \Sigma_k = \Lambda \Psi_k \Lambda^T + \Theta \). The factor loadings, \( \Lambda \), and the factor covariance matrix, \( \Psi_k \), are the same as specified earlier. The matrix of residual variances, \( \Theta \) was set as \( \Theta = I_2 \phi \), where \( \phi \) is a scalar and was set to be 1 to generate data (Lu et al., 2011).

Then, the mvrnorm R function was used to generate multivariate normal observation vectors using the vector mean and covariance matrix specified.

**Simulation Process**

Using the idea of Bayesian statistics, the fundamental information of population parameters was applied by updating the original distribution of parameters of interest by conditioning on data through the likelihood function (Gill, 2008). The final distribution is called the posterior distribution. Drawing samples from the target distribution or posterior distribution of all unknown parameters was conducted to understand the behavior of the statistical estimates through their sampling distributions. These sampling distributions were derived from the data generated by the method of Markov chain Monte Carlo method using the Metropolis-Hastings algorithm. The estimation of unknown parameters including the number of latent classes for the GMM data files was conducted.
in five steps: (a) data generation process, (b) calculation of the likelihood function,
(c) calculation of the joint prior distribution for all unknown parameters, (d) calculation
of the joint posterior probability distribution, and (e) conducting sampling from the joint
posterior distribution of parameters and hyperparameters.

Once the joint posterior distribution was obtained, simulation procedures using
the Metropolis-Hastings sampling method were implemented. Metropolis-Hastings
algorithm samples were generated from joint proposal distribution using the following four
steps at each iteration.

1. Generate Markov chains for model parameters through the Metropolis-Hastings
algorithm. The following procedures were used for drawing samples from the
posterior distribution.

   i. Set the start values. The initial values for model parameters were assigned
      from other simulation studies with growth mixture models (Hipps & Bauer,
      2006).

   ii. Simulate the draws by sampling the candidate parameters \( \theta^* \) from a candidate
      or proposal distribution \( q(\theta^*|\theta^{(t-1)}) \) at time \( t \). The proposal distributions used
      for generating candidates were set for each parameter as previously mentioned.

   iii. Evaluate the posterior density at the candidate point and previous point by
      the calculation of the ratio appearing in Equation 33 with \( \theta^{(t-1)} = \theta \) and \( \theta^* = \phi \), then
      \[ \alpha(\theta^*|\theta^{t-1}) = \min \left[ \frac{\pi(\theta^*)q(\theta^{(t-1)}|\theta^*)}{\pi(\theta^{(t-1)})q(\theta^*|\theta^{(t-1)})}, 1 \right]. \]

   iv. Generate the candidate distribution \( (U) \) from a uniform distribution

   v. Compare the candidate distribution \( (U) \) with \( \alpha(\theta^*|\theta^{t-1}) \);

      if \( U \leq \alpha(\theta^*|\theta^{t-1}) \) the move was accepted and define \( \theta^t = \theta^* \),

      if \( U > \alpha(\theta^*|\theta^{t-1}) \) the move was not accepted and define \( \theta^t = \theta^{t-1} \),
vi. Return to step 2 and repeat until the number of draws was obtained

vii. Alter the time $t$ to $t+1$ to get the sequence of random variables $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(t)}$

2. Run Markov chain for 20,000 iterations and record the iterations after 10,000 simulation draws (i.e., burn in period) to ensure that the Markov chains pass the convergence test with no autocorrelation.

3. Test the convergence of Markov chains.

4. Calculate the inferential statistics.

Analysis of Simulated Data

The Metropolis Hastings algorithm was implemented in the R program to estimate the number of components or latent classes in GMM on the basis of a Bayesian approach. All unknown parameters in the growth mixture model were estimated along with the number of components. A growth mixture model was computed with 20,000 total iterations that included 10,000 for burn-in and 10,000 for post burn-in to ensure independence in the sample values. The sample values were saved for convergence testing and data analysis. After running the simulation for a while, the convergence of the simulated sequences was monitored. The estimated posterior variance of any given parameter, $\hat{R}$, was calculated to assess convergence. The value of $\hat{R}$ closed to 1 (i.e., the closer to 1 the better) was used as a criterion to prove convergence (Kass, Carlin, Gelman, & Neal, 1998). For each simulation condition, the average estimate of each parameter was calculated first, and then summary statistics such as empirical standard error, and average lower and upper limits of the 95% confidence interval were reported based on 20,000 iterations of sampling run in the R program.
The Bayesian analysis described above was used to investigate 27 data sets, each of them coming from growth mixture models with three different waves of data (i.e., time points=3, 4, 5) and three different numbers of latent classes (i.e., 2, 3, 4). The sample sizes were in all cases \(N = 15, N = 50, N = 200\). Every data set of simulated GMM data was computed from two chains with 20,000 iterations for each chain. Samples drawn from a posterior distribution in each chain were divided into two groups with 10,000 iterations each. Therefore, the samples from the target distribution (i.e., posterior distribution) used to calculate inferential statistics consisted of four groups or four replications with 10,000 iterations in each replication. After fitting all unconditional growth mixture models (i.e., GMM without covariates), the parameters were estimated. The reported parameters of each class included class proportion, intercept and slope correlation, growth parameter means (i.e., intercept and slope), and variances (i.e., intercept and slope). \(P\)-value and 95% credibility (i.e., confidence) intervals were also produced in the growth mixture model estimates using Markov chain Monte Carlo.

Markov chain Monte Carlo estimation in Mplus Version 7.3 (Muthén & Muthén, 2010) was also used to fit the growth mixture model from each of the simulated data sets for comparison with the estimates using the Bayesian method developed in the current dissertation. The implementation of Bayesian methods can also be applied to the simulated data using Mplus for the computation. The different degrees of prior knowledge for the model parameters were specified. For example, the number of latent classes had a Poisson distribution; the mixture proportion received the Dirichlet distribution; the factor loadings presumed a multivariate normal prior; the error covariance specified an inverse Gamma; the slope factor had a multivariate normal distribution; the intercept factor received a normal distribution; and factor covariance had an inverse Wishart prior.
CHAPTER IV

RESULTS

Analyses for the total of 162 convergence diagnostics were conducted on growth mixture models. The 162 diagnostics refer to six priors, three sample sizes, three time points, and three levels of classes. These diagnostics were run two times with 20,000 iterations for each run which to two chains of MCMC. The convergence on two scalar parameters (i.e., \( K \) and \( \phi \)) in a growth mixture model were examined first. The parameter estimates on growth mixture models were then calculated based on the converged model. The goal of this dissertation was to estimate the number of latent classes in growth mixture models using a Bayesian method. The growth mixture model parameters such as the latent classes (\( K \)), the unobserved variable matrix (\( \eta_K \)), the factor mean vector (\( \alpha_K \)), the class proportion vector (\( \pi_K \)), a scalar for the covariance matrix (\( \phi \)), the factor covariance matrix (\( \Psi_K \)), and the class membership indicator matrix (\( z_K \)) were also estimated together with the number of latent classes. Moreover, the GMM generated data sets were fitted using Mplus (L. K. Muthén & Muthén, 2012). Mplus is a statistical modeling program that allows the analysis of both cross-sectional and longitudinal data with either observed or unobserved heterogeneity. Mplus was used to duplicate the estimation of some unknown parameters such as class proportion, intercept and slope of growth parameter means as well as intercept and slope of variances. However, the estimated parameters that can be produced using Mplus are not exactly the same as in
the developed R program in the current study. This dissertation addressed the following research questions:

Q1 Could the developed R program using the Bayesian method support researchers to estimate the number of latent classes on growth mixture models?

Q2 How can researchers select the candidate distribution in estimating the number of latent classes on growth mixture models?

Q3 What is the informative prior performance for different values of parameters on estimation for the number of latent classes in growth mixture models?

Questions were answered by a simulation study which was presented to evaluate the performance of the proposed Bayesian growth mixture models using the Metropolis-Hastings algorithm. To specify the performance of Bayesian approach, two, three, and four latent trajectory classes were estimated. Three, four, and five occasions of data were generated and sample sizes of 15, 50, and 200 were considered on each occasion. It was also assumed there was no covariate in the models used in this dissertation project. The results are detailed below according to sample size and the true number of latent classes specified in the simulated GMM data. Moreover, monitoring convergence was assessed first by calculating the estimated posterior variance (\(R\)) of two scalar parameters, the number of latent classes (K), and a scalar for the covariance matrix (\(\phi\)). The following section comprises the performance of the estimation for the number of latent classes in GMM using a Bayesian approach. The final section of this chapter shows the estimation of all parameters in GMM by using R programs as well as the parameter estimates using Mplus. In conclusion, the summary of results section shows the performance of the Bayesian method using an R program developed for the current dissertation to estimate the number of latent classes in growth mixture models.
Model Convergence

Two generating Markov chains using the Metropolis-Hastings sampling method with a total of 20,000 iterations each were run for the convergence testing. With the burn-in period of 20,000 iterations for each chain, the value of the estimated posterior variance (\(\hat{R}\)) of the two scalar quantities in questions were monitored. These quantities include the number of latent classes (K) and the designated scalar for the residual variance corresponding to the observed values (\(\phi\)). The \(\hat{R}\) for K with informative and noninformative prior is summarized in Tables 3 through 8 and in Tables 6 through 8 for \(\phi\). Each table is organized according to the level of noninformative or informative prior distribution, sample size, wave of data, and number of latent classes specified to generate GMM data. An R program was used to calculate \(\hat{R}\). Cells with \(\hat{R}\) at some low or high value (e.g., greater than 1) would be indicative of a problem with convergence testing. Therefore \(\hat{R}\) close to 1, indicates that the sequences have been mixed and the chain consists of a representative subset (Kass et al., 1998). Appendix A shows the details of implementation in the computer languages of R.

Number of Latent Class Convergence

Tables 3 through 8 suggest that \(\hat{R}\) for all sample sizes had nearly perfect convergence for all growth mixture models with different numbers of latent classes. The \(\hat{R}\) values ranged from 0.9999103 to 1.001169 for the Poisson distribution with hyperparameter (i.e., a parameter of a prior distribution) \(\lambda = 3\) and 0.9999 to 1.00039 for the discrete uniform with hyperparameter 3 (see Table 3). Table 5 provides the \(\hat{R}\) for Poisson with hyperparameter \(\lambda = 4\) and discrete uniform with hyperparameter 4. The wide range of the values show 0.9999002 to 1.000782 and 0.9999022 to 1.000939, respectively. For K with informative prior, all sample sizes under the Poisson distribution
with hyperparameter $\lambda = 5$ showed that the convergence had been reached based on the values ranging from 0.9999001 to 1.000794. This phenomenon also occurred in the connection with the $\hat{R}$ for discrete uniform with hyperparameter 5, the values were between 0.9999001 and 1.001137 (see Table 8).

Table 3

*The Estimated Posterior Variance ($\hat{R}$) of the Number of Latent Classes (K) for Informative Prior (Poisson with $\lambda = 3$)*

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>15</td>
<td>3</td>
<td>0.999949</td>
<td>1.000299</td>
<td>0.999907</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>0.999928</td>
<td>0.999910</td>
<td>1.000065</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.999938</td>
<td>0.999961</td>
<td>0.999993</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>1.001169</td>
<td>0.999950</td>
<td>0.999981</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.000282</td>
<td>0.999919</td>
<td>1.000033</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.999983</td>
<td>0.999903</td>
<td>0.999995</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>1.000092</td>
<td>0.999960</td>
<td>1.000109</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.000231</td>
<td>1.000167</td>
<td>1.000985</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.000161</td>
<td>1.000255</td>
<td>0.999959</td>
</tr>
</tbody>
</table>

*Note.* Poi(3) = Poisson distribution with $\lambda = 3$

N = Sample size

T = Time points
Table 4

*The Estimated Posterior Variance (\(\hat{R}\)) of the Number of Latent Classes (K) for Noninformative Prior (Discrete Uniform with parameter 3)*

<table>
<thead>
<tr>
<th>Number of True Latent Classes</th>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DU(3)</td>
<td>15</td>
<td>3</td>
<td>1.000390</td>
<td>0.9999756</td>
<td>1.000319</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>1.000029</td>
<td>0.999945</td>
<td>0.999925</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>1.000065</td>
<td>0.999958</td>
<td>1.000319</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>3</td>
<td>0.999939</td>
<td>0.9999466</td>
<td>0.999960</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.999904</td>
<td>0.999916</td>
<td>0.999913</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.999962</td>
<td>0.999927</td>
<td>0.999935</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>200</td>
<td>3</td>
<td>0.999900</td>
<td>0.999989</td>
<td>0.999984</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.999910</td>
<td>1.000042</td>
<td>1.000350</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.999939</td>
<td>0.999909</td>
<td>0.999900</td>
<td></td>
</tr>
</tbody>
</table>

*Note. DU(3) = Discrete uniform with hyperparameter 3
N = Sample size
T = Time points*
Table 5

The Estimated Posterior Variance ($\hat{R}$) of the Number of Latent Classes ($K$) for Informative Prior (Poisson with $\lambda = 4$)

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(4)</td>
<td>15</td>
<td>3</td>
<td>1.000053</td>
<td>0.999959</td>
<td>0.999959</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.000410</td>
<td>0.999933</td>
<td>0.999903</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.000712</td>
<td>0.999969</td>
<td>0.999902</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>1.000045</td>
<td>0.999903</td>
<td>1.000079</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.000280</td>
<td>0.999977</td>
<td>1.000208</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.999927</td>
<td>1.000091</td>
<td>0.999905</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>1.000106</td>
<td>1.000782</td>
<td>0.999987</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.000250</td>
<td>0.999919</td>
<td>0.999949</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.999900</td>
<td>1.000072</td>
<td>0.999923</td>
<td></td>
</tr>
</tbody>
</table>

Note. Poi(4) = Poisson distribution with $\lambda = 4$
N = Sample size
T = Time points
Table 6

*The Estimated Posterior Variance (\(\hat{R}\)) of the Number of Latent Classes (K) for Noninformative Prior (Discrete Uniform with parameter 4)*

<table>
<thead>
<tr>
<th>Prior</th>
<th>Number of True Latent Classes</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(4)</td>
<td>15</td>
<td>3</td>
<td></td>
<td>0.999952</td>
<td>0.999906</td>
<td>0.999911</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td>1.000437</td>
<td>1.000021</td>
<td>0.999922</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td>0.999902</td>
<td>0.999901</td>
<td>0.999972</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td></td>
<td>1.000258</td>
<td>0.999919</td>
<td>1.000068</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td>0.999926</td>
<td>1.000280</td>
<td>1.000159</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td>1.000939</td>
<td>1.000602</td>
<td>0.999994</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td></td>
<td>0.999915</td>
<td>1.000008</td>
<td>1.000330</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td>0.999903</td>
<td>0.999976</td>
<td>1.000296</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td>0.999940</td>
<td>0.999936</td>
<td>1.000330</td>
</tr>
</tbody>
</table>

*Note.* DU(4) = Discrete uniform with hyperparameter 4  
N = Sample size  
T = Time points
Table 7

*The Estimated Posterior Variance (\( \hat{R} \)) of the Number of Latent Classes (K) for Informative Prior (Poisson with \( \lambda = 5 \))*

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(5)</td>
<td>15</td>
<td>3</td>
<td>0.999978</td>
<td>0.999978</td>
<td>1.000022</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.000010</td>
<td>0.999909</td>
<td>0.999904</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.000794</td>
<td>0.999966</td>
<td>0.999966</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>0.999900</td>
<td>0.999900</td>
<td>0.999901</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.999936</td>
<td>0.999905</td>
<td>1.000119</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.999935</td>
<td>1.000638</td>
<td>0.999907</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>1.000612</td>
<td>1.000154</td>
<td>0.999936</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.999902</td>
<td>1.000126</td>
<td>1.000025</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.999936</td>
<td>0.999902</td>
<td>0.999948</td>
<td></td>
</tr>
</tbody>
</table>

*Note.* Poi(5) = Poisson distribution with \( \lambda = 5 \)

N = Sample size

T = Time points
Table 8

*The Estimated Posterior Variance ([\hat{R}]) of the Number of Latent Classes (K) for Noninformative Prior (Discrete Uniform with parameter 5)*

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(5)</td>
<td>15</td>
<td>3</td>
<td>0.999905</td>
<td>1.000050</td>
<td>0.999957</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>0.999900</td>
<td>1.000202</td>
<td>0.999965</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>1.000437</td>
<td>0.999902</td>
<td>0.999924</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>0.999905</td>
<td>1.000050</td>
<td>0.999966</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>0.999905</td>
<td>0.999906</td>
<td>0.999927</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>1.000044</td>
<td>0.999903</td>
<td>0.999928</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>1.001137</td>
<td>1.000002</td>
<td>1.000057</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>0.999966</td>
<td>0.999903</td>
<td>0.999934</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>1.000179</td>
<td>1.000828</td>
<td>0.999902</td>
</tr>
</tbody>
</table>

*Note.* DU(5) = Discrete uniform with hyperparameter 5  
N = Sample size  
T = Time points

**Residual Variance Convergence**

Tables 9 through 14 are concerned with assessing convergence for \( \phi \). Likewise, the estimated posterior variance (\( \hat{R} \)) closer to 1 would indicate more representative samples. Poisson distributions with parameter 3, 4, and 5 on parameter \( K \) were specified as informative priors, whereas discrete uniform distributions with \( \lambda = 3, 4, \) and 5 were considered as noninformative prior for each table. Based on \( \hat{R} \) for a 2-class, a 3-class, and
4-class growth mixture models across all sample sizes and time points, the outcome indicated rather similar values as in $\hat{R}$ for $K$ that is, all values were consistently close to 1.

As can be seen from Table 9, the $\hat{R}$ for $\phi$ for the Poisson prior with $\lambda = 3$, the lowest and the highest values were 0.9999 and 1.00033, respectively, while the $\hat{R}$ values for $\phi$ of uniform prior with parameter 3 ranged from 0.999901 to 1.000825. At all time points and numbers of latent classes specified, $\hat{R}$ values under Poisson prior with $\lambda = 4$ range from 0.9999 to 1.001193 (see Table 11). Regarding the discrete uniform distribution with parameter 4 $\hat{R}$ values ranged between 0.9999028 to 1.00055. These values are presented in Table 12. For the Poisson prior with $\lambda = 5$ (see Table 13), $\hat{R}$ for $\phi$ was mostly acceptable (i.e., all values were ranged around 1, that is from 0.9999 to 1.00157). This result also occurred for the $\hat{R}$ for $\phi$ when noninformative prior discrete uniform with parameter 5 was set, the values were between 0.9999017 and 1.000649. The converged results (i.e., $\hat{R}$ almost equal to 1) indicated that the chains had been in accordance with the posterior distribution.
Table 9

*The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Informative Prior (Poisson with \( \lambda = 3 \))*

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>15</td>
<td>3</td>
<td>1.00033</td>
<td>1.000083</td>
<td>0.999900</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>0.999926</td>
<td>0.999957</td>
<td>1.000108</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>1.000220</td>
<td>1.000116</td>
<td>0.999942</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>0.999901</td>
<td>1.000004</td>
<td>0.999914</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>0.9999</td>
<td>1.00033</td>
<td>0.999929</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.999900</td>
<td>0.999997</td>
<td>1.000203</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>1.000079</td>
<td>0.999917</td>
<td>0.9999626</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>1.000318</td>
<td>0.999917</td>
<td>1.000013</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.999904</td>
<td>0.999900</td>
<td>1.000063</td>
</tr>
</tbody>
</table>

*Note.* Poi(3) = Poisson distribution with \( \lambda = 3 \)

N = Sample size

T = Time points
Table 10

*The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Noninformative Prior (Discrete Uniform with parameter 3)*

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>15</td>
<td>3</td>
<td>1.000011</td>
<td>0.999901</td>
<td>1.000004</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>1.000016</td>
<td>1.000166</td>
<td>0.999979</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.999978</td>
<td>0.999942</td>
<td>0.999938</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>0.999975</td>
<td>1.000160</td>
<td>0.999921</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>0.999973</td>
<td>1.000124</td>
<td>1.000085</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>1.000825</td>
<td>1.000137</td>
<td>0.999934</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>0.999901</td>
<td>0.999972</td>
<td>1.000734</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>1.000650</td>
<td>0.999941</td>
<td>1.000359</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.999919</td>
<td>0.999926</td>
<td>0.999909</td>
</tr>
</tbody>
</table>

*Note. DU(3) = Discrete uniform with hyperparameter 3
N = Sample size
T = Time points*
Table 11

*The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Informative Prior (Poisson with \( \lambda = 4 \))*

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoI(4)</td>
<td>15</td>
<td>3</td>
<td>0.999933</td>
<td>0.999950</td>
<td>0.999951</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.001193</td>
<td>0.999983</td>
<td>1.000184</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.999916</td>
<td>0.999989</td>
<td>1.000598</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>0.999908</td>
<td>1.000164</td>
<td>1.000079</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.999924</td>
<td>0.999903</td>
<td>0.999905</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.99990</td>
<td>0.99997</td>
<td>1.000203</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>0.999919</td>
<td>1.000053</td>
<td>0.999908</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.999946</td>
<td>1.000128</td>
<td>1.000044</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.999986</td>
<td>0.999906</td>
<td>1.000036</td>
</tr>
</tbody>
</table>

*Note.* PoI(4) = Poisson distribution with \( \lambda = 4 \)
N = Sample size
T = Time points
Table 12

*The Estimated Posterior Variance (\( \hat{R} \)) of a scalar for the Covariance Matrix (\( \phi \)) for Noninformative Prior (Discrete Uniform with parameter 4)*

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(4)</td>
<td>15</td>
<td>3</td>
<td>0.999984</td>
<td>1.000004</td>
<td>0.999939</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>0.999909</td>
<td>0.999908</td>
<td>1.000550</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.999909</td>
<td>0.999922</td>
<td>1.000064</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>0.999919</td>
<td>0.999980</td>
<td>0.999903</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>1.000078</td>
<td>0.999911</td>
<td>0.999962</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>1.000148</td>
<td>1.000460</td>
<td>1.000022</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>1.000251</td>
<td>0.999903</td>
<td>1.000028</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>1.000064</td>
<td>1.000302</td>
<td>1.000000</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.999915</td>
<td>1.000013</td>
<td>1.000143</td>
</tr>
</tbody>
</table>

*Note.* DU(4) = Discrete uniform with hyperparameter 4  
N = Sample size  
T = Time points
### Table 13

*The Estimated Posterior Variance ($\hat{R}$) of a scalar for the Covariance Matrix ($\phi$) for Informative Prior (Poisson with $\lambda = 5$)*

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(5)</td>
<td>15</td>
<td>3</td>
<td>1.000011</td>
<td>1.000011</td>
<td>0.999919</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>1.000035</td>
<td>1.000089</td>
<td>0.999906</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>1.001570</td>
<td>1.000025</td>
<td>1.000025</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>0.999936</td>
<td>0.999936</td>
<td>0.999900</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>1.000243</td>
<td>0.999922</td>
<td>1.000692</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>0.999901</td>
<td>1.000446</td>
<td>1.000138</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>0.999908</td>
<td>0.999902</td>
<td>0.999915</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>1.000965</td>
<td>0.999938</td>
<td>0.999905</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>1.000037</td>
<td>1.000213</td>
<td>1.000513</td>
</tr>
</tbody>
</table>

*Note.* Poi(5) = Poisson distribution with $\lambda = 5$

N = Sample size

T = Time points
In general, convergence diagnosis for $K$ and $\phi$ in GMM was not a problematic issue in the current project when the Metropolis-Hastings technique based on MCMC was used. Both informative and noninformative priors, based on the number of component parameters $K$, had the similar representative region regarding parameter space. Therefore, whether or not the prior information on $K$ was known did not affect the value of $\hat{R}$ in this case. Also, sample size did not appear to have an effect on the diagnostics of $\hat{R}$, when the simulation had been run long enough. Since the key purpose of Markov chain simulation is to create a specific posterior or stationary distribution of the unknown parameters, it

Table 14

*The Estimated Posterior Variance ($\hat{R}$) of a scalar for the Covariance Matrix ($\phi$) for Noninformative Prior (Discrete Uniform with parameter 5)*

<table>
<thead>
<tr>
<th>Prior</th>
<th>N</th>
<th>T</th>
<th>2-class</th>
<th>3-class</th>
<th>4-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(5)</td>
<td>15</td>
<td>3</td>
<td>1.000907</td>
<td>0.999942</td>
<td>1.000057</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 0.999956</td>
<td>0.999914</td>
<td>1.000001</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 1.000376</td>
<td>0.999902</td>
<td>1.000601</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>3</td>
<td>1.000026</td>
<td>0.999942</td>
<td>1.000027</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 0.999907</td>
<td>1.000692</td>
<td>0.999927</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 1.000043</td>
<td>1.000154</td>
<td>1.000169</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3</td>
<td>0.999916</td>
<td>0.999903</td>
<td>1.000015</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 1.000649</td>
<td>1.000035</td>
<td>1.000111</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 0.999956</td>
<td>0.999905</td>
<td>1.000300</td>
</tr>
</tbody>
</table>

*Note.* DU(5) = Discrete uniform with hyperparameter 5  
N = Sample size  
T = Time points
was necessary to verify the convergence value of the simulated sequences when the simulation had been applied.

Therefore, it would have been necessary for the application of Markov chain simulation to run the simulation around 10,000 iterations or more to ensure the accuracy of distributions for the required parameters. In this dissertation I ran two chains with 20,000 iterations in each chain which passed the convergence testing. The results indicate that the distribution of the current draws are close to the posterior distribution. The convergence, in turn, allowed the inferential statistics to calculate the parameters of interest which are addressed in the last section of this chapter.

**The Performance of the Estimation**

In this section, the growth mixture model as shown in Equation 37 was applied to simulate data sets to demonstrate the use of a Bayesian method on the estimation for the number of latent classes or components in growth mixture models. Conditions set included different levels of sample size, waves of data or time points, and number of components specified as two-, three-, and four latent classes. Three statistics, defined earlier, based on converged simulation of posterior sampling data sets were considered. First, the average estimate over four batches with 5,000 iterations each was obtained from Equation 77,

\[
\bar{\hat{\theta}}_j = \frac{1}{4} \sum_{i=1}^{4} \hat{\theta}_{ij}.
\]

Second, empirical standard errors of scalar parameters were calculated using Equation 78,

\[
\text{empirical standard error} = \sqrt{\frac{1}{3} \sum_{i=1}^{4} (\hat{\theta}_{ij} - \bar{\hat{\theta}}_j)^2}.
\]

Third, average lower and upper limits of the 95% confidence intervals were obtained. The results for the three different data generating models with different sample sizes are presented in Tables ?? through 35. Each table shows the performance of the Bayesian method of estimation for the number of latent classes based on both noninformative (e.g., discrete uniform distribution; DU) and informative (e.g., Poisson distribution; Poi) priors.
Tables 15 and 16 shows the performance of the estimation on a 2-class mixture model with small sample size, $N = 15$. Although the true number of classes is two (i.e., $K = 2$), the mean of the posterior distribution for $K$ from the Metropolis-Hastings algorithm simulation indicates approximately three latent classes for the noninformative prior and four latent classes for the informative prior. The empirical standard errors are very small with the largest approximate value of 0.08 for GMM data with five repeated measures based on a Poisson informative prior with $\lambda = 5$. However, none of the confidence intervals contained the true value of $K$. The same estimation also holds true for the 2-class GMM with sample size of 50 (see Tables 21 and 22) and sample size of 200 (see Tables 27 and 28). It was noticed that for the less complex model, the Bayesian estimation using the Metropolis-Hastings method could not identify the correct number of latent classes, $K$, in a growth mixture model. The Bayesian approach had a tendency to over-extract the true number of latent classes in this case. The estimation on enumerating the number of latent classes always favored the 3-class model except for an estimation based on some informative priors. For example, the number of latent classes estimated was quite sensitive to the informative priors; with $\lambda = 4$ and 5 for the Poisson distribution, most of the estimation on the number of latent classes were identified approximately 3 to 4 classes. As such, no further estimation on unknown parameters in growth mixture model was needed.

Tables 15 through 20 provide the results for the growth mixture model with 2-, 3-, and 4-classes model at sample size 15, respectively. Tables 21 through 26 show the results for growth mixture model with 2-, 3-, and 4-class model at sample size 50, respectively. Also Tables 27 through 32 provide the results for growth mixture model with 2-, 3-, and 4-class model at sample size 200, respectively. Values in the aforementioned tables represent the estimation of the numbers of latent classes with lower and upper confidence
intervals including the empirical standard errors and convergences. The low values of the empirical standard error and the convergences closed to 1 representing a procedure of the estimation on the number of latent classes were correctly estimated. However, the parameter estimates focused on the aforementioned tables which are the number of latent classes, fell in the 95% confidence interval boundaries for only the 4-class growth mixture model at all sample sizes. This correct identification also happened in a 3-class growth mixture model at sample size 50 for discrete uniform with parameter 5.

The detailed summary results from the true model with three latent classes for sample size of $N = 15$ was obtained from Tables 17 and 18, $N = 50$ was obtained from Tables 23 and 24, and $N = 200$ was obtained from Tables 29 and 30. All simulation samples passed the convergence as specified earlier. The empirical standard errors of all conditions are considered negligible. The findings for the 3-class GMM varied dramatically across all sample sizes and prior information toward the unknown parameter $K$. Interestingly, at sample size 200 with both informative and noninformative priors, including all time points, none of the true values of $K$ (i.e., 3-class model) were contained within the 95% confidence interval of the estimation (see Tables 29 and 30). For sample sizes 15 and 50, the Bayesian method enumerated the number of latent classes accurately for some conditions. For the mildly complex model, the number of latent classes only achieved adequate accuracy in cases when a noninformative prior on $K$ was applied. For example, with $N = 15$, $T = 4$ based on a discrete uniform distribution with parameter 5 as reported in Tables 17 and 18, the point estimation for $K$ was 3.476 (e.g., 95% confidence interval $[CI] = 2.506477 - 4.445523$). When $N = 50$, $K$ was 3.4844 (95% confidence interval $[CI] = 2.966275 - 4.002525$) (see Tables 23 and 24).
The results for the growth mixture data with four classes for sample sizes of $N = 15$ are shown Tables 19 and 20, $N = 50$ are shown Tables 25 and 26, and $N = 200$ are shown Tables 31 and 32. All estimations for the number of latent classes fall in the range of the 95% confidence interval. When comparing the performance across sample sizes, with sample size of 200, the results yielded the narrowest interval. Likewise, a sample size of 50 resulted in a narrower range than a sample of 15. This phenomenon occurs throughout the results where 4-class models are set. The results theoretically confirmed that larger samples tend to give narrower confidence intervals for the estimation of latent classes than that of smaller samples which lead to more precise estimates.

As shown in Tables 19 and 20 with the true value of latent classes $K = 4$, at sample size of $N = 15$ for Poisson distribution with $\lambda = 3$ informative prior, for $T = 3$ the average number of latent classes was reported to be $K = 3.5894$ (95% CI, 2.6778 - 4.4930), for $T = 4$ the average number of latent classes was reported to be $K = 3.5539$ (95% CI, 2.6680 - 4.44398), and for $T = 5$ the average number of latent classes was reported to be $K = 3.6076$ (95% CI, 2.7013 - 4.5138). As for results of the estimation related to noninformative priors at the same sample size (i.e., $N = 15$), the estimation of the number of latent classes for the discrete uniform priors with parameter 5 were reported here. At time point $T = 3$, the average number of latent classes was 3.5004 (95% CI, 2.5015 - 4.4993); at $T = 4$, the average number of latent classes was 3.4494 (95% CI, 2.4847 - 4.4141); and at $T = 5$, the average number of latent classes was 3.4467 (95% CI, 2.5097 - 4.3837).

As shown in Tables 25 and 26, correct estimation in terms of 95% confidence interval for the number of latent classes, which is four, was reported for both noninformative and informative priors on $K$ for all time points and level of prior
distribution set with sample size 50. For Poisson distribution with \( \lambda = 3 \) informative prior, for \( T = 3 \) the mean number of latent classes was 3.5753 (95% CI, 3.0846 - 4.0660), for \( T = 4 \) the mean number of latent classes was 3.6022 (95% CI, 3.1039 - 4.1005), for \( T = 5 \) the mean number of latent classes was 3.5746 (95% CI, 3.0863 - 4.0629). Regarding correct estimation in terms of the 95% confidence interval, the pattern was similar to that for noninformative priors. For discrete uniform with parameter 5, with \( T = 3 \) the estimated value of latent classes was 3.5527 (95% CI, 3.0381 - 4.0673), with \( T = 4 \) the estimated value of latent classes was 3.5585 (95% CI, 3.0373 - 4.0797), and with \( T = 5 \) the estimated value of latent classes was 3.5613 (95% CI, 3.0339 - 4.0887). Finally, for growth mixture models with the true number of latent classes \( K = 4 \) for sample size 200, the pattern of correct estimation for the number of latent classes in terms of 95% confidence interval was similar to the estimation for other sample sizes (see Tables 31 and 32). When the simulation was run based on medium sample size 200 with the same previously specified conditions, for informative prior, Poisson distribution with \( \lambda = 3 \) at \( T = 3 \), the average number of latent classes was 3.6276 (95% CI, 3.3819 - 3.8733), at \( T = 4 \), the average number of latent classes was 3.6237 (95% CI, 3.3724 - 3.8748), and at \( T = 5 \), the average number of latent classes was 3.5507 (95% CI, 3.3017 - 3.7997). Additionally, there were no incorrect estimations on the number of latent classes for the true 4-class GMM in the form of 95% confidence interval for noninformative priors. For example, for discrete uniform with parameter 5, for \( T = 3 \) the average number of latent classes was 3.5905 (95% CI, 3.3351 - 3.8459), for \( T = 4 \) the average number of latent classes was 3.5580 (95% CI, 3.3054 - 3.8106), and for \( T = 5 \) the average number of latent classes was 3.6291 (95% CI, 3.3773 - 3.8809).
In comparison of all four mixtures received, the informative and noninformative priors for the number of latent classes parameter under Metropolis-Hastings sampling, produced the following results. These are presented as 95 % confidence intervals as previously mentioned in Tables 25 and 26. The prior information, concerning the unknown parameter of interest, affected the estimation in that when informative priors were used, it was more likely to obtain a narrower confidence interval than the one with noninformative priors. The narrower the interval indicates the more precise estimate for the parameter.

Therefore, based on the Metropolis-Hastings algorithm sampling method with a 4-class model, representing a complex model the 95 % confidence interval included the true value of the latent classes for each study conducted, with values falling somewhere between the lower and upper bound of the estimation. The performance of the estimates, when a 4-class growth mixture model was used, was also confirmed by the empirical standard error which showed small values for all conditions on both informative and noninformative priors. The results indicated that the estimation was accurate when the true number of latent classes was 4.
Table 15

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with N = 15 for Informative Prior (Poisson with \( \lambda = 3, 4, 5 \))

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>3</td>
<td>2</td>
<td>3.5874</td>
<td>2.7008</td>
<td>4.4740</td>
<td>0.017320</td>
<td>0.999949</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.5933</td>
<td>2.6831</td>
<td>4.5053</td>
<td>0.013510</td>
<td>0.999928</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.6241</td>
<td>2.7331</td>
<td>4.5151</td>
<td>0.015357</td>
<td>0.999938</td>
</tr>
<tr>
<td>Poi(4)</td>
<td>3</td>
<td>2</td>
<td>4.0070</td>
<td>3.03265</td>
<td>4.9813</td>
<td>0.033717</td>
<td>1.000053</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.9726</td>
<td>3.0021</td>
<td>4.9431</td>
<td>0.061199</td>
<td>1.000410</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>4.0173</td>
<td>3.0498</td>
<td>4.9848</td>
<td>0.005713</td>
<td>1.000712</td>
</tr>
<tr>
<td>Poi(5)</td>
<td>3</td>
<td>2</td>
<td>4.1817</td>
<td>3.1594</td>
<td>5.2040</td>
<td>0.025288</td>
<td>0.999978</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>4.2443</td>
<td>3.2164</td>
<td>5.2122</td>
<td>0.030138</td>
<td>1.000010</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>4.0365</td>
<td>3.0475</td>
<td>5.0255</td>
<td>0.082561</td>
<td>1.000794</td>
</tr>
</tbody>
</table>

Note. Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
Table 16

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with N = 15 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>3</td>
<td>2</td>
<td>3.5294</td>
<td>2.5169</td>
<td>4.5419</td>
<td>0.062585</td>
<td>1.000390</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.4306</td>
<td>2.4416</td>
<td>4.4196</td>
<td>0.031408</td>
<td>1.000029</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.4256</td>
<td>2.4271</td>
<td>4.4241</td>
<td>0.035796</td>
<td>1.000065</td>
</tr>
<tr>
<td>DU(4)</td>
<td>3</td>
<td>2</td>
<td>3.4497</td>
<td>2.4281</td>
<td>4.4713</td>
<td>0.020669</td>
<td>0.999952</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.4429</td>
<td>2.4925</td>
<td>4.3936</td>
<td>0.061545</td>
<td>1.000437</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.4805</td>
<td>2.5066</td>
<td>4.4544</td>
<td>0.004041</td>
<td>0.999902</td>
</tr>
<tr>
<td>DU(5)</td>
<td>3</td>
<td>2</td>
<td>3.4082</td>
<td>2.4084</td>
<td>4.4080</td>
<td>0.006235</td>
<td>0.999905</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.4685</td>
<td>2.4664</td>
<td>4.4706</td>
<td>0.000808</td>
<td>0.999900</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.4429</td>
<td>2.4925</td>
<td>4.3933</td>
<td>0.061545</td>
<td>1.000437</td>
</tr>
</tbody>
</table>

Note. Time = Time points
True = The true value of latent class (K)
Est. = Parameter Estimate for latent class (K)
Emp. = Empirical standard error
Conv. = Convergence
Table 17

*Parameter Estimates, 95% Confidence Intervals, Empirical Standard Error, and Convergence for Number of Latent Classes (K) on 3-class GMM with N = 15 for Informative Prior (Poisson with $\lambda = 3, 4, 5$)*

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>3</td>
<td>3</td>
<td>3.5971</td>
<td>2.6893</td>
<td>4.5059</td>
<td>0.050691</td>
<td>1.000299</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.5933</td>
<td>2.7109</td>
<td>4.5361</td>
<td>0.008198</td>
<td>0.999910</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.6363</td>
<td>2.7182</td>
<td>4.5544</td>
<td>0.019976</td>
<td>0.999961</td>
</tr>
<tr>
<td>Poi(4)</td>
<td>3</td>
<td>3</td>
<td>3.8297</td>
<td>2.9019</td>
<td>4.7575</td>
<td>0.019976</td>
<td>0.999959</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.9190</td>
<td>2.9775</td>
<td>4.8604</td>
<td>0.015011</td>
<td>0.999932</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>4.0660</td>
<td>3.1091</td>
<td>5.0229</td>
<td>0.022170</td>
<td>0.999969</td>
</tr>
<tr>
<td>Poi(5)</td>
<td>3</td>
<td>3</td>
<td>4.1817</td>
<td>3.1594</td>
<td>5.2039</td>
<td>0.025288</td>
<td>0.999978</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>4.5664</td>
<td>3.4785</td>
<td>5.6542</td>
<td>0.009238</td>
<td>0.999909</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>4.2631</td>
<td>3.2433</td>
<td>5.2829</td>
<td>0.023209</td>
<td>0.999966</td>
</tr>
</tbody>
</table>

*Note.* Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
Table 18

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Error, and Convergence for Number of Latent Classes (K) on 3-class GMM with N = 15 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>3</td>
<td>3</td>
<td>3.4716</td>
<td>2.4357</td>
<td>4.5075</td>
<td>0.025172</td>
<td>0.999756</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.4483</td>
<td>2.4806</td>
<td>4.4159</td>
<td>0.018129</td>
<td>0.999945</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.4804</td>
<td>2.4922</td>
<td>4.4686</td>
<td>0.021015</td>
<td>0.999958</td>
</tr>
<tr>
<td>DU(4)</td>
<td>3</td>
<td>3</td>
<td>3.4553</td>
<td>2.4420</td>
<td>4.4686</td>
<td>0.007044</td>
<td>0.999906</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.4840</td>
<td>2.4912</td>
<td>4.4768</td>
<td>0.030484</td>
<td>1.000021</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.4074</td>
<td>2.3993</td>
<td>4.4155</td>
<td>0.002309</td>
<td>0.999901</td>
</tr>
<tr>
<td>DU(5)</td>
<td>3</td>
<td>3</td>
<td>3.4985</td>
<td>2.4909</td>
<td>4.5061</td>
<td>0.034525</td>
<td>1.000050</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.4760</td>
<td>2.5065</td>
<td>4.4455</td>
<td>0.047112</td>
<td>1.000202</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.0950</td>
<td>2.5475</td>
<td>4.4715</td>
<td>0.004272</td>
<td>0.999902</td>
</tr>
</tbody>
</table>

Note. Time = Time points
True = The true value of latent class (K)
Est. = Parameter Estimate for latent class (K)
Emp. = Empirical standard error
Conv. = Convergence
Table 19

*Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 4-class GMM with N = 15 for Informative Prior (Poisson with $\lambda = 3, 4, 5$)*

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>3</td>
<td>4</td>
<td>3.5854</td>
<td>2.6778</td>
<td>4.4930</td>
<td>0.006697</td>
<td>0.999907</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.5539</td>
<td>2.6680</td>
<td>4.4398</td>
<td>0.031754</td>
<td>1.000065</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.6076</td>
<td>2.7013</td>
<td>4.5138</td>
<td>0.024479</td>
<td>0.999993</td>
</tr>
<tr>
<td>Poi(4)</td>
<td>3</td>
<td>4</td>
<td>3.8297</td>
<td>2.9019</td>
<td>4.7575</td>
<td>0.019976</td>
<td>0.999959</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.8043</td>
<td>2.8552</td>
<td>4.7567</td>
<td>0.004272</td>
<td>0.999903</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.9160</td>
<td>2.9542</td>
<td>4.8778</td>
<td>0.003695</td>
<td>0.999902</td>
</tr>
<tr>
<td>Poi(5)</td>
<td>3</td>
<td>4</td>
<td>4.1820</td>
<td>3.1948</td>
<td>5.1692</td>
<td>0.049190</td>
<td>1.000218</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>4.4599</td>
<td>3.4011</td>
<td>5.5187</td>
<td>0.006119</td>
<td>0.999904</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>4.2631</td>
<td>3.0475</td>
<td>5.2829</td>
<td>0.023209</td>
<td>0.999966</td>
</tr>
</tbody>
</table>

*Note.* Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
Table 20

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 4-class GMM with N = 15 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>3</td>
<td>4</td>
<td>3.4876</td>
<td>2.4981</td>
<td>4.4771</td>
<td>0.056580</td>
<td>1.000319</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.5068</td>
<td>2.4949</td>
<td>4.5186</td>
<td>0.014087</td>
<td>0.999925</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.4294</td>
<td>2.4483</td>
<td>4.4105</td>
<td>0.008791</td>
<td>1.000319</td>
</tr>
<tr>
<td>DU(4)</td>
<td>3</td>
<td>4</td>
<td>3.5161</td>
<td>2.5512</td>
<td>4.4809</td>
<td>0.009122</td>
<td>0.999911</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.4630</td>
<td>2.4310</td>
<td>4.4949</td>
<td>0.013394</td>
<td>0.999922</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.4551</td>
<td>2.5040</td>
<td>4.4062</td>
<td>0.022517</td>
<td>0.999972</td>
</tr>
<tr>
<td>DU(5)</td>
<td>3</td>
<td>4</td>
<td>3.5004</td>
<td>2.5015</td>
<td>4.4993</td>
<td>0.021015</td>
<td>0.999957</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.4494</td>
<td>2.4847</td>
<td>4.4141</td>
<td>0.021708</td>
<td>0.999965</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.4467</td>
<td>2.5097</td>
<td>4.3837</td>
<td>0.012817</td>
<td>0.999924</td>
</tr>
</tbody>
</table>

Note. Time = Time points
True = The true value of latent class (K)
Est. = Parameter Estimate for latent class (K)
Emp. = Empirical standard error
Conv. = Convergence
<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>3</td>
<td>2</td>
<td>3.6039</td>
<td>3.1073</td>
<td>4.1005</td>
<td>0.090182</td>
<td>1.001169</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.6111</td>
<td>3.1167</td>
<td>4.1055</td>
<td>0.049306</td>
<td>1.000282</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.6241</td>
<td>3.1360</td>
<td>4.1121</td>
<td>0.015357</td>
<td>0.999938</td>
</tr>
<tr>
<td>Poi(4)</td>
<td>3</td>
<td>2</td>
<td>3.7282</td>
<td>3.2245</td>
<td>4.2319</td>
<td>0.030946</td>
<td>1.000045</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.6923</td>
<td>3.1939</td>
<td>4.1907</td>
<td>0.049537</td>
<td>1.000280</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.7902</td>
<td>3.2789</td>
<td>4.3014</td>
<td>0.013625</td>
<td>0.999927</td>
</tr>
<tr>
<td>Poi(5)</td>
<td>3</td>
<td>2</td>
<td>3.8252</td>
<td>3.3117</td>
<td>4.3387</td>
<td>0.000693</td>
<td>0.999900</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.9600</td>
<td>3.4257</td>
<td>4.5104</td>
<td>0.016628</td>
<td>0.999936</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.7707</td>
<td>3.2561</td>
<td>4.2853</td>
<td>0.015588</td>
<td>0.999352</td>
</tr>
</tbody>
</table>

*Note.* Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
Table 22

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with N = 50 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>3</td>
<td>2</td>
<td>3.4895</td>
<td>2.9675</td>
<td>4.0115</td>
<td>0.016743</td>
<td>0.999939</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.5129</td>
<td>2.9955</td>
<td>4.0302</td>
<td>0.005427</td>
<td>0.999904</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.4594</td>
<td>2.9265</td>
<td>3.9923</td>
<td>0.021477</td>
<td>0.999962</td>
</tr>
<tr>
<td>DU(4)</td>
<td>3</td>
<td>2</td>
<td>3.5505</td>
<td>3.0265</td>
<td>4.0742</td>
<td>0.050576</td>
<td>1.000258</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.5109</td>
<td>2.9944</td>
<td>4.0274</td>
<td>0.013510</td>
<td>0.999926</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>4.3838</td>
<td>2.9639</td>
<td>4.0036</td>
<td>0.085448</td>
<td>1.000939</td>
</tr>
<tr>
<td>DU(5)</td>
<td>3</td>
<td>2</td>
<td>3.5960</td>
<td>3.0779</td>
<td>4.1140</td>
<td>0.006004</td>
<td>0.999905</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.5197</td>
<td>2.9886</td>
<td>4.0508</td>
<td>0.006119</td>
<td>0.999905</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.5537</td>
<td>3.0133</td>
<td>4.0941</td>
<td>0.033139</td>
<td>1.000044</td>
</tr>
</tbody>
</table>

Note. Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
Table 23

*Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 3-class GMM with N = 50 for Informative Prior (Poisson with $\lambda = 3, 4, 5$)*

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>3</td>
<td>3</td>
<td>3.5979</td>
<td>3.0971</td>
<td>4.0987</td>
<td>0.018129</td>
<td>0.999950</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.6122</td>
<td>3.1038</td>
<td>4.1206</td>
<td>0.011547</td>
<td>0.999919</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.6395</td>
<td>3.1554</td>
<td>4.1236</td>
<td>0.004503</td>
<td>0.999903</td>
</tr>
<tr>
<td>Poi(4)</td>
<td>3</td>
<td>3</td>
<td>3.8595</td>
<td>3.3318</td>
<td>4.3872</td>
<td>0.004272</td>
<td>0.999902</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.6858</td>
<td>3.1867</td>
<td>4.1849</td>
<td>0.022401</td>
<td>0.999977</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.7045</td>
<td>3.2087</td>
<td>4.2002</td>
<td>0.034987</td>
<td>1.000091</td>
</tr>
<tr>
<td>Poi(5)</td>
<td>3</td>
<td>3</td>
<td>3.8252</td>
<td>3.3117</td>
<td>4.3387</td>
<td>0.000693</td>
<td>0.999900</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.8518</td>
<td>3.3331</td>
<td>4.3705</td>
<td>0.006235</td>
<td>0.999905</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.8136</td>
<td>3.3086</td>
<td>4.3186</td>
<td>0.069975</td>
<td>1.000638</td>
</tr>
</tbody>
</table>

*Note.* Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
### Table 24

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 3-class GMM with \( N = 50 \) for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>3</td>
<td>3</td>
<td>3.5726</td>
<td>3.0455</td>
<td>4.0997</td>
<td>0.063970</td>
<td>1.000466</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.4967</td>
<td>2.9794</td>
<td>4.0140</td>
<td>0.010508</td>
<td>0.999916</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.5585</td>
<td>3.0373</td>
<td>4.0797</td>
<td>0.013741</td>
<td>0.999919</td>
</tr>
<tr>
<td>DU(4)</td>
<td>3</td>
<td>3</td>
<td>3.5748</td>
<td>3.0542</td>
<td>4.0954</td>
<td>0.011778</td>
<td>0.999919</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.5395</td>
<td>3.0295</td>
<td>4.0495</td>
<td>0.050691</td>
<td>1.000280</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.5346</td>
<td>3.0048</td>
<td>4.0644</td>
<td>0.071591</td>
<td>1.000602</td>
</tr>
<tr>
<td>DU(5)</td>
<td>3</td>
<td>3</td>
<td>3.4985</td>
<td>2.9466</td>
<td>4.0504</td>
<td>0.034525</td>
<td>1.000050</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.4844</td>
<td>2.9663</td>
<td>4.0025</td>
<td>0.006463</td>
<td>0.999906</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.5209</td>
<td>3.0144</td>
<td>4.0274</td>
<td>0.004734</td>
<td>0.999903</td>
</tr>
</tbody>
</table>

*Note.* Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
Table 25

*Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 4-class GMM with N = 50 for Informative Prior (Poisson with \( \lambda = 3, 4, 5 \))*

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>3</td>
<td>4</td>
<td>3.5753</td>
<td>3.0846</td>
<td>4.0660</td>
<td>0.022517</td>
<td>0.999981</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.6022</td>
<td>3.1039</td>
<td>4.1005</td>
<td>0.029329</td>
<td>1.000033</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.5746</td>
<td>3.0863</td>
<td>4.0629</td>
<td>0.034641</td>
<td>1.000079</td>
</tr>
<tr>
<td>Poi(4)</td>
<td>3</td>
<td>4</td>
<td>3.6970</td>
<td>3.1901</td>
<td>4.2039</td>
<td>0.004272</td>
<td>0.999902</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.7790</td>
<td>3.2682</td>
<td>4.2898</td>
<td>0.045726</td>
<td>1.000208</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.7412</td>
<td>3.23618</td>
<td>4.246219</td>
<td>0.006004</td>
<td>0.999905</td>
</tr>
<tr>
<td>Poi(5)</td>
<td>3</td>
<td>4</td>
<td>3.7926</td>
<td>3.2757</td>
<td>4.3092</td>
<td>0.002771</td>
<td>0.999901</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.8528</td>
<td>3.3269</td>
<td>4.3786</td>
<td>0.039722</td>
<td>1.000119</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.8486</td>
<td>3.3299</td>
<td>4.3672</td>
<td>0.006928</td>
<td>0.999907</td>
</tr>
</tbody>
</table>

*Note.* Time = Time points
True = The true value of latent class (K)
Est. = Parameter Estimate for latent class (K)
Emp. = Empirical standard error
Conv. = Convergence
Table 26

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 4-class GMM with \( N = 50 \) for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>95% CI</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>3</td>
<td>4</td>
<td>3.5279</td>
<td>3.0227</td>
<td>4.0331</td>
<td>0.019976</td>
<td>0.999960</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.5898</td>
<td>3.0766</td>
<td>4.1026</td>
<td>0.009468</td>
<td>0.999913</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.5329</td>
<td>3.0143</td>
<td>4.0515</td>
<td>0.015588</td>
<td>0.999935</td>
</tr>
<tr>
<td>DU(4)</td>
<td>3</td>
<td>4</td>
<td>3.5409</td>
<td>2.9975</td>
<td>4.0843</td>
<td>0.035911</td>
<td>1.000068</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.5495</td>
<td>3.0254</td>
<td>4.0736</td>
<td>0.043070</td>
<td>1.000159</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.5011</td>
<td>3.0043</td>
<td>3.9978</td>
<td>0.024595</td>
<td>0.999942</td>
</tr>
<tr>
<td>DU(5)</td>
<td>3</td>
<td>4</td>
<td>3.5527</td>
<td>3.0381</td>
<td>4.0673</td>
<td>0.021362</td>
<td>0.999966</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.5585</td>
<td>3.0373</td>
<td>4.0797</td>
<td>0.013741</td>
<td>0.999927</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.5613</td>
<td>3.0339</td>
<td>4.0887</td>
<td>0.014203</td>
<td>0.999928</td>
</tr>
</tbody>
</table>

*Note.* Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
Table 27

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with \( N = 200 \) for Informative Prior (Poisson with \( \lambda = 3, 4, 5 \))

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>3</td>
<td>2</td>
<td>3.5846</td>
<td>3.3395</td>
<td>3.8297</td>
<td>0.034641</td>
<td>1.000092</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.5839</td>
<td>3.3418</td>
<td>3.8259</td>
<td>0.044918</td>
<td>1.000231</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.5892</td>
<td>3.3398</td>
<td>3.8386</td>
<td>0.041107</td>
<td>1.000161</td>
</tr>
<tr>
<td>Poi(4)</td>
<td>3</td>
<td>2</td>
<td>3.6763</td>
<td>3.4295</td>
<td>3.9231</td>
<td>0.036142</td>
<td>1.000106</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.7142</td>
<td>3.4563</td>
<td>3.9721</td>
<td>0.049883</td>
<td>1.000260</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.6099</td>
<td>3.3612</td>
<td>3.8586</td>
<td>0.001270</td>
<td>0.999900</td>
</tr>
<tr>
<td>Poi(5)</td>
<td>3</td>
<td>2</td>
<td>3.6522</td>
<td>3.3993</td>
<td>3.9051</td>
<td>0.068820</td>
<td>1.000612</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.7001</td>
<td>3.4435</td>
<td>3.9567</td>
<td>0.003349</td>
<td>0.999902</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.6839</td>
<td>3.4297</td>
<td>3.9381</td>
<td>0.015588</td>
<td>0.999936</td>
</tr>
</tbody>
</table>

*Note.* Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
Table 28

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 2-class GMM with \( N = 200 \) for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>3</td>
<td>2</td>
<td>3.6138</td>
<td>3.3651</td>
<td>3.8625</td>
<td>0.0115</td>
<td>0.999900</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.3852</td>
<td>3.3279</td>
<td>3.8424</td>
<td>0.0083</td>
<td>0.999910</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.5292</td>
<td>3.2787</td>
<td>3.7796</td>
<td>0.0159</td>
<td>0.999939</td>
</tr>
<tr>
<td>DU(4)</td>
<td>3</td>
<td>2</td>
<td>3.6119</td>
<td>3.3553</td>
<td>3.8685</td>
<td>0.0102</td>
<td>0.999915</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.5728</td>
<td>3.3204</td>
<td>3.8325</td>
<td>0.0043</td>
<td>0.999903</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.6015</td>
<td>3.3496</td>
<td>3.8534</td>
<td>0.0162</td>
<td>0.999940</td>
</tr>
<tr>
<td>DU(5)</td>
<td>3</td>
<td>2</td>
<td>3.5652</td>
<td>3.3134</td>
<td>3.8170</td>
<td>0.0902</td>
<td>1.001137</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>3.6332</td>
<td>3.3769</td>
<td>3.8895</td>
<td>0.0212</td>
<td>0.999966</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3.5812</td>
<td>3.3253</td>
<td>3.8371</td>
<td>0.0436</td>
<td>1.000179</td>
</tr>
</tbody>
</table>

Note. Time = Time points
True = The true value of latent class (K)
Est. = Parameter Estimate for latent class (K)
Emp. = Empirical standard error
Conv. = Convergence
Table 29

*Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 3-class GMM with N = 200 for Informative Prior (Poisson with $\lambda = 3, 4, 5$)*

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>3</td>
<td>3</td>
<td>3.5933</td>
<td>3.3471</td>
<td>3.8395</td>
<td>0.019514</td>
<td>0.999960</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.6726</td>
<td>3.4179</td>
<td>3.9273</td>
<td>0.042493</td>
<td>1.000167</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.5632</td>
<td>3.3157</td>
<td>3.8107</td>
<td>0.047574</td>
<td>1.000255</td>
</tr>
<tr>
<td>Poi(4)</td>
<td>3</td>
<td>3</td>
<td>3.6461</td>
<td>3.3987</td>
<td>3.8935</td>
<td>0.074940</td>
<td>1.000782</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.6884</td>
<td>3.4341</td>
<td>3.9427</td>
<td>0.011316</td>
<td>0.999919</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.6642</td>
<td>3.4155</td>
<td>3.9129</td>
<td>0.033255</td>
<td>1.000072</td>
</tr>
<tr>
<td>Poi(5)</td>
<td>3</td>
<td>3</td>
<td>3.3710</td>
<td>3.4827</td>
<td>3.9793</td>
<td>0.040414</td>
<td>1.000154</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.7019</td>
<td>3.4424</td>
<td>3.9614</td>
<td>0.039837</td>
<td>1.000126</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.6801</td>
<td>3.4297</td>
<td>3.9308</td>
<td>0.003579</td>
<td>0.999902</td>
</tr>
</tbody>
</table>

*Note.* Time = Time points  
True = The true value of latent class (K)  
Est. = Parameter Estimate for latent class (K)  
Emp. = Empirical standard error  
Conv. = Convergence
Table 30

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 3-class GMM with N = 200 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>3</td>
<td>3</td>
<td>3.5882</td>
<td>3.3321</td>
<td>3.8443</td>
<td>0.024710</td>
<td>0.999989</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.5782</td>
<td>3.3266</td>
<td>3.8238</td>
<td>0.030253</td>
<td>1.000042</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.5151</td>
<td>3.2653</td>
<td>3.7649</td>
<td>0.007505</td>
<td>0.999909</td>
</tr>
<tr>
<td>DU(4)</td>
<td>3</td>
<td>3</td>
<td>3.6597</td>
<td>3.4018</td>
<td>3.9175</td>
<td>0.027366</td>
<td>1.000008</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.5725</td>
<td>3.3214</td>
<td>3.8236</td>
<td>0.022286</td>
<td>0.999976</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.5682</td>
<td>3.3173</td>
<td>3.7085</td>
<td>0.015473</td>
<td>0.999936</td>
</tr>
<tr>
<td>DU(5)</td>
<td>3</td>
<td>3</td>
<td>3.5792</td>
<td>3.3262</td>
<td>3.8322</td>
<td>0.026096</td>
<td>1.000002</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3.6112</td>
<td>3.3558</td>
<td>3.8666</td>
<td>0.004619</td>
<td>0.999903</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3.5568</td>
<td>3.3048</td>
<td>3.8088</td>
<td>0.078289</td>
<td>1.000828</td>
</tr>
</tbody>
</table>

*Note. Time = Time points
True = The true value of latent class (K)
Est. = Parameter Estimate for latent class (K)
Emp. = Empirical standard error
Conv. = Convergence*
Table 31

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 4-class GMM with N = 200 for Informative Prior (Poisson with $\lambda = 3, 4, 5$)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poi(3)</td>
<td>3</td>
<td>4</td>
<td>3.6276</td>
<td>3.3819</td>
<td>3.8733</td>
<td>0.036258</td>
<td>1.000109</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.6237</td>
<td>3.3724</td>
<td>3.8748</td>
<td>0.084409</td>
<td>1.000985</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.5507</td>
<td>3.3017</td>
<td>3.7997</td>
<td>0.019514</td>
<td>0.999959</td>
</tr>
<tr>
<td>Poi(4)</td>
<td>3</td>
<td>4</td>
<td>3.6409</td>
<td>3.3924</td>
<td>3.8894</td>
<td>0.023671</td>
<td>0.999987</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.6691</td>
<td>3.4155</td>
<td>3.9227</td>
<td>0.018129</td>
<td>0.999949</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.6581</td>
<td>3.4065</td>
<td>3.9097</td>
<td>0.012355</td>
<td>0.999923</td>
</tr>
<tr>
<td>Poi(5)</td>
<td>3</td>
<td>4</td>
<td>3.7252</td>
<td>3.4738</td>
<td>3.9766</td>
<td>0.015473</td>
<td>0.999936</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.6190</td>
<td>3.3704</td>
<td>3.8676</td>
<td>0.028406</td>
<td>1.000025</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.6903</td>
<td>3.4414</td>
<td>3.9392</td>
<td>0.017667</td>
<td>0.999948</td>
</tr>
</tbody>
</table>

Note. Time = Time points
True = The true value of latent class (K)
Est. = Parameter Estimate for latent class (K)
Emp. = Empirical standard error
Conv. = Convergence
Table 32

Parameter Estimates, 95% Confidence Intervals, Empirical Standard Errors, and Convergence Rates for Number of Latent Classes (K) on 4-class GMM with N = 200 for Noninformative Prior (Discrete Uniform with parameter 3, 4, 5)

<table>
<thead>
<tr>
<th>Dist.</th>
<th>Time</th>
<th>True</th>
<th>Est.</th>
<th>Lower</th>
<th>Upper</th>
<th>Emp.</th>
<th>Conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DU(3)</td>
<td>3</td>
<td>4</td>
<td>3.5924</td>
<td>3.3450</td>
<td>3.8398</td>
<td>0.023094</td>
<td>0.999984</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.5313</td>
<td>3.2789</td>
<td>3.7837</td>
<td>0.054617</td>
<td>1.000350</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.5666</td>
<td>3.3115</td>
<td>3.8218</td>
<td>0.000231</td>
<td>0.999900</td>
</tr>
<tr>
<td>DU(4)</td>
<td>3</td>
<td>4</td>
<td>3.5570</td>
<td>3.3058</td>
<td>3.8082</td>
<td>0.053116</td>
<td>1.000330</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.6038</td>
<td>3.3525</td>
<td>3.8551</td>
<td>0.051038</td>
<td>1.000296</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.6137</td>
<td>3.3632</td>
<td>3.8642</td>
<td>0.053001</td>
<td>1.000033</td>
</tr>
<tr>
<td>DU(5)</td>
<td>3</td>
<td>4</td>
<td>3.5905</td>
<td>3.3351</td>
<td>3.8459</td>
<td>0.032678</td>
<td>1.000057</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>3.5580</td>
<td>3.3054</td>
<td>3.8106</td>
<td>0.015011</td>
<td>0.999934</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>3.6291</td>
<td>3.3773</td>
<td>3.8809</td>
<td>0.003579</td>
<td>0.999902</td>
</tr>
</tbody>
</table>

Note. Time = Time points
True = The true value of latent class (K)
Est. = Parameter Estimate for latent class (K)
Emp. = Empirical standard error
Conv. = Convergence

Together, the point estimates, confidence intervals, and empirical standard errors provide information to assess performance of the Bayesian method and to estimate the number of latent classes on growth mixture models. Looking across all growth mixture models considered in this study, there are a few general trends worth noting. First, the influence of the complexity of growth mixture models influences heavily the estimation of the number of components. For any of the less complex modeling settings (e.g., K = 2),
Bayesian estimation does not seem to be the appropriate method for identifying the number of components on growth mixture models. The estimates are significantly higher than the true value of 2 even though the model has converged, and empirical standard errors were acceptable. As a result, conclusions based on the less complex model are severely misleading. Also, the estimates were somewhat less accurate when the mildly complex model (i.e., 3-class model) was applied, and correctly enumerated the latent classes only in noninformative priors with small sample size. Further, the Bayesian method correctly identified the number of latent classes (e.g., all true values were contained in the 95% CI) on a complex growth mixture model across all models, sample size, time points, and prior distributions. The findings as shown in Tables 11, 14, and 17 for true values of latent class $K = 4$ are consistent with previous research (Bauer & Curran, 2003; Tofighi & Enders, 2008). Second, the number of repeated measures (3 to 5) had a relatively little influence on class estimation. This conclusion is consistent with that of Tofighi and Enders (2008). Third, it is interesting to observe that all growth mixture models with four latent correctly specified classes estimated $K$ as shown by falling in the range of the bounds. Specially, the estimation of $K$ that reviewed informative priors produced more accuracy in the estimation than noninformative for complex models. Specifying informative priors on unknown parameters indicates that the researcher has knowledge about the unknown parameters. Prior information of all unknown parameters was added but fixed in the study except for the priors on the number of mixture classes. Specifically, for conditions when the unknown parameter $K$ received both informative (i.e., Poisson) and noninformative (i.e., discrete uniform), it was of interest whether or not these priors were appropriate and would be able to identify the correct extraction of classes with ignorance about the complexity of the growth mixture models. The results in
the simulation study reported here show that the parameter values taken on the prior
distribution influence the estimation of the number of latent classes. Fourth, the minor
difference between large and small sample size in all conditions not only demonstrates that
the Bayesian method used in the study can estimate the number of components on growth
mixture model quite adequately, even with small sample sizes, but also eliminated the
problem of failing to converge.

In general, the Bayesian method performed well under conditions of the high value
of mixture of latent classes regardless of the size of the observations and the time points of
data collection. As can be seen from the findings of this dissertation, the estimation for
the number of latent classes in growth mixture models showed minor difference between
informative and noninformative priors in each case.

**Summary of the Estimation**

The summary results are concerned with the point estimation of the parameter of
interest, $K$, (i.e., latent classes) of each condition. Findings from the current dissertation
should be taken as a caution for applied researchers with an interest of using growth
mixture model as a tool to analyze the longitudinal data. As can be seen from previous
tables, the estimated numbers for $K$ show numbers with decimals, which need to
approximate to the whole number. Since the whole number of estimation needs to be
specified before estimating the other parameters in growth mixture models, a criterion
needs to be set appropriately. The cut-off value for the estimation in the current study
was $K$ or $K+1$ depending on whether the decimal in the point estimation was greater than
0.75 or not. If the decimal in an average estimate was greater than 0.75 then the number
of latent classes was $K+1$, else the number of latent classes was $K$. Such a number was set
in order to control the inflated number of latent classes in the estimation. For example, if
the estimate average was 3.5197, then the estimated number of components in this case was 3 whereas the number of component for the estimated average 3.8297 is 4.

It is clear that when a growth mixture model favors a small number of subgroups (i.e., less complex model), the Bayesian method using the Metropolis-Hastings algorithm with and without information set on the unknown parameter $K$ did not accurately enumerate the number of latent classes. The inaccuracies were seen across a specified range of time points and sample size (see Table 33). As a result, conclusions based on the less complex growth mixture model are severely misleading. However, the estimation was most likely to identify either 3 or 4 latent classes for both less and more complex models. This might be the influence of the prior distribution added on the number of latent classes parameter such as Poisson with $\lambda = 3, 4, \text{ and } 5$, and also discrete uniform distribution with parameter 3, 4 and 5. The Bayesian method, however, was accurate in certain settings. The information in Tables 33 through 35 was organized differently than those appeared in this chapter to indicate the estimated numbers of latent classes in terms of the whole numbers.
Table 33

Summary of Estimated Number of Latent Classes (K) for the True Number of Latent Classes = 2 with Different Priors

<table>
<thead>
<tr>
<th>N</th>
<th>T</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>3</td>
<td>3.5294 (3)</td>
<td>3.4497 (3)</td>
<td>3.4902 (3)</td>
<td>3.5874 (3)</td>
<td>4.007 (4)</td>
<td>4.1817 (4)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.4306 (3)</td>
<td>3.4429 (3)</td>
<td>3.4685 (3)</td>
<td>3.5933 (3)</td>
<td>3.9726 (4)</td>
<td>4.2443 (4)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.4256 (3)</td>
<td>3.4805 (3)</td>
<td>3.4429 (3)</td>
<td>3.6241 (3)</td>
<td>4.0173 (4)</td>
<td>4.0365 (4)</td>
</tr>
<tr>
<td>50</td>
<td>3</td>
<td>3.4895 (3)</td>
<td>3.5504 (3)</td>
<td>3.5960 (3)</td>
<td>3.6039 (3)</td>
<td>3.7282 (3)</td>
<td>3.8252 (4)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.5129 (3)</td>
<td>3.5109 (3)</td>
<td>3.5197 (3)</td>
<td>3.6111 (3)</td>
<td>3.6923 (3)</td>
<td>3.9600 (4)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.4594 (3)</td>
<td>4.3838 (4)</td>
<td>3.5537 (3)</td>
<td>3.6241 (3)</td>
<td>3.7902 (4)</td>
<td>3.7707 (4)</td>
</tr>
<tr>
<td>200</td>
<td>3</td>
<td>3.6138 (3)</td>
<td>3.6119 (3)</td>
<td>3.5652 (3)</td>
<td>3.5846 (3)</td>
<td>3.6763 (3)</td>
<td>3.6522 (3)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.5852 (3)</td>
<td>3.5728 (3)</td>
<td>3.6332 (3)</td>
<td>3.5839 (3)</td>
<td>3.7142 (3)</td>
<td>3.7001 (3)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.5292 (3)</td>
<td>3.6015 (3)</td>
<td>3.5812 (3)</td>
<td>3.5892 (3)</td>
<td>3.6099 (3)</td>
<td>3.6839 (3)</td>
</tr>
</tbody>
</table>

Note. The numbers in the parentheses are the approximation to the whole number using a cut-off value of 0.75.
Table 34

*Summary of Estimated Number of Latent Classes (K) for the True Number of Latent Classes = 3 with Different Priors*

<table>
<thead>
<tr>
<th>N</th>
<th>T</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>3</td>
<td>3.4716 (3)</td>
<td>3.4553 (3)</td>
<td>3.4985 (3)</td>
<td>3.5971 (3)</td>
<td>3.8297 (4)</td>
<td>4.1817 (4)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.4483 (3)</td>
<td>3.484 (3)</td>
<td>3.476 (3)</td>
<td>3.5933 (3)</td>
<td>3.919 (4)</td>
<td>4.5664 (4)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.4804 (3)</td>
<td>3.4074 (3)</td>
<td>3.5095 (3)</td>
<td>3.6363 (3)</td>
<td>4.066 (4)</td>
<td>4.2631 (4)</td>
</tr>
<tr>
<td>50</td>
<td>3</td>
<td>3.5726 (3)</td>
<td>3.5748 (3)</td>
<td>3.4985 (3)</td>
<td>3.5979 (3)</td>
<td>3.8595 (4)</td>
<td>3.8252 (4)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.4967 (3)</td>
<td>3.5395 (3)</td>
<td>3.4844 (3)</td>
<td>3.6122 (3)</td>
<td>3.6858 (3)</td>
<td>3.8518 (4)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.5585 (3)</td>
<td>3.5346 (3)</td>
<td>3.5209 (3)</td>
<td>3.6395 (3)</td>
<td>3.7045 (3)</td>
<td>3.8136 (4)</td>
</tr>
<tr>
<td>200</td>
<td>3</td>
<td>3.5882 (3)</td>
<td>3.6597 (3)</td>
<td>3.5792 (3)</td>
<td>3.5933 (3)</td>
<td>3.6461 (4)</td>
<td>3.731 (4)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.5752 (3)</td>
<td>3.5725 (3)</td>
<td>3.6112 (3)</td>
<td>3.6726 (3)</td>
<td>3.6884 (3)</td>
<td>3.7019 (3)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.5151 (3)</td>
<td>3.5682 (3)</td>
<td>3.5568 (3)</td>
<td>3.5632 (3)</td>
<td>3.6642 (3)</td>
<td>3.6801 (3)</td>
</tr>
</tbody>
</table>

*Note.* The numbers in the parentheses are the approximation to the whole number using a cut-off value of 0.75.
Table 35

Summary of Estimated Number of Latent Classes (K) for the True Number of Latent Classes = 4 with Different Priors

<table>
<thead>
<tr>
<th>N</th>
<th>T</th>
<th>Discrete Uniform</th>
<th>Poisson</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>3.4876 (3)</td>
<td>3.5161 (3)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.5068 (3)</td>
<td>3.4630 (3)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.4294 (3)</td>
<td>3.4551 (3)</td>
</tr>
<tr>
<td>50</td>
<td>3</td>
<td>3.5279 (3)</td>
<td>3.5409 (3)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.5898 (3)</td>
<td>3.5495 (3)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.5329 (3)</td>
<td>3.3011 (3)</td>
</tr>
<tr>
<td>200</td>
<td>3</td>
<td>3.5924 (3)</td>
<td>3.557 (3)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.5313 (3)</td>
<td>3.6038 (3)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.5666 (3)</td>
<td>3.6137 (3)</td>
</tr>
</tbody>
</table>

Note. The numbers in the parentheses are the approximation to the whole number using a cut-off value of 0.75.

Parameter Estimates Using Mplus and R Program

This section reports the parameter estimates for some conditions of the accuracy of the estimation for the number of latent classes. The focus is on general trends, with some specific results. R, and Mplus syntax are available in Appendix A, and B, respectively.

Growth mixture models were fit using Mplus. It should be noted that in Mplus the number of latent classes K needs to be specified before fitting the model. Once the number of latent classes was specified in Mplus, the next step in estimation was to
estimate the other parameters in the growth mixture model. In the R program using the Bayesian approach, however, the number of components need not be specified as it can estimate the number of latent classes.

First, three growth mixture models were fit to each selected data to ensure the accuracy of the estimation in the proposed R routine for correctly identifying the number of latent classes on GMMs. For example, the 2-, 3-, and 4-class models with five occasions and sample size \( N = 15, 50, \) and 200 were fit using Mplus. The 3-, 4-, and 5-class models with four occasions and sample size \( N = 15, 50, \) and 200 were also fit using Mplus. For each run, the values of the fit indices were obtained from the Mplus output, and were saved for further analysis. The results are summarized in Table 36.

As shown in Table 36, small sample size \( N = 15 \) failed to produce fit indices. Both 2-class AIC and 2-class BIC were lower than those of the latent classes specified in the data. Results classified a fit index as incorrectly identifying the proper number of latent classes. However, the likelihood statistics and information criteria-based indexes were slightly different among all cases shown in Table 36.
Table 36

*Model Fit Statistics for Different Numbers of Latent Classes*

<table>
<thead>
<tr>
<th></th>
<th>AIC</th>
<th>BIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM model</td>
<td>15</td>
<td>50</td>
</tr>
<tr>
<td>GMM with T=5, K=3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two-class</td>
<td>NA</td>
<td>710</td>
</tr>
<tr>
<td>Three-class</td>
<td>NA</td>
<td>713</td>
</tr>
<tr>
<td>Four-class</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>GMM with T=4, K=4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Three-class</td>
<td>NA</td>
<td>649</td>
</tr>
<tr>
<td>Four-class</td>
<td>NA</td>
<td>657</td>
</tr>
<tr>
<td>Five-class</td>
<td>NA</td>
<td>656</td>
</tr>
</tbody>
</table>

*Note.* T = Time points  
K = Number of latent classes

Next, 3-class GMM and 4-class GMM specifications were fitted using Mplus and R programs. The results for Mplus are presented in Tables 22 to 24. As for the R program, results are displayed in Tables 25 to 38. All cases shown in the tables were compared with the true values specified in data generation. Because the purpose of the current analysis was to demonstrate the application of proposed growth mixture models, the correctly identified numbers of latent classes were fit to the data. The growth parameters set in the data generation were between -3 and 3 for both intercept and slope of each class and the variance-covariance matrix was

\[
\begin{pmatrix}
(0.05, 1.10) & 0.7500 \\
0.7500 & (0.05, 0.68)
\end{pmatrix}
\]. The true latent intercept variances are in the interval
[0.05, 1.10], the true latent slope variances are in the interval [0.05, 0.68], and the true
covariance between the intercept and slope factors is 0.75.

Specifications for the three- and four-class growth mixture models using Mplus are
presented in Appendix B as previously mentioned. Key elements in the model
specification are the number of latent classes, the specification of each latent class
including mean intercept and slope, and covariance between the two growth parameters
(intercept and slope). Model fitting for 3-class GMM with sample size 50 are detailed. For
class 1, estimates for intercept I = -0.017, t = -0.054, $p > 0.05$ and slope S = -0.250, t =
-1.560, $p > 0.05$ means were not significant. Estimate for the latent intercept variance $\text{Var}(I)$
= 0.339, t = 0.692, $p > 0.05$ was not significant, and latent slope variance $\text{Var}(S) = 0.093$, t
= 1.604, $p > 0.05$ was significant. The covariance between the intercept and slope factors
was not significant, $R = 0.342, t = 1.841, p > 0.05$. For class 2, estimates for intercept I =
-1.557, t = -4.321, $p < 0.001$ and slope S = -1.538, t = -5.481, $p < 0.001$ means were
significant. Estimate for the latent intercept variance $\text{Var}(I) = 1.656$, t = 4.339, $p < 0.001$,
and latent slope variance $\text{Var}(S) = 1.094$, t = 3.360, $p < 0.01$ also were significant. The
covariance between the intercept and slope factors was also significant, $R = 1.199$, t =
3.426, $p < 0.01$. For class 3, estimates for intercept I = 0.940, t = 4.761, $p < 0.001$ and slope S
= 0.884, t = 7.406, $p < 0.001$ means were significant. Estimate for the latent intercept
variance $\text{Var}(I) = -0.111$, t = 1.001, $p > 0.05$ was not significant, and latent slope variance
$\text{Var}(S) = 0.048$, t = 1.256, $p < 0.01$ was significant. The covariance between the intercept
and slope factors was significant, $R = 0.293$, t = 3.066, $p < 0.01$ (results are summarized
from the outputs of the Mplus syntax in Appendix B).

To further investigate the growth parameter estimates using Mplus, the latent
intercept and variance factors as well as the covariance between intercept and slope were
compared with the true values. For example, Table 37 for \( N = 50 \) shows that the estimated intercept for latent class 1, 2, and 3 were -0.017, -1.557, and 0.940, respectively, while the true intercept value were in the interval \([-3, 3]\). Additionally, the estimated slope for latent class 1, 2, and 3 were -0.250, -1.538, and 0.884, respectively (see Table 37, \( N = 50 \)), while the true slope values were in the interval \([-3, 3]\). Latent class 1 has an intercept and slope covariance matrix of

\[
\begin{bmatrix}
0.339 & 0.342 \\
0.342 & 0.093
\end{bmatrix}
\]

latent class 2 has an intercept and slope covariance matrix of

\[
\begin{bmatrix}
1.656 & 1.199 \\
1.199 & 1.094
\end{bmatrix}
\]

latent class 3 has an intercept and slope covariance matrix of

\[
\begin{bmatrix}
-0.111 & 0.293 \\
0.293 & -0.048
\end{bmatrix}
\]

(see Table 37, \( N = 50 \)), while the true value has an intercept and slope covariance matrix of

\[
\begin{bmatrix}
(0.05, 1.10) & 0.75 \\
0.75 & (0.05, 0.68)
\end{bmatrix}
\]

In cases examined so far, estimated parameters produced both significant and not significant estimates. However, \textit{Mplus} estimates provided an acceptable coverage because all estimated values fell in between the interval on the true values except for the intercept and slope variance for class 3 of small sample size, \( N = 50 \), for 3-class growth mixture model that showed negative variance.

From Tables 38 and 39, the same conclusions can be drawn as those in the 3-class GMM for sample sizes 50 and 200. In the 4-class GMM specification, sample size 15 also failed to converge, so the estimation for growth parameters was discarded. Results are
summarized from the outputs of the Mplus syntax in Appendix B. The results show the following:

1. Estimates for the mean latent intercepts were significant for both sample sizes 50 and 200.

2. Estimates for the mean latent slopes were mostly significant except the slope for latent class 1 of a 4-class GMM at sample size 50 and the slope for latent class 2 at sample size 200.

3. Estimates for the variances of intercept and slope for class 2 of both sample size 50 and 200 were not significant.

4. The covariances between intercept and slope factors for most of latent classes were not significant.

In both of the true models, the estimation of the latent growth means for the intercept and slope factors as well as other growth parameters (e.g., intercept and slope variances, the covariance between intercept and slope factors) is accurate for most conditions. The estimation for some latent means for sample size 200 was out of bounds which were considered inaccurate while factor variance estimates for small sample size, $N = 15$, showed a Heywood case (i.e., negative error variance). In the Mplus standard output, negative error variance was considered nonconverged. This phenomenon was considered a normal variation of sampling from small sample sizes (M. Liu & Hancock, 2014). In addition, the results showed no standard errors for sample size 15 due to failure to converge of the solutions.
Table 37

*Estimates of Parameters from Simulated Data Set for the Growth Mixture Model with 5 Time Points and 3-class Model Using Mplus*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>N = 15</th>
<th>SE</th>
<th>N = 50</th>
<th>SE</th>
<th>N = 200</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters: C1</strong>&lt;br&gt;Intercept (I)</td>
<td>0.902</td>
<td>-</td>
<td>-0.017</td>
<td>0.323</td>
<td>3.889</td>
<td>0.140</td>
</tr>
<tr>
<td>Slope (S)</td>
<td>0.653</td>
<td>-</td>
<td>-0.250</td>
<td>0.160</td>
<td>3.644</td>
<td>0.177</td>
</tr>
<tr>
<td>Var(I)</td>
<td>0.641</td>
<td>-</td>
<td>0.339</td>
<td>0.489</td>
<td>0.316</td>
<td>0.100</td>
</tr>
<tr>
<td>Var(S)</td>
<td>0.523</td>
<td>-</td>
<td>0.093</td>
<td>0.058</td>
<td>0.838</td>
<td>0.199</td>
</tr>
<tr>
<td>Cov(IS)</td>
<td>0.653</td>
<td>-</td>
<td>0.342</td>
<td>0.186</td>
<td>0.713</td>
<td>0.135</td>
</tr>
<tr>
<td><strong>Parameters: C2</strong>&lt;br&gt;Intercept (I)</td>
<td>-0.559</td>
<td>-</td>
<td>-1.557</td>
<td>0.360</td>
<td>-2.702</td>
<td>0.140</td>
</tr>
<tr>
<td>Slope (S)</td>
<td>-0.531</td>
<td>-</td>
<td>-1.538</td>
<td>0.281</td>
<td>-2.590</td>
<td>0.101</td>
</tr>
<tr>
<td>Var(I)</td>
<td>1.338</td>
<td>-</td>
<td>1.656</td>
<td>0.382</td>
<td>1.228</td>
<td>0.273</td>
</tr>
<tr>
<td>Var(S)</td>
<td>1.341</td>
<td>-</td>
<td>1.094</td>
<td>0.326</td>
<td>0.680</td>
<td>0.150</td>
</tr>
<tr>
<td>Cov(IS)</td>
<td>1.351</td>
<td>-</td>
<td>1.199</td>
<td>0.350</td>
<td>0.860</td>
<td>0.191</td>
</tr>
<tr>
<td><strong>Parameters: C3</strong>&lt;br&gt;Intercept (I)</td>
<td>2.045</td>
<td>-</td>
<td>0.940</td>
<td>0.198</td>
<td>3.229</td>
<td>0.134</td>
</tr>
<tr>
<td>Slope (S)</td>
<td>2.248</td>
<td>-</td>
<td>0.884</td>
<td>0.119</td>
<td>3.298</td>
<td>0.090</td>
</tr>
<tr>
<td>Var(I)</td>
<td>0.702</td>
<td>-</td>
<td>-0.111</td>
<td>0.111</td>
<td>1.079</td>
<td>0.191</td>
</tr>
<tr>
<td>Var(S)</td>
<td>0.304</td>
<td>-</td>
<td>-0.048</td>
<td>0.038</td>
<td>0.494</td>
<td>0.096</td>
</tr>
<tr>
<td>Cov(IS)</td>
<td>0.237</td>
<td>-</td>
<td>0.293</td>
<td>0.096</td>
<td>0.673</td>
<td>0.123</td>
</tr>
</tbody>
</table>

*Note.* SE = Standard error
Table 38

*Estimates of Parameters from Simulated Data Set for the Growth Mixture Model with 4 Time Points and 4-class Model Using Mplus*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>SE</th>
<th>Estimate</th>
<th>SE</th>
<th>Estimate</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N = 15</td>
<td></td>
<td>N = 50</td>
<td></td>
<td>N = 200</td>
<td></td>
</tr>
<tr>
<td>Parameters: C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept (I)</td>
<td>3.263</td>
<td>-</td>
<td>1.621</td>
<td>0.266</td>
<td>0.978</td>
<td>0.127</td>
</tr>
<tr>
<td>Slope (S)</td>
<td>-0.307</td>
<td>-</td>
<td>2.021</td>
<td>0.163</td>
<td>1.208</td>
<td>0.127</td>
</tr>
<tr>
<td>Var(I)</td>
<td>0.902</td>
<td>-</td>
<td>1.172</td>
<td>0.512</td>
<td>0.274</td>
<td>0.152</td>
</tr>
<tr>
<td>Var(S)</td>
<td>0.166</td>
<td>-</td>
<td>0.636</td>
<td>0.160</td>
<td>0.491</td>
<td>0.167</td>
</tr>
<tr>
<td>Cov(IS)</td>
<td>-0.394</td>
<td>-</td>
<td>0.543</td>
<td>0.250</td>
<td>0.104</td>
<td>0.088</td>
</tr>
<tr>
<td>Parameters: C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept (I)</td>
<td>1.869</td>
<td>-</td>
<td>-0.715</td>
<td>0.345</td>
<td>2.040</td>
<td>0.380</td>
</tr>
<tr>
<td>Slope (S)</td>
<td>3.033</td>
<td>-</td>
<td>-0.611</td>
<td>0.060</td>
<td>0.113</td>
<td>0.743</td>
</tr>
<tr>
<td>Var(I)</td>
<td>-0.001</td>
<td>-</td>
<td>0.551</td>
<td>0.367</td>
<td>0.324</td>
<td>0.278</td>
</tr>
<tr>
<td>Var(S)</td>
<td>0.081</td>
<td>-</td>
<td>-0.120</td>
<td>0.054</td>
<td>1.534</td>
<td>1.054</td>
</tr>
<tr>
<td>Cov(IS)</td>
<td>0.045</td>
<td>-</td>
<td>0.126</td>
<td>0.134</td>
<td>0.785</td>
<td>0.538</td>
</tr>
</tbody>
</table>

*Note.* SE = Standard error
Table 39

*Estimates of Parameters from Simulated Data Set for the Growth Mixture Model with 4 Time Points and 4-class Model Using Mplus (continued)*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( N = 15 )</th>
<th>( N = 50 )</th>
<th>( N = 200 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Estimate</td>
<td>SE</td>
<td>Estimate</td>
</tr>
<tr>
<td>Parameters: C3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept (I)</td>
<td>-0.238</td>
<td>-</td>
<td>-1.717</td>
</tr>
<tr>
<td>Slope (S)</td>
<td>-0.902</td>
<td>-</td>
<td>-0.562</td>
</tr>
<tr>
<td>Var(I)</td>
<td>0.828</td>
<td>-</td>
<td>0.575</td>
</tr>
<tr>
<td>Var(S)</td>
<td>0.677</td>
<td>-</td>
<td>1.970</td>
</tr>
<tr>
<td>Cov(IS)</td>
<td>0.466</td>
<td>-</td>
<td>0.130</td>
</tr>
<tr>
<td>Parameters: C4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept (I)</td>
<td>1.551</td>
<td>-</td>
<td>1.223</td>
</tr>
<tr>
<td>Slope (S)</td>
<td>1.370</td>
<td>-</td>
<td>0.343</td>
</tr>
<tr>
<td>Var(I)</td>
<td>-0.202</td>
<td>-</td>
<td>0.053</td>
</tr>
<tr>
<td>Var(S)</td>
<td>0.032</td>
<td>-</td>
<td>-0.040</td>
</tr>
<tr>
<td>Cov(IS)</td>
<td>0.122</td>
<td>-</td>
<td>0.061</td>
</tr>
</tbody>
</table>

*Note.* SE = Standard error

For the R program, the growth mixture model parameters for data generation were set at the same values as used for running Mplus. A number of the 2-class solutions failed to estimate the number of latent classes, so the parameter estimates were not able to be obtained from the R procedure. Overall, across independent variable conditions, several general conclusions were evident from the results (see Tables 40 to 53).
Tables 40 and 41 display parameter estimates for growth mixture model when informative priors, Poisson with hyperparameter \( \lambda = 4 \), were placed on the number of latent class parameter. Average estimates of each unknown parameter in growth mixture models were reported. The point estimate for the number of latent class showed the value of 3.916 while the true number of latent class was 4. The scalar value specified in the residual variance of observed values (\( \phi \)) was 0.998, while the true \( \phi \) was 0.30. This parameter estimate using developed R program was far from the true value. The class separation level was 0.10/0.20/0.30/0.40 that was misspecified to a mixture class model under the 0.25/0.25/0.25/0.25 mixture proportions. The estimated intercept and slope factors were produced in the form of vector in R program. The estimated growth factors for latent class 1, 2, 3, and 4 were

\[
\begin{bmatrix}
0.9784 & 1.0274 \\
1.0050 & 0.9563 \\
1.0021 & 1.0103
\end{bmatrix}, \quad \begin{bmatrix}
1.0141 & 1.0084 \\
0.9980 & 0.9980 \\
0.9980 & 0.9980
\end{bmatrix},
\]

\( \psi_1 \) and \( \psi_2 \), respectively, while the interval of both intercept and slope factors for the true values were (-3, 3). The variance of intercept and slope including covariance between intercept and slope of each latent class in growth mixture model were produced in the form of matrix in developed R program. As can be seen from Tables 42 and 41, the covariance matrix for each class was reported as

\[
\begin{bmatrix}
0.6045 & -0.0071 \\
-0.0071 & 0.6217
\end{bmatrix}, \quad \begin{bmatrix}
0.4497 & 0.1527 \\
0.1537 & 0.4420
\end{bmatrix}, \quad \begin{bmatrix}
0.6051 & 0.0016 \\
0.0016 & 0.6163
\end{bmatrix}, \quad \begin{bmatrix}
0.4416 & 0.1421 \\
0.1560 & 0.4368
\end{bmatrix},
\]

\( \psi_3 \) and \( \psi_4 \), respectively, comparing to the true value of \( \Psi = \begin{bmatrix}
(0.05, 1.10) & 0.75 \\
0.75 & (0.05, 0.68)
\end{bmatrix} \). The growth factors and covariance between latent intercept and slope estimates provided
acceptable coverage because all estimated values included in the interval of the true values. In these results, the class membership indicator ($z_i$) and the latent variable ($\eta_i$) were also calculated using developed R program, the estimation can be seen in Appendix C. From Tables 42 through 53, the same conclusions can be drawn as those in the three-, and four-class GMM for sample sizes 15, 50 and 200.

In general, regarding performance differences under prior distributions toward the number of latent classes, sample sizes, time points, and two-, three-, and four-class growth mixture models specified, five general findings were noted: 1) a scalar parameter, $\phi$, was not very close to the true values, 2) the estimated class proportions for four-class model were incorrectly identified but for three-class model, the estimated proportions were fairly close to the true values, 3) mean intercept and slope growth factors contained in the true interval specified, 4) intercept and slope variances were contained in the interval previously specified, and 5) covariances between intercept and slope were incorrectly estimated.
Table 40

Parameter Estimates for Growth Mixture Models with Poisson Priors ($\lambda = 4$) at Sample size 15, 5 Time Points and 4-class model Using R Program

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>4</td>
<td>3.9160</td>
</tr>
<tr>
<td>$\Phi (\Phi)$</td>
<td>0.30</td>
<td>0.9998</td>
</tr>
<tr>
<td>$z$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>See Appendix C (Procedure 18)</td>
</tr>
</tbody>
</table>

Parameters: C1

$\Pi (\pi)$

<table>
<thead>
<tr>
<th></th>
<th>0.25</th>
<th>0.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>growth factors$^a$</td>
<td>[ (-3, 3) (-3, 3) ]</td>
<td>[ 0.9784 1.0274 ]</td>
</tr>
<tr>
<td>$\Psi (\Psi)^b$</td>
<td>[ (0.05,1.10) 0.7500 ]</td>
<td>[ 0.6045 -0.0071 ]</td>
</tr>
<tr>
<td></td>
<td>0.7500 (0.05,0.68)</td>
<td>-0.0071 0.6217</td>
</tr>
<tr>
<td>$\Eta$</td>
<td></td>
<td>See Appendix C (Procedure 18)</td>
</tr>
</tbody>
</table>

Parameters: C2

$\Pi (\pi)$

<table>
<thead>
<tr>
<th></th>
<th>0.25</th>
<th>0.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>growth factors$^a$</td>
<td>[ (-3, 3) (-3, 3) ]</td>
<td>[ 1.0141 1.0084 ]</td>
</tr>
<tr>
<td>$\Psi (\Psi)^b$</td>
<td>[ (0.05,1.10) 0.7500 ]</td>
<td>[ 0.4497 0.1527 ]</td>
</tr>
<tr>
<td></td>
<td>0.7500 (0.05,0.68)</td>
<td>0.1537 0.4420</td>
</tr>
<tr>
<td>$\Eta$</td>
<td></td>
<td>See Appendix C (Procedure 18)</td>
</tr>
</tbody>
</table>

Note. $^a$ = Vector of intercept and slope  
$^b$ = Covariance matrix
Table 41

*Parameter Estimates for Growth Mixture Models with Poisson Priors (λ = 4) at Sample size 15, 5 Time Points and 4-class model Using R Program (continued)*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters: C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pi (π)</td>
<td>0.25</td>
<td>0.30</td>
</tr>
<tr>
<td>growth factors$^a$</td>
<td>$[-3,3] [-3,3]$</td>
<td>$[1.0050 0.9563]$</td>
</tr>
<tr>
<td>Psi (Ψ)$^b$</td>
<td>$[0.05,1.10] 0.7500$</td>
<td>$[0.6051 0.0016]$</td>
</tr>
<tr>
<td>Eta</td>
<td>See Appendix C (Procedure 18)</td>
<td></td>
</tr>
<tr>
<td>Parameters: C4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pi (π)</td>
<td>0.25</td>
<td>0.40</td>
</tr>
<tr>
<td>growth factors$^a$</td>
<td>$[-3,3] [-3,3]$</td>
<td>$[1.0021 1.0103]$</td>
</tr>
<tr>
<td>Psi (Ψ)$^b$</td>
<td>$[0.05,1.10] 0.7500$</td>
<td>$[0.4416 0.1421]$</td>
</tr>
<tr>
<td>Eta</td>
<td>See Appendix C (Procedure 18)</td>
<td></td>
</tr>
</tbody>
</table>

*Note. a = Vector of intercept and slope  
  b = Covariance matrix*
Table 42

Parameter Estimates for Growth Mixture Models with Poisson Priors (λ = 5) at Sample size 15, 3 Time Points and 4-class model Using R Program

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>4</td>
<td>4.182</td>
</tr>
<tr>
<td>Phi (Φ)</td>
<td>0.30</td>
<td>0.9999</td>
</tr>
<tr>
<td>z</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>See Appendix C (Procedure 21)</td>
</tr>
</tbody>
</table>

Parameters: C1

<table>
<thead>
<tr>
<th>Pi (π)</th>
<th>0.25</th>
<th>0.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>growth factors&lt;sup&gt;a&lt;/sup&gt;</td>
<td>[ (-3, 3) \begin{pmatrix} \begin{pmatrix} (0.05, 1.10) &amp; 0.7500 \ 0.7500 &amp; 0.05, 0.68 \end{pmatrix} \end{pmatrix} ]</td>
<td>[ \begin{pmatrix} 1.0210 &amp; 1.0416 \ 0.6113 &amp; 0.0169 \end{pmatrix} ]</td>
</tr>
</tbody>
</table>

Parameters: C2

<table>
<thead>
<tr>
<th>Pi (π)</th>
<th>0.25</th>
<th>0.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>growth factors&lt;sup&gt;a&lt;/sup&gt;</td>
<td>[ (-3, 3) \begin{pmatrix} \begin{pmatrix} (0.05, 1.10) &amp; 0.7500 \ 0.7500 &amp; 0.05, 0.68 \end{pmatrix} \end{pmatrix} ]</td>
<td>[ \begin{pmatrix} 1.0034 &amp; 1.0305 \ 0.4595 &amp; 0.1569 \end{pmatrix} ]</td>
</tr>
</tbody>
</table>

Note.  
a = Vector of intercept and slope  
b = Covariance matrix
Table 43

Parameter Estimates for Growth Mixture Models with Poisson Priors ($\lambda = 5$) at Sample size 15, 3 Time Points and 4-class model Using R Program (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters: C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Pi (\pi)$</td>
<td>0.25</td>
<td>0.30</td>
</tr>
<tr>
<td>growth factors$^a$</td>
<td>$[(-3, 3) (-3, 3)]$</td>
<td>$[0.9599 \ 0.9401]$</td>
</tr>
<tr>
<td>$\Psi (\Psi)^b$</td>
<td>$\begin{bmatrix} (0.05, 1.10) &amp; 0.7500 \ 0.7500 &amp; (0.05, 0.68) \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.5755 &amp; -0.0009 \ -0.0009 &amp; 0.6069 \end{bmatrix}$</td>
</tr>
<tr>
<td>Eta</td>
<td>See Appendix C (Procedure 21)</td>
<td></td>
</tr>
<tr>
<td>Parameters: C4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Pi (\pi)$</td>
<td>0.25</td>
<td>0.40</td>
</tr>
<tr>
<td>growth factors$^a$</td>
<td>$[(-3, 3) (-3, 3)]$</td>
<td>$[1.0437 \ 1.0094]$</td>
</tr>
<tr>
<td>$\Psi (\Psi)^b$</td>
<td>$\begin{bmatrix} (0.05, 1.10) &amp; 0.7500 \ 0.7500 &amp; (0.05, 0.68) \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.4597 &amp; 0.1506 \ 0.1511 &amp; 0.4431 \end{bmatrix}$</td>
</tr>
<tr>
<td>Eta</td>
<td>See Appendix C (Procedure 21)</td>
<td></td>
</tr>
</tbody>
</table>

Note. $a =$ Vector of intercept and slope  
$b =$ Covariance matrix
Table 44

Parameter Estimates for Growth Mixture Models with Poisson Priors ($\lambda = 5$) at Sample size 50, 4 Time Points and 4-class model Using R Program

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>4</td>
<td>3.8528</td>
</tr>
<tr>
<td>Phi ($\Phi$)</td>
<td>0.30</td>
<td>0.9949</td>
</tr>
<tr>
<td>z</td>
<td>See Appendix C (Procedure 51)</td>
<td></td>
</tr>
</tbody>
</table>

Parameters: C1

\[
\begin{align*}
\text{Pi (}\pi\text{)} & \quad 0.25 & \quad 0.10 \\
\text{growth factors}\,^a & \quad \begin{bmatrix} -3, 3 & -3, 3 \end{bmatrix} & \quad \begin{bmatrix} 0.9789 & 1.0171 \end{bmatrix} \\
\text{Psi (}\Psi\,^b & \quad \begin{bmatrix} 0.05, 1.10 & 0.7500 \end{bmatrix} & \quad \begin{bmatrix} 0.6076 & 0.0036 \end{bmatrix} \\
\text{Eta} & \quad \text{See Appendix C (Procedure 51)} & \\
\end{align*}
\]

Parameters: C2

\[
\begin{align*}
\text{Pi (}\pi\text{)} & \quad 0.25 & \quad 0.20 \\
\text{growth factors}\,^a & \quad \begin{bmatrix} -3, 3 & -3, 3 \end{bmatrix} & \quad \begin{bmatrix} 1.0025 & 1.0219 \end{bmatrix} \\
\text{Psi (}\Psi\,^b & \quad \begin{bmatrix} 0.05, 1.10 & 0.7500 \end{bmatrix} & \quad \begin{bmatrix} 0.6280 & -0.0076 \end{bmatrix} \\
\text{Eta} & \quad \text{See Appendix C (Procedure 51)} & \\
\end{align*}
\]

Note. a = Vector of intercept and slope  
       b = Covariance matrix
Table 45

*Parameter Estimates for Growth Mixture Models with Poisson Priors (λ = 5) at Sample size 50, 4 Time Points and 4-class model Using R Program (continued)*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**Phi (Φ)**

Parameters: C3

- **Pi (π)** 0.25 0.30
- Growth factors $^a$ \[\begin{bmatrix} (−3, 3) & (−3, 3) \end{bmatrix}\] \[\begin{bmatrix} 0.9627 & 0.9861 \end{bmatrix}\]
- **Psi (Ψ)$^b$** \[\begin{bmatrix} (0.05, 1.10) & 0.7500 \end{bmatrix}\] \[\begin{bmatrix} 0.6104 & −0.0120 \end{bmatrix}\]
- **Psi (Ψ)$^b$** \[\begin{bmatrix} 0.7500 & (0.05, 0.68) \end{bmatrix}\] \[\begin{bmatrix} −0.0120 & 0.5896 \end{bmatrix}\]

**Eta** See Appendix C (Procedure 51)

Parameters: C4

- **Pi (π)** 0.25 0.40
- Growth factors $^a$ \[\begin{bmatrix} (−3, 3) & (−3, 3) \end{bmatrix}\] \[\begin{bmatrix} 0.9773 & 0.9827 \end{bmatrix}\]
- **Psi (Ψ)$^b$** \[\begin{bmatrix} (0.05, 1.10) & 0.7500 \end{bmatrix}\] \[\begin{bmatrix} 0.5900 & 0.0084 \end{bmatrix}\]
- **Psi (Ψ)$^b$** \[\begin{bmatrix} 0.7500 & (0.05, 0.68) \end{bmatrix}\] \[\begin{bmatrix} 0.0084 & 0.6137 \end{bmatrix}\]

**Eta** See Appendix C (Procedure 51)

*Note.* $^a$ = Vector of intercept and slope

$^b$ = Covariance matrix
Table 46

*Parameter Estimates for Growth Mixture Models with Poisson Priors (λ = 5) at Sample size 50, 5 Time Points and 4-class model Using R Program*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>4</td>
<td>3.8486</td>
</tr>
<tr>
<td>Phi (Φ)</td>
<td>0.30</td>
<td>1.0074</td>
</tr>
</tbody>
</table>

z See Appendix C (Procedure 54)

Parameters: C1

<table>
<thead>
<tr>
<th>Pi (π)</th>
<th>0.25</th>
<th>0.10</th>
</tr>
</thead>
</table>

growth factors<sup>a</sup>  
\[
\begin{bmatrix}
-3, 3 & -3, 3 \\
0.05, 1.10 & 0.7500 \\
0.7500 & 0.05, 0.68
\end{bmatrix}
\begin{bmatrix}
1.0154 & 0.9465 \\
0.5950 & 0.0042 \\
0.0042 & 0.5982
\end{bmatrix}
\]

Psi (Ψ)<sup>b</sup>  
\[
\begin{bmatrix}
0.7500 & (0.05, 0.68)
\end{bmatrix}
\begin{bmatrix}
0.6104 & 0.0116 \\
0.0116 & 0.5667
\end{bmatrix}
\]

Eta See Appendix C (Procedure 54)

Parameters: C2

<table>
<thead>
<tr>
<th>Pi (π)</th>
<th>0.25</th>
<th>0.20</th>
</tr>
</thead>
</table>

growth factors<sup>a</sup>  
\[
\begin{bmatrix}
-3, 3 & -3, 3 \\
0.05, 1.10 & 0.7500 \\
0.7500 & 0.05, 0.68
\end{bmatrix}
\begin{bmatrix}
1.0035 & 1.0334 \\
0.6104 & 0.0116 \\
0.0116 & 0.5667
\end{bmatrix}
\]

Eta See Appendix C (Procedure 54)

*Note.*  
a = Vector of intercept and slope  
b = Covariance matrix
Table 47

*Parameter Estimates for Growth Mixture Models with Poisson Priors (λ = 5) at Sample size 50, 5 Time Points and 4-class model Using R Program (continued)*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters: C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi$</td>
<td>0.25</td>
<td>0.30</td>
</tr>
<tr>
<td>growth factors$^a$</td>
<td>$\begin{pmatrix} -3,3 &amp; -3,3 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 0.9793 &amp; 1.0084 \end{pmatrix}$</td>
</tr>
<tr>
<td>$\Psi$</td>
<td>$\begin{pmatrix} 0.05,1.10 &amp; 0.7500 \ 0.7500 &amp; 0.05,0.68 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 0.5906 &amp; -0.0108 \ -0.0108 &amp; 0.6061 \end{pmatrix}$</td>
</tr>
<tr>
<td>Eta</td>
<td>See Appendix C (Procedure 54)</td>
<td></td>
</tr>
<tr>
<td>Parameters: C4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi$</td>
<td>0.25</td>
<td>0.40</td>
</tr>
<tr>
<td>growth factors$^a$</td>
<td>$\begin{pmatrix} -3,3 &amp; -3,3 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 1.0251 &amp; 1.0311 \end{pmatrix}$</td>
</tr>
<tr>
<td>$\Psi$</td>
<td>$\begin{pmatrix} 0.05,1.10 &amp; 0.7500 \ 0.7500 &amp; 0.05,0.68 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 0.6197 &amp; -0.0090 \ -0.0090 &amp; 0.5813 \end{pmatrix}$</td>
</tr>
<tr>
<td>Eta</td>
<td>See Appendix C (Procedure 54)</td>
<td></td>
</tr>
</tbody>
</table>

*Note. a = Vector of intercept and slope
b = Covariance matrix*
Table 48

Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors ($N = 3$) at Sample size 15, 5 Time Points and 3-class model Using R Program

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>3</td>
<td>3.4804</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>0.30</td>
<td>0.9937</td>
</tr>
<tr>
<td>$z$</td>
<td>See Appendix C (Procedure 89)</td>
<td></td>
</tr>
</tbody>
</table>

Parameters: C1

$\Pi (\pi)$
- 0.20
- 0.17

growth factors$^a$
- $\begin{bmatrix} (-3, 3) & (-3, 3) \end{bmatrix}$
- $\begin{bmatrix} 1.0214 & 0.9899 \end{bmatrix}$

$\Psi (\Psi)^b$
- $\begin{bmatrix} (0.05, 1.10) & 0.7500 \end{bmatrix}$
- $\begin{bmatrix} 0.5848 & -0.0132 \end{bmatrix}$

Eta
- See Appendix C (Procedure 89)

Parameters: C2

$\Pi (\pi)$
- 0.33
- 0.33

growth factors$^a$
- $\begin{bmatrix} (-3, 3) & (-3, 3) \end{bmatrix}$
- $\begin{bmatrix} 0.9602 & 1.0259 \end{bmatrix}$

$\Psi (\Psi)^b$
- $\begin{bmatrix} (0.05, 1.10) & 0.7500 \end{bmatrix}$
- $\begin{bmatrix} 0.4372 & 0.1426 \end{bmatrix}$

Eta
- See Appendix C (Procedure 89)

Note. $^a$ = Vector of intercept and slope
$^b$ = Covariance matrix
Table 49

*Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors (N = 3) at Sample size 15, 5 Time Points and 3-class model Using R Program (continued)*

<table>
<thead>
<tr>
<th>Parameters: C3</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pi ((\pi))</td>
<td>0.47</td>
<td>0.50</td>
</tr>
</tbody>
</table>
| growth factors\(^a\) | \[
\begin{bmatrix}
-3, 3 \\
-3, 3
\end{bmatrix}
\] | \[
\begin{bmatrix}
1.0218 & 1.0001 \\
1.0218 & 1.0001
\end{bmatrix}
\] |
| Psi (\(\Psi\))\(^b\) | \[
\begin{bmatrix}
0.05, 1.10 \\
0.7500
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.5961 & -0.0135 \\
-0.0135 & 0.6285
\end{bmatrix}
\] |
| Eta | See Appendix C (Procedure 89) |

*Note.*  
\(^a\) = Vector of intercept and slope  
\(^b\) = Covariance matrix
Table 50

Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors ($N = 4$) at Sample size 15, 5 Time Points and 3-class model Using R Program

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>3</td>
<td>3.4074</td>
</tr>
<tr>
<td>Phi ($\Phi$)</td>
<td>0.30</td>
<td>0.9840</td>
</tr>
<tr>
<td>z</td>
<td>See Appendix C (Procedure 98)</td>
<td></td>
</tr>
</tbody>
</table>

Parameters: C1

| Pi ($\pi$) | 0.20 | 0.17 |
| growth factors$^a$ | $\begin{bmatrix} (-3, 3) & (-3, 3) \end{bmatrix}$ | $\begin{bmatrix} 1.0291 & 1.0110 \end{bmatrix}$ |

$\Psi$ ($\Psi$)$^b$ | $\begin{bmatrix} (0.05, 1.10) & 0.7500 \\ 0.7500 & (0.05, 0.68) \end{bmatrix}$ | $\begin{bmatrix} 0.6009 & 0.0033 \\ 0.0033 & 0.6041 \end{bmatrix}$ |

Eta | See Appendix C (Procedure 98) |

Parameters: C2

| Pi ($\pi$) | 0.33 | 0.33 |
| growth factors$^a$ | $\begin{bmatrix} (-3, 3) & (-3, 3) \end{bmatrix}$ | $\begin{bmatrix} 1.0300 & 1.0102 \end{bmatrix}$ |

$\Psi$ ($\Psi$)$^b$ | $\begin{bmatrix} (0.05, 1.10) & 0.7500 \\ 0.7500 & (0.05, 0.68) \end{bmatrix}$ | $\begin{bmatrix} 0.4527 & 0.1544 \\ 0.1605 & 0.4566 \end{bmatrix}$ |

Eta | See Appendix C (Procedure 98) |

Note. a = Vector of intercept and slope  
b = Covariance matrix
### Table 51

**Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors (N = 3) at Sample size 15, 5 Time Points and 3-class model Using R Program (continued)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pi (π)</td>
<td>0.47</td>
<td>0.50</td>
</tr>
</tbody>
</table>
| growth factors<sup>a</sup> | \[
\begin{pmatrix}
-3.3 \\
-3.3
\end{pmatrix}
\] | \[
\begin{pmatrix}
1.0022 \\
1.0603
\end{pmatrix}
\] |
| Psi (Ψ)<sup>b</sup> | \[
\begin{pmatrix}
(0.05, 1.10) & 0.7500 \\
0.7500 & (0.05, 0.68)
\end{pmatrix}
\] | \[
\begin{pmatrix}
0.6119 & 0.0005 \\
0.0005 & 0.5955
\end{pmatrix}
\] |

Eta | See Appendix C (Procedure 98)

*Note.*  
<sup>a</sup> Vector of intercept and slope  
<sup>b</sup> Covariance matrix
Table 52

Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors \((N = 3)\) at Sample size 200, 3 Time Points and 3-class model Using R Program

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k)</td>
<td>3</td>
<td>3.5882</td>
</tr>
<tr>
<td>(\Phi)</td>
<td>0.30</td>
<td>0.9962</td>
</tr>
<tr>
<td>(z)</td>
<td>See Appendix C (Procedure 137)</td>
<td></td>
</tr>
</tbody>
</table>

Parameters: C1

\[
\begin{align*}
\text{Pi (\(\pi\))} & \quad 0.20 & \quad 0.17 \\
growth \text{ factors}^a & \quad \begin{bmatrix} -3,3 & -3,3 \end{bmatrix} & \quad \begin{bmatrix} 0.9827 & 1.0009 \end{bmatrix} \\
\text{Psi (\(\Psi\))}^b & \quad \begin{bmatrix} 0.05,1.10 & 0.7500 \end{bmatrix} & \quad \begin{bmatrix} 0.6051 & 0.0063 \end{bmatrix} \\
\text{Eta} & \quad \text{See Appendix C (Procedure 137)}
\end{align*}
\]

Parameters: C2

\[
\begin{align*}
\text{Pi (\(\pi\))} & \quad 0.33 & \quad 0.33 \\
growth \text{ factors}^a & \quad \begin{bmatrix} -3,3 & -3,3 \end{bmatrix} & \quad \begin{bmatrix} 0.9556 & 1.0289 \end{bmatrix} \\
\text{Psi (\(\Psi\))}^b & \quad \begin{bmatrix} 0.05,1.10 & 0.7500 \end{bmatrix} & \quad \begin{bmatrix} 0.6032 & 0.0031 \end{bmatrix} \\
\text{Eta} & \quad \text{See Appendix C (Procedure 137)}
\end{align*}
\]

Note.  
\(a = \text{Vector of intercept and slope}\)  
\(b = \text{Covariance matrix}\)
Table 53

Parameter Estimates for Growth Mixture Models with Discrete Uniform Priors \((N = 3)\) at Sample size 200, 3 Time Points and 3-class model Using R Program (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters: C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\Pi (\pi))</td>
<td>0.47</td>
<td>0.50</td>
</tr>
</tbody>
</table>
| growth factors\(^a\) | \([-3, 3] \times [-3, 3]\) | \[
\begin{bmatrix}
0.9609 & 1.0004 \\
0.9609 & 1.0004 \\
\end{bmatrix}
\] |
| \(\Psi (\Psi)\)\(^b\) | \([0.05, 1.10] \times 0.7500\) | \[
\begin{bmatrix}
0.5924 & -0.0071 \\
-0.0071 & 0.6153 \\
\end{bmatrix}
\] |
| Eta | See Appendix C (Procedure 137) |

*Note.* \(^a\) Vector of intercept and slope  
\(^b\) Covariance matrix

Chapter Summary

The goal of the current study is to develope R program using Bayesian method to support applied researchers to estimate the number of latent classes when growth mixture models are conducted. Two main concerns in applying Bayesian method through Markov chain Monte Carlo using Metropolis-Hastings algorithm are: 1) the way to choose the appropriate candidate distribution for parameter of interest, and 2) the effect of informative and noninformative priors placed on the latent classes on growth mixture models.

The performance of the Bayesian method of estimation for the number of latent classes in growth mixture models was demonstrated using data sets generated from
growth mixture model specifying. Four main factors were considered: the prior placed on
the number of latent class parameter, the sample size, the time points, and the complexity
of growth mixture model. First, the candidate distribution was selected and specified in
the Metropolis-Hastings procedure due to the theoretical characteristic of such parameters
and the recommendation from previous research. Most of proposed distribution applied in
estimating the unknown parameter in growth mixture model in the current study were the
same as those used in the prior distribution. The same candidate distributions for each
unknown parameter in growth mixture models are used for all simulation conditions.
Second, it is shown that different specifications of prior information placed on \( K \) have
relatively modest effects on the accuracy of estimation but extend of the effect depends on
the magnitude of the hyperparameter of each prior distribution.
CHAPTER V

CONCLUSIONS

The purpose of this dissertation was to develop an R program to estimate the number of latent classes in growth mixture models. Longitudinal data sets of sample sizes 15, 50, and 200 with 3-, 4-, and 5-time point as well as 2-, 3-, and 4-class linear growth mixture models were fit to evaluate the performance of a program to enumerate the number of latent classes in GMM analysis. Specifically, models with and without information concerning the parameter of interest, $K$, were applied when using the Bayesian approach of estimation. The correctly estimated number of latent classes from the R program were applied in Mplus 7.3 to fit growth mixture model in order to obtain parameter estimates so as to ensure the accuracy of the R package estimation for applied researchers conducting GMM analysis. In this chapter I summarize and discuss the performance of my R program in the following sections. First, the selection of the candidate distribution of each parameter in GMM is discussed followed by convergence and parameter estimates. Then, recommendations are set forth for applied researchers. Finally, limitations of the present project and recommendations for future research are presented.

Choosing the Proposed and Prior Distribution

Proposed Distribution

When using the Metropolis-Hastings algorithm to draw simulated samples, the algorithm might involve estimation on the number of latent classes in GMM such as the
distribution for candidacy in the Markov chain Monte Carlo simulation. Therefore, when conducting GMM analysis, the proposed distribution of each unknown parameter should be carefully selected to ensure the accuracy of the estimation based on the Bayesian approach. In the current dissertation I applied the candidate distribution on the number of latent classes following the distribution’s characteristic and recommendation from previous research. This method is a conservative approach to selecting the proposed distribution while there may be other ways to select such a distribution and provide the proper results in the estimation of parameters of interest that need more advanced methodology to be applied.

**Prior Distribution**

The prior distributions were placed on all unknown parameters in GMM as specified in Chapter III. However, specifying known and unknown information about the parameter of interest, \( K \), was actually applied. The informative prior in the study was the Poisson distribution with three hyperparameter levels (i.e., \( \lambda = 3, 4, \) and \( 5 \)), while the noninformative prior was uniform with three hyperparameter levels (i.e., \( N = 3, 4, \) and \( 5 \)). There were some unexpected results obtained in this dissertation regarding the effect of informative and non-informative priors. First, the estimation performance of estimation of the number of latent mixture classes was not expected to show a lack of difference for both informative and noninformative priors in identifying the correct number of latent classes while holding the sample size constant. Lambert, Sutton, Burton, Abrams, and Jones (2005) pointed out that even noninformative priors provides a large influence on parameter estimates as shown in the findings of some conditions that the correct estimation of \( K \) was also found when using noninformative priors. Second, the impact of the degree of informativeness in the current project was more noticeable when sample size
was small. Studies by S. Depaoli (2014), Berger and Bernado (1992), and Lambert et al. (2005) support this effect. It may be identified as that unknown parameter, $K$, was sensitive with small samples. That is, the value of the estimated number of latent classes increased when the value of the hyperparameter on the prior distribution increased in the GMM analysis with small sample size.

Applications of the Bayesian method of estimation for the number of latent classes in GMM ought to consider prior distributions of the parameter $K$ for two main reasons. One is that the estimation on the number of latent classes was sensitive to the hyperparameter specified in informative priors when research was conducted with small amounts of data (Gifford & Swaminathan, 1990). Another reason is due to the effect of the priors’ input of other parameters in growth mixture model. The prior information toward these parameters may play an important role in the estimation of $K$.

**Convergence and Parameter Estimates**

**Convergence**

Convergence diagnostics for two scalar parameters (i.e., $K$ and $\phi$) were performed to express the representative subset of the parameter space of parameters. There are various ways to test for convergence of iterative simulations or the joint posterior density such as the estimated posterior variance of the parameter ($\hat{R}$) and trace plot. Monitoring the estimated posterior variance of the parameter, $\hat{R}$, was the choice in this dissertation. Even though there were seven unknown parameters to be estimated, only two were diagnosed for the convergence due to the characteristics of parameters which are scalar quantities (Gelman et al., 2014). The criteria for $\hat{R}$ was values closer to 1 represented proper representative (Kass et al., 1998). The convergence did not present any problem overall with GMM data generated in this dissertation. Convergence rates across different
numbers of latent classes of GMM with various sample sizes were almost identical and were close to 1. The $\hat{R}$ closed to 1 means that the sequences of samples have mixed, which showed a good sign of representativeness of the sample in the simulation. The results corresponded to the study by (Lu et al., 2011) in which she ran 20,000 iterations in generating Markov chains through the Gibbs sampling method. Therefore, the adequate number of iterations for running Markov chain Monte Carlo simulations in estimating the number of latent classes in growth mixture models could be 20,000 iterations regardless of sample sizes and time points of collecting the data according the current study. The finding in the current dissertation confirms that the sequences are mixed and suggests no need to run any more simulations.

**Parameter Estimates**

The R program developed for the current dissertation was tested using 27 data sets with (3 sample sizes) × (3 time points) × (3 level of classes on GMM) including 2 priors on $K$ with 3 levels of hyperparameter each resulting in 162 simulations. Each simulation was run two times with 20,000 iterations each to provide two chains of posterior distributions of each parameter. For Bayesian inference, four important values to assess the performance of MCMC applied in this dissertation are the estimated posterior variance of the parameter ($\hat{R}$), empirical standard errors, 95% confidence interval on the posterior inference for a parameter, and estimated parameters.

With the $\hat{R}$ reported earlier, approximate convergence for all conditions was reached (i.e., $\hat{R}$ close to 1). The properties of the simulation seem acceptable. That is, the sequences of the values in the chain were mixed which means that there was no autocorrelation of simulations indicating it was not necessary to run any more simulations. The empirical standard error for all time points, and sample sizes with
informative and noninformative priors in this dissertation seem to have fallen to near zero. Theoretically, this low value of empirical standard error showed that the Bayesian method using the Metropolis-Hastings algorithm can be used to identify the number of latent classes for growth mixture models.

However, when 95% posterior inference for a parameter of interest $K$, was introduced across sample size, time points, and prior information on $K$, the true values for number of latent classes in GMM was out of bound for some situations. In general, for complex growth mixture models (i.e., 4-class model), all true values of $K$ were contained in the 95% confidence interval. Specifically, higher accuracy of estimations was exhibited in larger sample size as expected. In contrast with error in the estimation and $\hat{R}$ diagnosis, there was evidence of incorrectly specifying the number of latent classes in less complex mixture models. Close inspection of the 95% posterior inference confidence intervals revealed that the narrower interval occurred with large sample size regardless of time points of data. The posterior mean of unknown parameter $K$ was also estimated. Theoretically, the sample mean of all Markov chain Monte Carlo samples should be a reasonably good estimates whenever the mean is calculated from large sample sizes. But it was not always true for Bayesian estimation in the current dissertation. The posterior mean showed the accuracy of estimation for some situations only in complex growth mixture models. It is important to note, however, that MCMC under the Bayesian framework allows a huge amount of model flexibility when evaluating high-dimensional integration concerning the unknown parameters (Gelman et al., 2014; O’Neill, 2002). The findings of estimation for the number of latent classes in the less complex growth mixture models in the current investigation were disappointing.
Surprisingly, the R program provided some different details regarding parameter estimates than what the Mplus program provides. For example, Mplus can estimate the growth parameters such as latent slopes and latent intercepts for each latent class including variances and covariances. While the developed R program provides class membership specification ($z_K$), matrix of unobserved variables ($\eta_K$), matrix of latent means ($\alpha_K$), matrix of covariance ($\Psi_K$), and vector of class proportion ($\pi_K$). The extra finding shown in the current dissertation demonstrates the advantage of the Bayesian method over the frequentist method in that it treats unknown parameters as random variables. The posterior distribution plays an important role in the distribution of each and researchers can obtain whatever inferential statistics they are looking for by employing them.

**Implications for Practice**

The Bayesian method of estimation provided by the R program developed for the current dissertation provides guidance to applied researchers doing growth mixture modeling regarding extracting the number of latent classes. It is recommended from the findings in this dissertation that the Bayesian method using the Metropolis-Hastings Markov chain Monte Carlo implemented in the R program developed for the current dissertation, is not appropriate to be used with fewer latent classes (i.e., $K = 2$) growth mixture models until further studies clarify its performance.

Procedures to appropriately estimate the number of latent classes on GMM are being developed regularly (Nylund et al., 2007; Tofighi & Enders, 2008). The findings from Tofighi and Enders (2008) concluded that the accuracy of the estimation improved when sample size increased. The sample sizes used in their study were 400, 700, and 1,000. The method used in Tofighi and Enders’ study is based on comparing several fit
indices that can be used as a criterion to estimate the number of latent classes in a GMM analysis. Applied researchers cannot simply assume that applying a large sample size to GMM analysis will automatically remedy the weakness of class enumeration. Specifically using a Bayesian approach, it is recommended from the current dissertation that large sample size does not prevent applied researcher from extracting too many classes in 2-class growth mixture models (i.e., less complex GMM). Instead, applied researchers must be aware of the interrelation between the structures of the models, nature of the outcomes (categorical and continuous), and covariates (Huang et al., 2010; L. Li & Hser, 2011; Tofighi & Enders, 2008), which is beyond the scope of the current project.

Based on information from previous research and the Bayesian method of estimation studied in the current dissertation I need to hold back on recommending its use to applied researchers based on the results found, which generally identified the incorrect number of latent classes. Three main concerns in applying Bayesian estimation using the Metropolis-Hastings algorithm are candidate distribution, prior distribution on unknown parameters, and covariates in GMM. The first two concerns were of interest in the current dissertation. As can be seen, in this dissertation I applied the same type of information in both candidate and prior distribution toward an unknown number of latent classes parameter, $K$, which was based on the Poisson distribution with no zero specified in sampling through MCMC algorithm. Convergence was not an issue, and empirical standard errors showed low values for all situations. Therefore, 20,000 iterations of simulation are sufficient to provide the proper posterior distribution of each unknown parameters. Focusing on the number of latent classes, however, the results of the current investigation indicated that the sample of the number of classes varied in each iteration of the Metropolis-Hastings simulation method. The posterior mean of the unknown
parameter, $K$, calculated from this posterior distribution for $K$ was not accurate for some conditions. This means that either the prior and candidate distributions on $K$ should be changed or the Metropolis-Hastings algorithm might be an inappropriate method for sampling from the posterior distribution. Researchers who want to uphold the use of the Metropolis-Hastings method to assess the number of latent classes on GMM will have to change the type of prior and/or candidate distribution on the number of latent classes or maintain the same type of those but alter the hyperparameter for each. As for the researcher who wants to challenge himself/herself, the advanced technique such as reversible jump Markov chain Monte Carlo (RJMCMC) could be performed (Gelman et al., 2014; Green, 1995). Therefore, the reliance on techniques becomes an important issue for the determination of the number of latent classes pending further study.

**Limitations**

Despite its advantages, the Bayesian method of estimation for the number of latent classes on GMM using the Metropolis-Hastings algorithm has several limitations and caveats that should be addressed. First, the Metropolis-Hastings algorithm was applied for sampling from a Bayesian posterior distribution when the dimension was fixed. Since this method is theoretically straightforward, it is recommended applied researchers consider this approach when conducting GMM analysis with an unknown number of latent classes. The most practical use of this method is that the number of latent classes, $K$, in each iteration will be estimated first, then such estimated $K$ can be used for the estimation of other unknown parameters in GMM analysis. When $K$ is not fixed, the parameter dimension will vary due to the $K$. In such cases, the reversible jump Markov chain Monte Carlo (RJMCMC) is needed in order to move between models. Reversible jump Markov chain Monte Carlo extends the scope of Metropolis-Hasings methods which
are applied to draw samples from a posterior distribution on the basis of MCMC. In this
dissertation I proposed the use of RJMCMC to estimate the unknown number of mixture
components in a growth mixture model. A key idea of RJMCMC is the additional random
variables that give the matching of parameter space dimensions across models (Gelman et
al., 2014). Reversible jump Markov chain Monte Carlo qualifies two main conditions using
this algorithm, namely, the condition of reversibility (i.e., the proposal functions must be
invertible) which is always satisfied in most functions and conditions of dimension
matching. However, to develop the methodology for an analysis of mixtures with unknown
numbers of latent classes on the basis of Bayesian methods using RJMCMC needs more
advanced algebra than the Metropolis-Hastings algorithm to accomplish the objective.

The second limitation of this dissertation is that covariates for latent variables
were not included in the simulation. B. O. Muthén (2003) suggested that covariates
should be included in the model to correctly specify the model, enumerate the proper
number of classes, and correctly identify class proportions and class memberships. As
mentioned by Tofighi and Enders (2008), there are some benefits of including covariates in
GMM class enumeration when sample size is at least 2,000. It should be noted from
Tofighi and Enders’ study that the power of the enumeration decreased when the model
complexity increased. However, the finding from Tofighi and Enders’ study may not be
applicable to the growth mixture models commonly used in practice because of the higher
model complexity normally found in applied research. Huang et al. (2010) and L. Li and
Hser (2011) also conducted a study regarding the impact of covariates in GMM class
enumeration. Haung and his colleagues pointed out that whether or not to include
covariates in GMM should be driven by the research question specified for the study.
Another issue raised by Hung et al. is that the relationship between covariates and
trajectories and the distribution of the covariates included in the study sample should also be considered. Li and Hser evaluated the covariate effect on GMM class enumeration and concluded that the inclusion of covariates could also lead to more misspecification of the membership prediction on the fitting model than the model without covariates included. This phenomenon brings the inclusion of extra classes as sample size or the misspecification in membership prediction increase.

A third limitation of this dissertation is that some other possible influential factors potentially affecting GMM class enumeration were not addressed. Those factors include the difference in the mixture of latent classes and the number of variables specified in each latent class. As can be seen in the simulation design, the proportions for the 2-class model were very different from the proportions used in the 3- and 4-class models. These proportions might have affected the results in that the procedure developed works well for a four class model and not for a two class model. The procedure in this study indicates the number of estimated latent classes, but it does not provide information to help determine if class assignment is correct. In addition, the data sets generated in the current dissertation were based on only linear trajector classes. However, different types of latent classes could be obtained in growth mixture models. For example, one class could follow linear growth functions, while the other follows quadratic growth functions.

**Recommendation for Future Research**

There is no standard approach for determining the optimal number of latent classes in growth mixture modeling. Therefore, further simulation studies and/or real longitudinal data investigation would be helpful for evaluating the appropriateness of constraints imposed on the estimation and would provide a more comprehensive picture of development of a statistical software program to estimate the number of latent classes in
GMM using a Bayesian approach. The impact of different informative priors on the number of latent classes and also on the other growth parameters is in need of further exploration. Another issue in the use of the Metropolis-Hastings algorithm is the candidate distribution. Proposed or candidate distributions for unknown parameters should be selected carefully. The improper proposed distribution, especially on the number of latent classes, $K$, could lead to the misspecification of the unknown number of latent classes and also affect all of the parameter estimates in growth mixture models. Regarding the recommendations on what to do with the proposed distribution on unknown parameters, for example, when conducting GMM analysis based on the Metropolis-Hastings algorithm sampling method, researchers should be aware of the characteristics of such parameters and the distribution that matches them following previous studies.

Moreover, the influence of covariates on the estimation of number of latent classes in GMM may prove beneficial under much smaller sample size than those used in Tofighi and Enders’ study. Further research should be conducted using the Bayesian method to clarify the role that covariates play in correctly identifying GMM latent classes. Since class estimation with and without covariates could produce different conclusions, it is recommended that the results of including and not including covariates for latent factors be compared. If researchers would like to follow the Metropolis-Hastings algorithm sampling technique as presented in the current dissertation, it is recommended they conduct growth mixture analysis including covariates with small and large sample sizes to compare the performance in each case.

To keep the Bayesian method to estimate the unknown number of latent classes in GMM, further research is required to develop methods for simulating posterior
distributions. One recommended method is the reversible jump Markov chain Monte Carlo (Richardson & Green, 1997). Even though this method needs more advanced methodology to conduct, the performance of the estimation could be prove worthwhile. Applied researchers can possibly use the result from future research to help them specify the number of latent classes when conducting the growth mixture model analysis.

Conclusions

This dissertation demonstrated the extent to which an alternative method under the Bayesian framework was able to estimate the number of latent classes in growth mixture models. Although previous research has manipulated a variety of fit indexes and tests for determining the correct number of latent classes, in this dissertation I developed the R code to estimate the number of latent classes in growth mixture models with different prior information. R code was developed to both generate the growth mixture data sets and analyze the data drawn using the Metropolis-Hastings algorithm as a sampling method. The most striking findings were that at least in some situations the number of latent classes was correctly identified. The current dissertation underscores the importance of considering a Bayesian method that has been explored to estimate the number of latent classes in growth mixture models. This method of estimation, however, did not perform very accurately in most conditions, it can be considered as the pioneer of using the theory associated with a Bayesian approach on growth mixture models in estimating the number of latent classes. Until the literature within the applied research identifies appropriate methods under a Bayesian framework for the estimation on the number of latent classes, we will not have a sense for how accurate estimation of the number of latent classes on growth mixture models can be for applied researchers. Results from this dissertation may require further validation using growth mixture models with
covariates or using more advanced methods such as reversible jump Markov chain Monte Carlo.
REFERENCES


APPENDIX A

R CODE
# Programming for Data Generate Function

# Generate 10,000 observations of mixture of population follow multivariate normal distribution.
# Using parameters from Tofighi and Enders (2008) for class proportions (k)
# Using parameters from Liu & Hancock (2014) for linear factor loadings for latent class 1 across five occasions
# Using parameters from Tofighi & Enders (2008) for quadratic factor loadings for latent class 2 across five occasions
# Using parameters from Hip & Bauer (2006) for non-linear factor loadings for latent class 3 across five occasions
# Assuming the residuals of the repeated measures to be uncorrelated,
# \( \text{big}_k \theta \) is a diagonal matrix for the \( k \)-th class

# Need these libraries installed to run the program

library(MASS)  # for ginv function
library(rockchalk)  # for Multivariate Normal distribution
library(Matrix)  # for Matrix exponential
library(gtools)  # for Dirichlet distribution
library(pscl)  # for Inverse Gamma distribution
library(MCMCpack)  # for Inverse Wishart distribution
library(stats)  # for Wishart distribution
library(mvtnorm)  # for Multivariate Normal distribution

# Change sample size (n), time points (T), and number of class (k), where \( k < n \)

n = 100  # Number of subjects
T = 4  # Time points or waves of data
k = 5  # Number of components
beta = 0.8  # Coefficient matrix
a = -3  # Minimum value of alpha
b = 3  # Maximum value of alpha
set.seed(1234)

generateData <- function(n, T, k, beta, a, b){
  if(k<=n){
    # Calculate variance of observed variable
    phi <- 0.3
    Theta <- phi*diag(T)  # Hip & Bauer (2006)

    # Generate multinomially distributed random number vectors and
# compute multinomial probabilities.

```r
nn <- rmultinom(1, size = n-k, prob = rdirichlet(1, c(1:k)))
nn <- nn + 1
```

```r
mix.data<-c()
for(i in 1:k){
  I <- diag(2)
  Beta <- matrix(beta, ncol = 2, nrow = 2) # Bauer & Curran (2003)
  alpha <- runif(2, a, b)# Bauer & Curran (2003)
  Diff <- (I-Beta)
  Inv_diff <- ginv(Diff) # to find the inverse of (I-beta)

  # Generate disturbance (zeta)- from multivariate normal
  # distribution
  repeat{
    temp <- runif(1,-0.75,0.75)
    psi <- matrix(c(runif(1, 0.05, 1.1), .75, .75,
                      runif(1, 0.05, 0.68)), 2, 2)
    if(det(psi)>0)
      break
  }
  zeta <- mvrnorm(n, c(0, 0), psi)

  # Compute latent class variable (eta)
  sum = alpha + zeta
  eta <- Inv_diff %*% t(sum)

  # Specify parameters of latent growth model
  lambda <- matrix(
    c(rep(1,T),
      0:(T-1)),
    nrow=T,
    ncol=2) # linear growth factors across 3 occasions

  # Calculate mean of observed variable
  mean <- lambda %*% eta
  mutemp <- c()
  for(c in 1:T){
    mutemp[c] <- mean(mean[c,])
  }
  mu <- matrix(
    mutemp,
    nrow=T,
    ncol=2)
```
```r
nrow = T,
ncol = 1)

# Variance of observed variable
sigma <- lambda %*% psi %*% t(lambda) + Theta

# Calculate parameters of unobserved variable
mean_eta <- Inv_diff %*% alpha
covar_eta <- Inv_diff %*% psi %*% t(Inv_diff)

# Generate observed variables using Multivariate Normal Distribution
dist_k <- mvrnorm(nn[i], mu, sigma)
mix.data <- rbind(mix.data, dist_k)

# Find the mixture of the distribution
row.names(mix.data) <- c()
return(mix.data)
}
else{print("error: k must be less than or equal to n")}
}

mix.data <- generateData(n, T, k, beta, a, b)

# Store data in Excel file
# Location and file name can be changed
write.csv(mix.data, "C:/Users/Owner/Desktop/S100_T6_C5.csv")

# Store data in Text file
# Location and file name can be changed
write.table(mix.data, "C:/Dissertation/DATA2/Sample_15/S15_T3_C2.txt", sep="\t")
```
# Generate 15, 50, and 200 observations of mixture of population follow multivariate normal distribution.
# Using parameters from Tofighi and Enders (2008) for class proportions (k).
# Using parameters from Liu & Hancock (2014) for linear factor loadings for latent class 1 across five occasions.
# Using parameters from Tofighi & Enders (2008) for quadratic factor loadings for latent class 2 across five occasions.
# Using parameters from Hip & Bauer (2006) for non-linear factor loadings for latent class 3 across five occasions.
# Assuming the residuals of the repeated measures to be uncorrelated, big_theta is a diagonal matrix for the kth class.

# Need these libraries installed to run the program

library(MASS) # for ginv function
library(gtools) # for Dirichlet distribution
library(mvtnorm) # for Multivariate Normal distribution
library(stats) # for Wishart distribution
library(pscl) # for Inverse Gamma distribution
library(MCMCpack) # for Inverse Wishart distribution
library(multinomRob) # for Multinomial random number generation

# Step 1: Import the data


# Step 2: Calculate Likelihood Function

y = data.matrix(mix.data)
n <- nrow(y)
T <- ncol(y)
# k <- 2 # k is the best guess. Test with k equals what it really is.
phi <- 0.3

Likelihood <- function (k, eta, alpha, prop, phi, psi, z){

likelihood <- 0

beta <- list()
Lambda <- list()
for(i in 1:koriginal){
    # the dimension of this matrix is always two by two
    beta[[i]] <- array(.8, c(2, 2))
    # Lambda corresponds to linear trend with different intercept and slope
    Lambda[[i]] <- cbind(c(rep( 1, ncol(y) )),c( 1:(ncol(y)) )*runif(1) )
}

# the dimension following the linear trend which consists of two columns
I <- rep(list(matrix(diag(2), 2, 2)),koriginal)

for (i in 1:n){
    for (j in 1:koriginal){
        if(j>length(eta)){print(eta);print(j)}
        mean_k0 <- Lambda[[j]]%*%eta[[j]]
        mean_k <- apply(mean_k0, 1, mean)
        sigma_k <- Lambda[[j]]%*% psi[[j]] %*% t(Lambda[[j]])
                  + (phi * diag(ncol(y)))
        mu.eta <- ginv(I[[j]]- beta[[j]]) %*% as.vector(alpha[[j]])
        cov.eta <- ginv(I[[j]]- beta[[j]]) %*% psi[[j]] %*% t(ginv(I[[j]]- beta[[j]]))
        L_ob <- dmvnorm(y[i,], mean_k, sigma_k)
        L_unob <- dmvnorm(eta[[j]][,i], mu.eta, cov.eta)
        logLob <- ifelse(L_ob == 0, 0, log(L_ob))
        logLunob <- ifelse(L_unob == 0, 0, log(L_unob))
        likelihood <- likelihood + z[i,j] * (log(prop[j]) + logLob + logLunob)
    }
}

likelihood
delta <- c(rep(1, koriginal))
a <- 0.002
b <- 0.002
phi <- 0.3
proportion <- prop
mn <- rmultinom(1, size = nrow(y), prob = c(rdirichlet(1, c(1:koriginal))))

# Prior 1: Discrete Uniform for number of component (k)
prior.k <- dunif(1, min=1, max=3)

# Prior 4: Dirichlet distribution for class proportion (pi); hyperparameter (delta = 1)
prior.pi <- ddirichlet(proportion, delta)

# Prior 5: Inverse Gamma for the observed variable: phi,
# hyperparameter (a, b) = (0.002, .002)
prior.phi <- dinvgamma(phi, a, b)

# Prior 7: z
prior.z <- dmultinom(nn, prob=prop)

prior <- 0
prior <- prior + log(prior.k) + log(prior.pi) + log(prior.phi) + log(prior.z)
for (j in 1:koriginal){
  for (i in 1:n){
    # Prior 2: eta
    prior.eta <- dmvnorm(eta[j], i)
    prior <- prior + log(prior.eta)
  }

  # Prior 3: Normal distribution for factor means (alpha)
prior.alpha <- dmvnorm(alpha[j])

  # Prior 6: Inverse Wishart for factor covariance (psi)
  # appeared in unobserved equation
df <- ncol(psi[j])
  scale <- diag(0.3, ncol(psi[j]))
  prior.psi <- diwish(psi[j], df, scale) # psi has to be positive finite matrix

  # For k = {1...K} needs to be combined in some way
  prior <- prior + log(prior.alpha) + log(prior.psi)
}

return(prior)
# Step 4: Calculate posterior distribution

posterior <- function(k, eta, alpha, prop, phi, psi, z){
    post <- Likelihood(k, eta, alpha, prop, phi, psi, z) +
    Prior(k, eta, alpha, prop, phi, psi, z)

    return(post)
}

# Step 5: Generate the random number from the candidate distribution

candidate <- function (k, eta, alpha, prop, phi, psi, z){
    # constants for candidate distribution

    lambda = 5
    n <- nrow(mix.data)

    # candidate 1: Truncated Poisson distribution for
    # number of component (k)

    T <- 3.5         # pre-truncation mean of Poisson
    U <- runif(n)    # the uniform sample
    t = -log(1 - U*(1 - exp(-T))) # the "first" event-times
    T1 <- (T - t)    # the set of (T-t)

    rtruncpois <- rpois(1,T1)+1
    candidate.k <- rtruncpois

    # candidate 4: Dirichlet distribution for class proportion (pi)

    candidate.pi <- prop

    # candidate 5: Exponential (it can be gamma) for the observed variable residual: phi

    candidate.phi <- phi

    #Prior 7: z

    candidate.z <- z

    candidate.eta <- eta
    candidate.alpha <- alpha
    candidate.psi <- psi

    return(list( candidate.k, candidate.eta,
                 candidate.alpha, candidate.pi, candidate.phi,
                 candidate.psi, candidate.z))

}  

# Step 6: Generate the density function from the candidate distribution

```r
candidate_den <- function (k, eta, alpha, prop, phi, psi, z){

  # constants for candidate distribution
  lambda = 5
  n <- nrow(mix.data)

  # candidate_d 1: Truncated Poisson distribution for number of component (k)
  dtruncpois_den <- function(x,lambda,log=FALSE) {
    r <- ifelse(x==0,-Inf,dpois(x,lambda,log=TRUE)-
               log(1-dpois(0,lambda,log=FALSE)))
    if (log) r else exp(r)
  }
  candidate_d.k <- dtruncpois_den(k, lambda, log=FALSE)

  # candidate_d 5: Dirichlet distribution for class proportion (pi)
  candidate_d.pi <- ddirichlet(prop, c(rep(1,koriginal)))

  # candidate_d 5: Exponential (it can be gamma)
  # for the observed variable residual: phi
  # randomly select the value of 1. This value
  # can be changed to others.
  candidate_d.phi <- dexp(1)

  # candidate_d 7: Multinormal for z
  sumz <- c()
  for(l in 1:koriginal){
    sumz[l]<-sum(z[,l])
  }
  candidate_d.z <- dmultinom(sumz,prob = prop)

  density <- 0

  density <- density + log(candidate_d.k) + log(candidate_d.pi) +
            log(candidate_d.phi) + log(candidate_d.z)

  for (l in 1:koriginal){

    # candidate_d 2: Multivariate normal distribution
    # for latent variable (eta)
    for (i in 1:n){
      candidate_d.eta <- dmvnorm(eta[[l]][,i])
      density <- density + log(candidate_d.eta)
    }

    # candidate_d 3: Multivariate normal distribution for factor
```

# means (alpha)
candidate_d.alpha <- dmvnorm(alpha[[l]])

# candidate_d 6: Wishart for factor covariance (psi)
# appeared in unobserved equation
df <- ncol(psi[[l]])
  scale <- diag(0.3, ncol(psi[[l]]))
  candidate_d.psi <- dwish(psi[[l]], df, scale)
density <- density + log(candidate_d.alpha) +
           log(candidate_d.psi)
return(density)

candidate_E <- function (k, eta, alpha, prop, phi, psi, z){
  # constants for candidate distribution
  lambda = 3
  n <- nrow(mix.data)

  # candidate 1: Truncated Poisson distribution
  # for number of component (k)
candidate.k <- k
  # candidate 4: Dirichlet distribution for class proportion (pi)
candidate.pi <- rdirichlet(1, c(1:candidate.k))

  # candidate 5: Exponential (it can be gamma) for
  # the observed variable residual: phi
candidate.phi <- rexp(1)

  # candidate 7: z
  x <- rmultinomial(nrow(mix.data), candidate.pi)

  x<-rmultinom(1, size = nrow(mix.data), prob = candidate.pi)
  candidate.z <- matrix(,nrow=nrow(mix.data), ncol=candidate.k)

  temp1 <- 0
  for(p in 1:candidate.k){
    temp2 <- nrow(mix.data) - x[p] - temp1
    candidate.z[,p] <- c(rep(0,temp1), rep(1, x[p]), rep(0,temp2))
    temp1 <- temp1 + x[p]
  }

  candidate.eta <- list()
  candidate.alpha <- list()
  candidate.psi <- list()

  for (l in 1:candidate.k){

    # candidate 2: Multivariate normal distribution for
    # unobserved variable (eta)
candidate.eta[[l]] <- rmvnorm(2, mean= c(rep(1,n)) , sigma=diag(n))

  }
}
# two rows of eta for linear trend (i.e., intercept and slope)

# candidate 3: Multivariate normal distribution
# for factor means (alpha)
# the dimension of alpha is always 2 for linear trend
candidate.alpha[[l]] <- rmvnorm(1, c(rep(1,2)))

# candidate 6: Wishart for factor covariance (psi) appeared in unobserved equation

df <- 2 # two for linear trend
scale <- diag(0.3, 2) # two for linear trend
wishFix <- rWishart(1, df, scale)
candidate.psi[[l]] <- wishFix[,1]
}

return(list(candidate.k, candidate.eta, candidate.alpha, candidate.pi,
candidate.phi, candidate.psi, candidate.z))

########################################################################
#Step 7: Sampling from Posterior using MCMC by Metropolis Hastings
########################################################################

# Specified values in the list of startvalues

k <- 7
koriginal <- k

ki = k
etai = rep(list(matrix(c(rep(1)), 2, nrow(y))), k)
alphai = rep(list(c(0,0)), k)
propi = c(rdirichlet(1, c(1:k)))
phii = 0
psii = rep(list(matrix(c(1, 0, 0, 1), 2, 2)), k)

# psi has to be Positive definite matrix

# x <- rmultinomial(nrow(mix.data), candidate.pi)
xx <- rmultinom(1, size = nrow(y), prob = c(rdirichlet(1, c(1:k))))
z <- matrix(, nrow=nrow(mix.data), ncol=k)
temp1 <- 0
for(p in 1:k){
temp2 <- nrow(mix.data) - xx[p] - temp1
z[,p] <- c(rep(0,temp1), rep(1, xx[p]), rep(0,temp2))
temp1 <- temp1 + xx[p]
}

zi = z

startvalue <- list(ki, etai, alphai, propi,
Metropolis_MCMC <- function(startvalue, iterations){

    k <- startvalue[[1]]
    eta <- startvalue[[2]]
    alpha <- startvalue[[3]]
    prop <- startvalue[[4]]
    phi <- startvalue[[5]]
    psi <- startvalue[[6]]
    z <- startvalue[[7]]

    chain <- list()
    chain[[1]] = startvalue

    for (i in 1:iterations){
        k <- chain[[i]][[1]]
        eta <- chain[[1]][[2]]
        alpha <- chain[[1]][[3]]
        prop <- chain[[1]][[4]]
        phi <- chain[[1]][[5]]
        psi <- chain[[1]][[6]]
        z <- chain[[1]][[7]]

        print(i)

        proposal = candidate(k, eta, alpha, prop, phi, psi, z)

        pk <- proposal[[1]]
        peta <- proposal[[2]]
        palpha <- proposal[[3]]
        pprop <- proposal[[4]]
        pphi <- proposal[[5]]
        ppsi <- proposal[[6]]
        pz <- proposal[[7]]

        probab <- NaN
        while(is.nan(probab)){
            pold <- posterior(k, eta, alpha, prop, phi, psi, z)
            pnew <- posterior(pk, peta, palpha, pprop, pphi, ppsi, pz)
            cold <- candidate_den(k, eta, alpha, prop, phi, psi, z)
            cnew <- candidate_den(pk, peta, palpha, pprop, pphi, ppsi, pz)
            probab <- exp(pnew-pold) * exp(cold-cnew)
        }

        asd <- runif(1)
        if (asd < probab){
            chain[[i+1]] = candidate_E(pk, eta, alpha, prop, phi, psi, z)
        }
    }
}
else{
    chain[[i+1]] = chain[[i]]
}
}
return(chain)
}

chain <- Metropolis_MCMC(startvalue, 20000)

#############################################################################
# Show all estimated parameters
#############################################################################

# Call unknown numbers of parameter 'k'

kchain <- c()
for(i in 1:length(chain)){
    kchain[i] <- unlist(chain[[i]][[1]])
}

# kchain

# mode

kchain2 <- table(kchain)
names(kchain2)[kchain2 == max(kchain2)]

# Call unobserved variable 'eta'

etachain <- list()
for(i in 1:length(chain)){
    etachain[[i]] <- unlist(chain[[i]][[2]])
}

# etachain

# array(etachain[[1]], c(2, 15, k))

# Call unobserved variable 'alpha'

alphachain <- list()
for(i in 1:length(chain)){
    alphachain[[i]] <- unlist(chain[[i]][[3]])
}

alphachain
mean(unlist(alphachain))

# Call proportion 'pi'

propchain <- list()
for(i in 1:length(chain)){
    propchain[[i]] <- unlist(chain[[i]][[4]])
}
# Call constant value 'phi'

phichain <- c()
for(i in 1:length(chain)){
    phichain[i] <- unlist(chain[[i]][[5]])
}

# Call covariance matrix 'psi'

psichain <- list()
for(i in 1:length(chain)){
    psichain[[i]] <- unlist(chain[[i]][[6]])
}

# Call classmembership variable 'z'

zchain <- list()
for(i in 1:length(chain)){
    zchain[[i]] <- unlist(chain[[i]][[7]])
}

# Data analysis

summary(kchain) # summary the number of component 'k'
summary(unlist(etachain)) # summary the unobserved variable 'eta'
Calculation for the Convergence and Parameter Estimates

Assessing convergence is the formal method to monitor the value of drawn from the posterior density. For any given parameter, the estimated posterior variance of the parameter, $\hat{R}$, is used to assess the convergence (Gelman et al., 2014, p 283). Once $\hat{R}$ is near 1 for all scalar parameters of interest, all the sequences parameter together can be treated as a sample from the target distribution. If $\hat{R}$ is not near 1 for all of parameters, continue the runs (perhaps increase the number of iterations).

In the current study, simulation through MCMC using M-H algorithm was ran two chains which is the minimum chain required recommended by Gelman et al., (2014).

Write files to excel

Run the MCMC simulation (i.e., 10000, or 20000, or 50000, etc.). Save the data from simulation. Make sure to open the data file simulated from MCMC. Given the following steps:

Go to 'R console' -> File -> Load Workspace -> select file name

If the simulation from posterior distributions are drawn for two chains, for example, these two data sets need to open seperately.

Then use the command below to safe the data file in Excel spreadsheet included only parameter of interest for assessing convergence.

Open data file from MCMC simulation chain 1)

```
write.csv((data.frame(cbind(kchain,phichain))), file="data1.csv", row.names=F)
```

Open data file from MCMC simulation chain 2)

```
write.csv((data.frame(cbind(kchain,phichain))), file="data2.csv", row.names=F)
```

import files

```
df1 <- read.csv("data1.csv")
df2 <- read.csv("data2.csv")
```

set parameters
In the current study, 20000 iterations were run for two chains for MCMC simulation. The first half of the simulation was set to the burn-in period and was discarded to calculate the convergence. The second half of each chain was then separated into two parts, the number batches or groups are now $m = 4$, giving the sample size of 5000 for each. These four batches of data sets were used to calculate convergence.

\[
\begin{align*}
n &\leftarrow 5000 \quad \text{n is the number of draws from each batch} \\
m &\leftarrow 4
\end{align*}
\]

# remove burn in

\[
\begin{align*}
\text{burn1} &\leftarrow \text{df1}[10002:20001,] \\
\text{burn2} &\leftarrow \text{df2}[10002:20001,]
\end{align*}
\]

# separate chain

\[
\begin{align*}
\text{chain1} &\leftarrow \text{burn1}[1:5000,] \\
\text{chain2} &\leftarrow \text{burn1}[5001:10000,] \\
\text{chain3} &\leftarrow \text{burn2}[1:5000,] \\
\text{chain4} &\leftarrow \text{burn2}[5001:10000,]
\end{align*}
\]

# Create vectors for chain means and variances of K and PHI

\[
\begin{align*}
\text{meanjK} &\leftarrow \text{c}(\text{mean}(\text{chain1}[,,1]),\text{mean}(\text{chain2}[,,1]),\text{mean}(\text{chain3}[,,1]),\text{mean}(\text{chain4}[,,1])) \\
\text{meanjPHI} &\leftarrow \text{c}(\text{mean}(\text{chain1}[,,2]),\text{mean}(\text{chain2}[,,2]),\text{mean}(\text{chain3}[,,2]),\text{mean}(\text{chain4}[,,2])) \\
\text{varjK} &\leftarrow \text{c}(\text{var}(\text{chain1}[,,1]),\text{var}(\text{chain2}[,,1]),\text{var}(\text{chain3}[,,1]),\text{var}(\text{chain4}[,,1])) \\
\text{varjPHI} &\leftarrow \text{c}(\text{var}(\text{chain1}[,,2]),\text{var}(\text{chain2}[,,2]),\text{var}(\text{chain3}[,,2]),\text{var}(\text{chain4}[,,2]))
\end{align*}
\]

# k and phi estimation

# calculate grand mean for K and PHI

\[
\begin{align*}
\text{meanK} &\leftarrow \text{mean}(\text{meanjK}) \\
\text{meanPHI} &\leftarrow \text{mean}(\text{meanjPHI})
\end{align*}
\]

# k and phi convergence

# calculate convergence

\[
\begin{align*}
B &\leftarrow (n / (m-1)) \times \text{sum}((\text{meanjK}-\text{meanK})^2) \\
W &\leftarrow (1/m) \times \text{sum}(\text{varjK})
\end{align*}
\]
\[
\text{varp} \leftarrow \frac{(n-1)}{n} \times W + \frac{1}{n} \times B \\
\text{rhat}_k \leftarrow \sqrt{\text{varp}}/W
\]

\# \phi 
\[
\text{Bphi} \leftarrow \frac{n}{m-1} \times \text{sum}((\text{meanjPHI} - \text{meanPHI})^2) \\
\text{Wphi} \leftarrow \frac{1}{m} \times \text{sum}(\text{varjPHI}) \\
\text{varphi} \leftarrow \frac{(n-1)}{n} \times \text{Wphi} + \frac{1}{n} \times \text{Bphi} \\
\text{rhat}_\phi \leftarrow \sqrt{\text{varphi}}/\text{Wphi}
\]

\#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
\# Find estimated parameter 
\#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#

\# * n in this step of computer programming to estimate 
\# the number of component in growth mixture models is the 
\# number of sample size. 
\# * n can be changed corresponding to the number of observations 
\# in each data set. 
\# * k can be changed due the k from the estimation
\#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#

\#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~# 
\# eta estimation 
\#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#

\# eta 
\# to export a eta chain as list of 3 dimensional arrays 
\# file1 
\# copy this section to workspace 1 (get data from MCMC simulation chain 1)

\#set n 
\n\text{n} \leftarrow 50

\# needed variables 
\n\text{etachainA} \leftarrow \text{list}() \# change this name throughout this code for separate simulations 
\text{etachainB} \leftarrow \text{list}() 
\text{counter} \leftarrow 1

\# loop to go through chain

\text{for}(\text{i} \text{ in } 2: \text{length} \left(\text{chain})\\{
\# if statement to reset counter for chainB 
\text{if}(\text{i}==10002)\{
\text{counter} \leftarrow 1 \\
\}
\# if statement to select where kchain = 3 
\text{if}(\text{kchain}[\text{i}] == 3)\# change for different k values 
\text{if to get chain A} 
\text{if}(\text{i}<=10001)\{
etachainA[[counter]] <- array(etachain[[i]],c(2,n,3))
  counter <- counter + 1
}
# else to get chain B
else{
  etachainB[[counter]] <- array(etachain[[i]],c(2,n,3))
  counter <- counter + 1
}
}
}

# write file
save(etachainA,etachainB,file="eta1.txt")

# file 2
# copy this section to workspace 2 (get data from MCMC simulation chain 2)

etachainC <- list() # change this name throughout this code for separate simulations
etachainD <- list()
n <- 50
counter <- 1
for(i in 2:length(chain)){
  if(i==10002){
    counter <- 1
  }
  if(kchain[i] == 3){
    if(i<=10001){
      etachainC[[counter]] <- array(etachain[[i]],c(2,n,3))
      counter <- counter + 1
    }
    else{
      etachainD[[counter]] <- array(etachain[[i]],c(2,n,3))
      counter <- counter + 1
    }
  }
}
save(etachainC,etachainD,file="eta2.txt")

# import files
load("eta1.txt")
load("eta2.txt")

# set parameters
n <- 50
k <- 3

# needed variables

counter<-0
meanETAA <- c()
meanETAB <- c()
meanETAC <- c()
meanETAD <- c()
# loop order k dim, col, row

for (p in 1:k){
  for(o in 1:n){
    for(i in 1:2){
      counter <- counter+1
      temp <- c() # temp storage variable
      # collects etachainA items for [i,o,p]
      for(c in 1:length(etachainA)){
        temp[c] <- etachainA[[c]][i,o,p]
      }
      # calculates the mean for position [i,o,p] in etachainA
      meanETAA[counter] <- mean(temp)
      temp <- c() # reset temp
      # collects etachainB items for [i,o,p]
      for(c in 1:length(etachainB)){
        temp[c] <- etachainB[[c]][i,o,p]
      }
      # calculates the mean for position [i,o,p] in etachainB
      meanETAB[counter] <- mean(temp)
      temp <- c() # reset temp
      # collects etachainC items for [i,o,p]
      for(c in 1:length(etachainC)){
        temp[c] <- etachainC[[c]][i,o,p]
      }
      # calculates the mean for position [i,o,p] in etachainC
      meanETAC[counter] <- mean(temp)
      temp <- c() # reset temp
      # collects etachainD items for [i,o,p]
      for(c in 1:length(etachainD)){
        temp[c] <- etachainD[[c]][i,o,p]
      }
      # calculates the mean for position [i,o,p] in etachainD
      meanETAD[counter] <- mean(temp)
    }
  }
}

# form three dimensional arrays from vectors

meanETAA <- array(meanETAA, c(2,n,k))
meanETAB <- array(meanETAB, c(2,n,k))
meanETAC <- array(meanETAC, c(2,n,k))
meanETAD <- array(meanETAD, c(2,n,k))

# grand eta mean

meanETA <- c()
counter<-0
for(p in 1:k){
  for(o in 1:n){
    for(i in 1:2){
      counter <- counter+1
    }
  }
}

# weighted mean
meanETA[counter] <- ((meanETAA[i,o,p]*length(etachainA)) +
(meanETAB[i,o,p]*length(etachainB)) +
(meanETAC[i,o,p]*length(etachainC)) +
(meanETAD[i,o,p]*length(etachainD)) ) /
(length(etachainA)+length(etachainB)+
length(etachainC)+length(etachainD))

meanETA <- array(meanETA,c(2,n,k))

#----------------------------------------------------------------------#
# alpha estimation
#----------------------------------------------------------------------#

# Alpha
# to export alpha chain as list of 3 dimensional arrays

# file1
# copy this section to workspace 1 (get data from MCMC simulation chain 1)

# set n
n <- 50

# needed variables
alphachainA <- list()#change this name throughout this code for separate simulations
alphachainB <- list()
counter <- 1

# loop to go through chain
for(i in 2:length(chain)){

# if statement to reset counter for chainB
if(i==10002){
  counter <- 1
}

# if statement to select where kchain = 3
if(kchain[i] == 3){#change for different k values

# if to get chain A
if(i<=10001){
  alphachainA[[counter]] <- array(alphachain[[i]],c(1,2,3))
  counter <- counter + 1
}
# else to get chain B
else{
  alphachainB[[counter]] <- array(alphachain[[i]],c(1,2,3))
  counter <- counter + 1
}
# write file
save(alphachainA,alphachainB,file="alpha1.txt")

# file2
# copy this section to workspace 2 (get data from MCMC simulation chain 2)

# set n
n <- 50

# needed variables
alphachainC <- list()
#change this name throughout this code for separate simulations
alphachainD <- list()
counter <- 1

# loop to go through chain
for(i in 2:length(chain)){
  #if statement to reset counter for chainB
  if(i==10002){
    counter <- 1
  }
  #if statement to select where kchain = 3
  if(kchain[i] == 3){#change for different k values
    #if to get chain A
    if(i<=10001){
      alphachainC[[counter]] <- array(alphachain[[i]],c(1,2,3))
      counter <- counter + 1
    }
    #else to get chain B
    else{
      alphachainD[[counter]] <- array(alphachain[[i]],c(1,2,3))
      counter <- counter + 1
    }
  }
}

# write file
save(alphachainC,alphachainD,file="alpha2.txt")

# import files
load("alpha1.txt")
load("alpha2.txt")

# set parameters
n <- 50
k <- 3

# needed variables
counter<-0
meanALPHAA <- c()
meanALPHAB <- c()
meanALPHAC <- c()
meanALPHAD <- c()

# loop order p=k dim, o=col, i=row
for(p in 1:k){
  for(o in 1:2){
    for(i in 1:1){
      counter <- counter+1
      temp <- c()#temp storage variable
      #collects psichainA items for [i,o,p]
      for(c in 1:length(alphachainA)){
        temp[c] <- alphachainA[[c]][i,o,p]
      }
      #calculates the mean for position [i,o,p] in psichainA
      meanALPHAA[counter] <- mean(temp)
      temp <- c()#reset temp
      #collects etachainB items for [i,o,p]
      for(c in 1:length(alphachainB)){
        temp[c] <- alphachainB[[c]][i,o,p]
      }
      #calculates the mean for position [i,o,p] in psichainB
      meanALPHAB[counter] <- mean(temp)
      temp <- c()#reset temp
      #collects psichainC items for [i,o,p]
      for(c in 1:length(alphachainC)){
        temp[c] <- alphachainC[[c]][i,o,p]
      }
      #calculates the mean for position [i,o,p] in psichainC
      meanALPHAC[counter] <- mean(temp)
      temp <- c()#reset temp
      #collects psichainD items for [i,o,p]
      for(c in 1:length(alphachainD)){
        temp[c] <- alphachainD[[c]][i,o,p]
      }
      #calculates the mean for position [i,o,p] in psichainD
      meanALPHAD[counter] <- mean(temp)
    }
  }
}

# form three dimensional arrays from vectors
meanALPHAA <- array(meanALPHAA,c(1,2,k))
meanALPHAB <- array(meanALPHAB,c(1,2,k))
meanALPHAC <- array(meanALPHAC,c(1,2,k))
meanALPHAD <- array(meanALPHAD,c(1,2,k))

# grand alpha mean

meanALPHA <- c()
counter<-0
for(p in 1:k){
  for(o in 1:2){
    for(i in 1:1){
      counter <- counter+1
      #weighted mean
      meanALPHA[counter] <- ((meanALPHAA[i,o,p]*length(alphachainA)) +
                              (meanALPHAB[i,o,p]*length(alphachainB)) +
                              (meanALPHAC[i,o,p]*length(alphachainC)) +
                              (meanALPHAD[i,o,p]*length(alphachainD))) /
                          (length(alphachainA)+length(alphachainB)+
                           length(alphachainC)+length(alphachainD))
    }
  }
}

meanALPHA <- array(meanALPHA,c(1,2,k))

#----------------------------------------------------------------------#
# proportion estimation
#----------------------------------------------------------------------#

# Prop chain export file 1
# copy this section to workspace 1 (get data from MCMC simulation chain 1)

propchainA <- list()#change this name throughout this code for seperate simulations
propchainB <- list()
n <- 50
counter <- 1
for(i in 2:length(chain)){
  if(i==10002){
    counter <- 1
  }
  if(kchain[i] == 3){
    if(i<=10001){
      propchainA[[counter]] <- propchain[[i]]
      counter <- counter + 1
    }
    else{
      propchainB[[counter]] <- propchain[[i]]
      counter <- counter + 1
    }
  }
}
save(propchainA,propchainB,file="Prop1.txt")

# export prop file 2
# copy this section to workspace 2 (get data from MCMC simulation chain 2)
# change this name throughout this code for separate simulations
propchainC <- list()
propchainD <- list()
n <- 50
counter <- 1
for(i in 2:length(chain)){
  if(i==10002){
    counter <- 1
  }
  if(kchain[i] == 3){
    if(i<=10001){
      propchainC[[counter]] <- propchain[[i]]
      counter <- counter + 1
    }
    else{
      propchainD[[counter]] <- propchain[[i]]
      counter <- counter + 1
    }
  }
}
save(propchainC,propchainD,file="Prop2.txt")

# load Prop file
load("Prop1.txt")
load("Prop2.txt")

n <- 50
k <- 3
counter<-0
meanPROPA <- c()
meanPROPB <- c()
meanPROPC <- c()
meanPROPD <- c()

# loop order p=k dim, o=col, i=row
for(o in 1:k){
  for(i in 1:1){
    counter <- counter+1
    temp <- c()
    for(c in 1:length(propchainA)){
      temp[c] <- propchainA[[c]][i,o]
    }
    meanPROPA[counter] <- mean(temp)
    temp <- c()
    for(c in 1:length(propchainB)){
      temp[c] <- propchainB[[c]][i,o]
    }
  }
}
meanPROPB[counter] <- mean(temp)
temp <- c()
for(c in 1:length(propchainC)){
    temp[c] <- propchainC[[c]][i,o]
}
meanPROPC[counter] <- mean(temp)
temp <- c()
for(c in 1:length(propchainD)){
    temp[c] <- propchainD[[c]][i,o]
}
meanPROPD[counter] <- mean(temp)
}

meanPROPA <- array(meanPROPA,c(1,k))
meanPROPB <- array(meanPROPB,c(1,k))
meanPROPC <- array(meanPROPC,c(1,k))
meanPROPD <- array(meanPROPD,c(1,k))

meanPROP <- c()
counter <- 0
for(o in 1:k){
    for(i in 1:1){
        counter <- counter+1
        meanPROP[counter] <- ((meanPROPA[i,o]*length(propchainA)) +
            (meanPROPB[i,o]*length(propchainB)) +
            (meanPROPC[i,o]*length(propchainC)) +
            (meanPROPD[i,o]*length(propchainD))) /
            (length(propchainA)+length(propchainB)+
            length(propchainC)+length(propchainD))
    }
}
meanPROP <- array(meanPROP,c(1,k))

#----------------------------------------------------------------------#
# psi estimation
#----------------------------------------------------------------------#

# Psi
# to export a psi chain as list of 3 dimensional arrays

# file1
# copy this section to workspace 1 (get data from MCMC simulation chain 1)

# set n
n <- 50

# needed variables
psichainA <- list() # change this name throughout this code for separate simulations
psichainB <- list()
counter <- 1
# loop to go through chain

for(i in 2:length(chain)){

    #if statement to reset counter for chainB
    if(i==10002){
        counter <- 1
    }

    #if statement to select where kchain = 3
    if(kchain[i] == 3){#change for different k values

        #if to get chain A
        if(i<10001){
            psichainA[[counter]] <- array(psichain[[i]],c(2,2,3))
            counter <- counter + 1
        }

        #else to get chain B
        else{
            psichainB[[counter]] <- array(psichain[[i]],c(2,2,3))
            counter <- counter + 1
        }
    }
}

# write file

save(psichainA,psichainB,file="psi1.txt")

# file2

# copy this section to workspace 2 (get data from MCMC simulation chain 2)

psichainC <- list()#change this name throughout this code for seperate simulations
psichainD <- list()
n <- 50
counter <- 1
for(i in 2:length(chain)){
    if(i==10002){
        counter <- 1
    }

    if(kchain[i] == 3){
        if(i<10001){
            psichainC[[counter]] <- array(psichain[[i]],c(2,2,3))
            counter <- counter + 1
        }

        else{
            psichainD[[counter]] <- array(psichain[[i]],c(2,n,3))
            counter <- counter + 1
        }
    }
}

save(psichainC,psichainD,file="psi2.txt")
# import files

load("psi1.txt")
load("psi2.txt")

# set parameters
n <- 50
k <- 3

# needed variables

counter<-0
meanPSIA <- c()
meanPSIB <- c()
meanPSIC <- c()
meanPSID <- c()

# loop order k dim, col, row

for(p in 1:k){
  for(o in 1:2){
    for(i in 1:2){
      counter <- counter+1
      temp <- c()#temp storage variable

      #collects psichainA items for [i,o,p]
      for(c in 1:length(psichainA)){
        temp[c] <- psichainA[[c]][i,o,p]
      }

      #calculates the mean for position [i,o,p] in psichainA
      meanPSIA[counter] <- mean(temp)
      temp <- c()#reset temp

      #collects etachainB items for [i,o,p]
      for(c in 1:length(psichainB)){
        temp[c] <- psichainB[[c]][i,o,p]
      }

      #calculates the mean for position [i,o,p] in psichainB
      meanPSIB[counter] <- mean(temp)
      temp <- c()#reset temp

      #collects psichainC items for [i,o,p]
      for(c in 1:length(psichainC)){
        temp[c] <- psichainC[[c]][i,o,p]
      }

      #calculates the mean for position [i,o,p] in psichainC
      meanPSIC[counter] <- mean(temp)
      temp <- c()#reset temp

      #collects psichainD items for [i,o,p]
      for(c in 1:length(psichainD)){
        temp[c] <- psichainD[[c]][i,o,p]
      }

      #calculates the mean for position [i,o,p] in psichainD
      meanPSID[counter] <- mean(temp)
      temp <- c()#reset temp
    }
  }
}

# needed variables

counter<-0
meanPSIA <- c()
meanPSIB <- c()
meanPSIC <- c()
meanPSID <- c()
temp[c] <- psichainD[[c]][i,o,p]
}

# calculates the mean for position [i,o,p] in psichainD
meanPSID[counter] <- mean(temp)
}

# form three dimensional arrays from vectors
meanPSIA <- array(meanPSIA,c(2,2,k))
meanPSIB <- array(meanPSIB,c(2,2,k))
meanPSIC <- array(meanPSIC,c(2,2,k))
meanPSID <- array(meanPSID,c(2,2,k))

# grand psi mean
meanPSI <- c()
counter<-0
for(p in 1:k){
  for(o in 1:2){
    for(i in 1:2){
      counter <- counter+1
      # weighted mean
      meanPSI[counter] <- ((meanPSIA[i,o,p]*length(psichainA)) +
                           (meanPSIB[i,o,p]*length(psichainB)) +
                           (meanPSIC[i,o,p]*length(psichainC)) +
                           (meanPSID[i,o,p]*length(psichainD))) /
                           (length(psichainA)+length(psichainB)+
                            length(psichainC)+length(psichainD))
    }
  }
}

meanPSI <- array(meanPSI,c(2,2,k))

#-------------------------------#
# z estimation
#-------------------------------#

# z chain export file 1
# copy this section to workspace 1 (get data from MCMC simulation chain 1)
zchainA <- list()#change this name throughout this code for separate simulations
zchainB <- list()
n <- 50
counter <- 1
for(i in 10002:length(chain)){
  if(i==15002){
    counter <- 1
  }
}
if(kchain[i] == 3){
    if(i<=15001){
        zchainA[[counter]] <- zchain[i]
        counter <- counter + 1
    }
    else{
        zchainB[[counter]] <- zchain[i]
        counter <- counter + 1
    }
}

save(zchainA,zchainB,file="z1.txt")

# export z file2
# copy this section to workspace 2 (get data from MCMC simulation chain 2)

zchainC <- list()  # change this name throughout this code for separate simulations
zchainD <- list()

n <- 50
counter <- 1
for(i in 10002:length(chain)){
    if(i==15002){
        counter <- 1
    }
    if(kchain[i] == 3){
        if(i<=15001){
            zchainC[[counter]] <- zchain[i]
            counter <- counter + 1
        }
        else{
            zchainD[[counter]] <- zchain[i]
            counter <- counter + 1
        }
    }
}

save(zchainC,zchainD,file="z2.txt")

# load z file
load("z1.txt")
load("z2.txt")

n <- 50
k <- 3
counter<-0
meanZA <- c()
meanZB <- c()
meanZC <- c()
meanZD <- c()}
# loop order p=k dim, o=col, i=row

for(p in 1:k){
  for(o in 1:n){
    counter <- counter+1
    temp <- c()
    for(c in 1:length(zchainA)){
      temp[c] <- zchainA[[c]][o,p]
    }
    meanZA[counter] <- mean(temp)
    temp <- c()
    for(c in 1:length(zchainB)){
      temp[c] <- zchainB[[c]][o,p]
    }
    meanZB[counter] <- mean(temp)
    temp <- c()
    for(c in 1:length(zchainC)){
      temp[c] <- zchainC[[c]][o,p]
    }
    meanZC[counter] <- mean(temp)
    temp <- c()
    for(c in 1:length(zchainD)){
      temp[c] <- zchainD[[c]][o,p]
    }
    meanZD[counter] <- mean(temp)
  }
}

meanZA <- array(meanZA,c(n,k))
meanZB <- array(meanZB,c(n,k))
meanZC <- array(meanZC,c(n,k))
meanZD <- array(meanZD,c(n,k))

meanZ <- c()
counter<-0
for(p in 1:k){
  for(o in 1:n){
    counter <- counter+1
    meanZ[counter] <- ((meanZA[o,p]*length(zchainA)) +
                        (meanZB[o,p]*length(zchainB)) +
                        (meanZC[o,p]*length(zchainC)) +
                        (meanZD[o,p]*length(zchainD))) /
                        (length(zchainA)+length(zchainB)+
                        length(zchainC)+length(zchainD))
  }
}

meanZ <- array(meanZ,c(n,k))

# Find the maximum of z in each row to speicfy the class membership

meanZfix <- meanZ
for(p in 1:k){
  for(o in 1:n){
if(meanZ[o,p]==max(meanZ[o,]))
    meanZfix[o,p] <- 1
else
    meanZfix[o,p] <- 0
}

meanZfix <- array(meanZfix,c(n,k))

#----------------------------------------------------------------------#
# Data Analysis
#----------------------------------------------------------------------#

#----------------------------------------------------------------------#
# k and phi standard error and confidence interval
#----------------------------------------------------------------------#

# Empirical standard error

error_k = sqrt((sum((meanjK-meanK)^2))/3)

error_phi = sqrt((sum((meanjPHI-meanPHI)^2))/3)

# 95% confidence interval

se_k = qnorm(0.975)*sqrt(varp)/sqrt(n)
left = meanK-se_k
right = meanK+se_k
CI_k = c(left, right)

se_phi = qnorm(0.975)*sqrt(varphi)/sqrt(n)
left = meanK-se_phi
right = meanK+se_phi
CI_phi = c(left, right)
APPENDIX B

MPLUS SYNTAX
Title: G**rowth Mixture modeling: 3 latent classes with 3 time points and 50 observations**

Data: File "C:\Users\Owner\Desktop\Mplus(2)\DATA3\Sample_50\S50_T3_C3.dat" ;

Variable:
Names are id y1 y2 y3;
Usevariables are y1 y2 y3;
Classes = c(3);

Analysis:
Type = mixture;
miteration = 20000;

Model:
%overall%
int by y1-y3@1;
slp by y1@0 y2@1 y3@2;
[y1-y3@0];
y1 y2 y3;
int*.109 slp*;
int with slp*;
[c#1*0];

! latent class designation %c#1%
%c#1%
[int*4.096 slp*.843];
y1 y2 y3;
int*1.986 slp*.160;
int with slp*-.531;

! latent class designation %c#2%
%c#2%
[int*2.758 slp*.516];
y1 y2 y3;
int*1.418 slp*.240;
int with slp*-.593;
%c#3%
[int*1.233 slp*];
int*1.018 slp*.140;
int with slp*-.193;

Output:
techn1 techn7;

---

Title: G**rowth Mixture modeling: 4 latent classes with 4 time points and 200 observations**

Data: File "C:\Users\Owner\Desktop\Mplus(2)\DATA3\Sample_200\S200_T4_C4.dat" ;

Variable:
Names are id y1 y2 y3 y4;
Usevariables are y1 y2 y3 y4;
Classes = c(4);

Analysis:
Type = mixture;
miteration = 20000;

Model:
%overall%
  int by y1-y4@1;
  slp by y1@0 y2@1 y3@2 y4@3;
  [y1-y4@0];
  y1 y2 y3 y4;
  int*.109 slp*;
  int with slp*;
  [c#1*0];

! latent class designation %c#1%
  %c#1%
  [int*4.096 slp*.843];
  y1 y2 y3 y4;
  int*1.986 slp*.160;
  int with slp*-.531;

! latent class designation %c#2%
  %c#2%
  [int*2.758 slp*.516];
  y1 y2 y3 y4;
  int*1.418 slp*.240;
  int with slp*-.593;

%c#3%
  [int*1.233 slp*];
  int*1.018 slp*.140;
  int with slp*-.193;
%c#4%
  [int*1.7 slp*];
  int*1.8 slp*.10;
  int with slp*-.13;

Output:
  tech1 tech7;
APPENDIX C

OUTPUTS
Procedure 18: P4_S15_T5_C4

> rhat_k
[1] 0.9999019

> rhat_phi
[1] 1.000598

> meanK
[1] 3.916

> meanPHI
[1] 0.9998188

> meanETA
,, 1

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.034380</td>
<td>0.9883329</td>
<td>1.0249292</td>
<td>1.005917</td>
<td>0.9889663</td>
<td>1.026553</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.056125</td>
<td>1.0083979</td>
<td>0.9840337</td>
<td>0.990470</td>
<td>0.9708889</td>
<td>1.012266</td>
</tr>
<tr>
<td>[],8</td>
<td>[],9</td>
<td>[],10</td>
<td>[],11</td>
<td>[],12</td>
<td>[],13</td>
<td>[],14</td>
</tr>
<tr>
<td>[1,]</td>
<td>0.9883675</td>
<td>1.0318628</td>
<td>1.011854</td>
<td>0.9958426</td>
<td>0.9717406</td>
<td>0.9962496</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9818900</td>
<td>0.9936045</td>
<td>1.011532</td>
<td>1.0182152</td>
<td>0.9833018</td>
<td>1.0141270</td>
</tr>
<tr>
<td>[],15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1,]</td>
<td>0.9664126</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9909582</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

,, 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.0255021</td>
<td>1.005622</td>
<td>0.950920</td>
<td>0.9773973</td>
<td>0.9903796</td>
<td>1.0190841</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9827574</td>
<td>1.001322</td>
<td>1.029712</td>
<td>1.0283494</td>
<td>0.9776913</td>
<td>0.9807996</td>
</tr>
<tr>
<td>[],8</td>
<td>[],9</td>
<td>[],10</td>
<td>[],11</td>
<td>[],12</td>
<td>[],13</td>
<td>[],14</td>
</tr>
<tr>
<td>[1,]</td>
<td>0.9910646</td>
<td>1.0344866</td>
<td>0.9992211</td>
<td>0.9942835</td>
<td>1.012501</td>
<td>0.9786964</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.0266088</td>
<td>0.9754011</td>
<td>0.9600123</td>
<td>1.0122714</td>
<td>1.032097</td>
<td>1.0076381</td>
</tr>
<tr>
<td>[],15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1,]</td>
<td>1.0256146</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9639947</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

,, 3

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.017580</td>
<td>0.9912899</td>
<td>1.0149725</td>
<td>0.9933408</td>
<td>1.0109554</td>
<td>1.025652</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.009252</td>
<td>0.9676235</td>
<td>0.9991543</td>
<td>0.9946464</td>
<td>0.9999128</td>
<td>1.032550</td>
</tr>
<tr>
<td>[],8</td>
<td>[],9</td>
<td>[],10</td>
<td>[],11</td>
<td>[],12</td>
<td>[],13</td>
<td>[],14</td>
</tr>
<tr>
<td>[1,]</td>
<td>1.013965</td>
<td>0.9949019</td>
<td>0.9883431</td>
<td>0.9663639</td>
<td>0.9728912</td>
<td>0.9756837</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.035196</td>
<td>1.0179536</td>
<td>0.9404442</td>
<td>0.9744640</td>
<td>0.9857204</td>
<td>1.0107969</td>
</tr>
</tbody>
</table>
[,15]
[1,] 1.0128058
[2,] 0.9823979

, , 4

[1,] 0.9786179 1.014212 1.019851 1.020287 1.022600 0.9953675 1.007881
[2,] 0.9856237 1.000883 0.9611572 0.9920528 0.9927001 1.022102

[1,] 1.0029329 0.9959045 0.9966545 0.995426 0.9959074 1.008910 0.9502569
[2,] 0.9919957 0.9798034 0.9775794 1.022755 0.9928915 1.013354 0.9962608

[,15]
[1,] 0.9744895
[2,] 1.0074156

> meanALPHA

, , 1

[,1] [,2]
[1,] 0.9784437 1.027454

, , 2

[,1] [,2]
[1,] 1.014086 1.008378

, , 3

[,1] [,2]
[1,] 1.005031 0.9563068

, , 4

[,1] [,2]
[1,] 1.002159 1.010265

> meanPROP

[1,] 0.1007232 0.2005551 0.2964706 0.4022511

> meanPSI

, , 1
\[
\begin{bmatrix}
[,1] & [,2] \\
[1,] & 0.604464327 & -0.007062152 \\
[2,] & -0.007062152 & 0.621664847 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
[,1] & [,2] \\
[1,] & 0.4496711 & 0.1527147 \\
[2,] & 0.1537636 & 0.4420207 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
[,1] & [,2] \\
[1,] & 0.605136852 & 0.001567654 \\
[2,] & 0.001567654 & 0.616633851 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
[,1] & [,2] \\
[1,] & 0.4416162 & 0.1421302 \\
[2,] & 0.1560493 & 0.4367691 \\
\end{bmatrix}
\]

\[
> \text{meanZfix}
\]

\[
\begin{bmatrix}
[1,] & 1 & 0 & 0 & 0 \\
[2,] & 0 & 1 & 0 & 0 \\
[3,] & 0 & 1 & 0 & 0 \\
[4,] & 0 & 1 & 0 & 0 \\
[5,] & 0 & 0 & 1 & 0 \\
[6,] & 0 & 0 & 1 & 0 \\
[7,] & 0 & 0 & 1 & 0 \\
[8,] & 0 & 0 & 1 & 0 \\
[9,] & 0 & 0 & 1 & 0 \\
[10,] & 0 & 0 & 0 & 1 \\
[11,] & 0 & 0 & 0 & 1 \\
[12,] & 0 & 0 & 0 & 1 \\
[13,] & 0 & 0 & 0 & 1 \\
[14,] & 0 & 0 & 0 & 1 \\
[15,] & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

\[
> \text{error}_k \\
\begin{bmatrix}
[1] & 0.003695042 \\
\end{bmatrix}
\]

\[
> \text{error}_\phi \\
\begin{bmatrix}
[1] & 0.03695079 \\
\end{bmatrix}
\]
> CI_k
[1] 2.954226 4.877774

> CI_phi
[1] 3.415252 4.416748

%==========================================================================
%==========================================================================

Procedure 21: P5_S15_T3_C4

> rhat_k
[1] 1.000218

> rhat_phi
[1] 0.9999196

> meanK
[1] 4.182

> meanPHI
[1] 0.9953382

> meanETA
, , 1

[1,] 1.0227401 0.9428978 0.9650888 1.0187955 1.0525254 1.0111180 0.9886998
[2,] 0.9814714 1.0284302 0.9985116 1.0162800 0.9994083 0.9840050 0.9980331
[1,] 1.0126819 0.9932253 0.9906128 1.0162433 1.0095911 0.9848499
[2,] 0.9377471 1.0230010 0.9863404 1.0021961 0.9384679 0.9992223 0.9988755
          [,15]
[1,] 0.9878429
[2,] 1.0092075

, , 2

[1,] 0.9670483 0.9852444 0.9784298 0.9977314 1.0014900 0.9758823 0.9917337
[2,] 0.9955094 0.9799316 1.0162348 0.9914050 1.0323470 0.9944905 0.9758485
[1,] 1.0177522 1.0277325 0.9892482 0.9915558 1.0460929 0.9984251 0.9910689
[2,] 1.0217380 0.9964756 0.9894804 0.9737278 0.9794428 1.0027068 1.0384737
          [,15]
```
> meanALPHA

1

[,1] [,2]
[1,] 1.021048 1.041581

2

[,1] [,2]
[1,] 1.003454 1.030519

3

[,1] [,2]
[1,] 0.9599314 0.9400886

4

[,1] [,2]
[1,] 1.043745 1.009454
```
> meanPROP

[1,] 0.1018177 0.1994395 0.3037814 0.3949615

> meanPSI

[, 1]

   [,1]       [,2]
[1,] 0.61135761 0.01690694
[2,] 0.01690694 0.57038700

[, 2]

   [,1]       [,2]
[1,] 0.4595232 0.1568850
[2,] 0.1591862 0.4443558

[, 3]

   [,1]       [,2]
[1,] 0.5754772598 -0.0008705681
[2,] -0.0008705681 0.6069494228

[, 4]

   [,1]       [,2]
[1,] 0.4596938 0.1506575
[2,] 0.1511132 0.4430959

> meanZfix

[1,] 1 0 0 0
[2,] 0 1 0 0
[3,] 0 1 0 0
[4,] 0 1 0 0
[5,] 0 0 1 0
[6,] 0 0 1 0
[7,] 0 0 1 0
[8,] 0 0 1 0
[9,] 0 0 1 0
[10,] 0 0 0 1
[11,] 0 0 0 1
[12,] 0 0 0 1
[13,] 0 0 0 1
```r
[14,] 0 0 0 1
[15,] 0 0 0 1
>
> error_k
[1] 0.04919024
>
> error_phi
[1] 0.006155521
>
> CI_k
[1] 3.194769 5.169231
>
> CI_phi
[1] 3.684007 4.679993
>
>'%========================================================================
>'%========================================================================
'
Procedure 51: P5_S50_T4_C4

rhat_k
[1] 1.000119
>
> rhat_phi
[1] 1.000692
>
> meanK
[1] 3.8528
>
> meanPHI
[1] 0.9948971
>
> meanETA
, , 1

[1,] 0.9944321 1.0089823 1.0027000 1.013403 1.024262 0.9971473 0.973444 0.9982912 1.000340 0.9579288 0.9819845 1.032963 0.9839288 0.9720505 1.0418625 0.9968262 0.9839288 0.9818242 1.036572 0.9818242 1.036572
[2,] 1.00275298 0.9877944 0.9875857 1.034000 1.009377 0.9817907 1.002845 0.9502864 1.022827 0.9870680 0.9809418 1.011841 1.042737 1.001624
[3,] 0.9982912 1.000340 0.9579288 0.9819845 1.032963 0.9839288 0.9819845 0.9809418 1.011841 1.042737 1.001624
```

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.006008</td>
<td>1.0004220</td>
<td>0.9974485</td>
<td>0.9828407</td>
<td>0.9974083</td>
<td>0.9858027</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.043830</td>
<td>0.9576322</td>
<td>1.0260097</td>
<td>0.9644966</td>
<td>0.9878752</td>
<td>1.0132748</td>
</tr>
<tr>
<td>[1,]</td>
<td>1.0369744</td>
<td>1.022688</td>
<td>0.9981738</td>
<td>0.9841505</td>
<td>1.008438</td>
<td>0.9720758</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9794629</td>
<td>0.990936</td>
<td>1.0090823</td>
<td>1.0361173</td>
<td>1.029821</td>
<td>1.0017715</td>
</tr>
<tr>
<td>[1,]</td>
<td>0.9875684</td>
<td>0.9691106</td>
<td>0.9678538</td>
<td>0.9991720</td>
<td>1.061501</td>
<td>1.004015</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.0230347</td>
<td>1.0280673</td>
<td>0.9604913</td>
<td>0.9831158</td>
<td>1.006009</td>
<td>0.9991688</td>
</tr>
<tr>
<td>[1,]</td>
<td>0.9930733</td>
<td>1.020389</td>
<td>0.9985793</td>
<td>1.016247</td>
<td>0.9869224</td>
<td>1.0190550</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9594466</td>
<td>0.968020</td>
<td>1.0129838</td>
<td>1.004898</td>
<td>0.9795545</td>
<td>0.9872628</td>
</tr>
<tr>
<td>[1,]</td>
<td>1.0364476</td>
<td>0.9887084</td>
<td>1.019464</td>
<td>0.9842285</td>
<td>1.0157588</td>
<td>0.9783332</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9877732</td>
<td>0.9964391</td>
<td>1.023179</td>
<td>1.0126132</td>
<td>0.9433976</td>
<td>0.9771589</td>
</tr>
<tr>
<td>[1,]</td>
<td>0.9729374</td>
<td>1.022614</td>
<td>0.9966887</td>
<td>0.9904555</td>
<td>0.9506009</td>
<td>1.006376</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.0366309</td>
<td>1.009714</td>
<td>1.0214429</td>
<td>0.9895837</td>
<td>1.0102626</td>
<td>0.985141</td>
</tr>
<tr>
<td>[1,]</td>
<td>0.9940128</td>
<td>1.005544</td>
<td>1.040875</td>
<td>1.048235</td>
<td>0.9977796</td>
<td>1.0135197</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9536781</td>
<td>1.023562</td>
<td>1.001507</td>
<td>1.037862</td>
<td>1.0096490</td>
<td>0.9782512</td>
</tr>
<tr>
<td>[,50]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1,]</td>
<td>1.011961</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[2,]</td>
<td>1.004038</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>0.9822935</td>
<td>1.010542</td>
<td>1.0095392</td>
<td>1.0100647</td>
<td>1.030829</td>
<td>1.0268291</td>
</tr>
<tr>
<td></td>
<td>1.0448886</td>
<td>1.018570</td>
<td>0.9905296</td>
<td>1.034956</td>
<td>0.9707337</td>
<td>0.9951823</td>
</tr>
<tr>
<td>----</td>
<td>-----------</td>
<td>----------</td>
<td>------------</td>
<td>----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>2,</td>
<td>[ ,8 ]</td>
<td>[ ,9 ]</td>
<td>[ ,10 ]</td>
<td>[ ,11 ]</td>
<td>[ ,12 ]</td>
<td>[ ,13 ]</td>
</tr>
<tr>
<td></td>
<td>1.0377509</td>
<td>0.9687201</td>
<td>0.9885569</td>
<td>1.026386</td>
<td>0.9991311</td>
<td>1.0045581</td>
</tr>
<tr>
<td></td>
<td>[ ,15 ]</td>
<td>[ ,16 ]</td>
<td>[ ,17 ]</td>
<td>[ ,18 ]</td>
<td>[ ,19 ]</td>
<td>[ ,20 ]</td>
</tr>
<tr>
<td></td>
<td>1.0209358</td>
<td>0.9527359</td>
<td>0.9724078</td>
<td>1.015092</td>
<td>1.0157872</td>
<td>0.9986870</td>
</tr>
<tr>
<td></td>
<td>[ ,22 ]</td>
<td>[ ,23 ]</td>
<td>[ ,24 ]</td>
<td>[ ,25 ]</td>
<td>[ ,26 ]</td>
<td>[ ,27 ]</td>
</tr>
<tr>
<td></td>
<td>0.9343899</td>
<td>0.9769821</td>
<td>1.0044486</td>
<td>0.9936816</td>
<td>0.9312600</td>
<td>1.0015410</td>
</tr>
<tr>
<td></td>
<td>[ ,29 ]</td>
<td>[ ,30 ]</td>
<td>[ ,31 ]</td>
<td>[ ,32 ]</td>
<td>[ ,33 ]</td>
<td>[ ,34 ]</td>
</tr>
<tr>
<td></td>
<td>0.9829932</td>
<td>0.9026741</td>
<td>0.9875233</td>
<td>0.9897096</td>
<td>1.0381059</td>
<td>1.0154107</td>
</tr>
<tr>
<td></td>
<td>[ ,35 ]</td>
<td>[ ,36 ]</td>
<td>[ ,37 ]</td>
<td>[ ,38 ]</td>
<td>[ ,39 ]</td>
<td>[ ,40 ]</td>
</tr>
<tr>
<td></td>
<td>1.035106</td>
<td>0.9723714</td>
<td>0.9872612</td>
<td>1.0320425</td>
<td>0.9980688</td>
<td>0.9654896</td>
</tr>
<tr>
<td></td>
<td>[ ,42 ]</td>
<td>[ ,43 ]</td>
<td>[ ,44 ]</td>
<td>[ ,45 ]</td>
<td>[ ,46 ]</td>
<td>[ ,47 ]</td>
</tr>
<tr>
<td></td>
<td>0.9815276</td>
<td>0.9702355</td>
<td>1.0226387</td>
<td>1.0020321</td>
<td>0.9937074</td>
<td>1.0040726</td>
</tr>
<tr>
<td></td>
<td>[ ,49 ]</td>
<td>[ ,50 ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,</td>
<td>[ ,1 ]</td>
<td>[ ,2 ]</td>
<td>[ ,3 ]</td>
<td>[ ,4 ]</td>
<td>[ ,5 ]</td>
<td>[ ,6 ]</td>
</tr>
<tr>
<td></td>
<td>1.029798</td>
<td>0.9739493</td>
<td>0.967364</td>
<td>0.9735461</td>
<td>0.9903999</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,8 ]</td>
<td>[ ,9 ]</td>
<td>[ ,10 ]</td>
<td>[ ,11 ]</td>
<td>[ ,12 ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0401901</td>
<td>0.9646683</td>
<td>1.018345</td>
<td>1.008636</td>
<td>0.9794036</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,15 ]</td>
<td>[ ,16 ]</td>
<td>[ ,17 ]</td>
<td>[ ,18 ]</td>
<td>[ ,19 ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0152848</td>
<td>0.9940748</td>
<td>0.975392</td>
<td>1.000894</td>
<td>0.9956020</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,22 ]</td>
<td>[ ,23 ]</td>
<td>[ ,24 ]</td>
<td>[ ,25 ]</td>
<td>[ ,26 ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0312869</td>
<td>0.9738103</td>
<td>0.9961238</td>
<td>0.9606777</td>
<td>0.9919338</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,28 ]</td>
<td>[ ,29 ]</td>
<td>[ ,30 ]</td>
<td>[ ,31 ]</td>
<td>[ ,32 ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.9889468</td>
<td>0.9734963</td>
<td>0.9970932</td>
<td>1.0249025</td>
<td>1.0176648</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,35 ]</td>
<td>[ ,36 ]</td>
<td>[ ,37 ]</td>
<td>[ ,38 ]</td>
<td>[ ,39 ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0074233</td>
<td>0.9026346</td>
<td>0.9068831</td>
<td>0.9564391</td>
<td>1.007223</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,42 ]</td>
<td>[ ,43 ]</td>
<td>[ ,44 ]</td>
<td>[ ,45 ]</td>
<td>[ ,46 ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0436344</td>
<td>0.9928135</td>
<td>1.008722</td>
<td>0.919352</td>
<td>1.0116109</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,49 ]</td>
<td>[ ,50 ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
> meanALPHA
  , , 1

    [,1]    [,2]
[1,] 0.9788637 1.017082

  , , 2

    [,1]    [,2]
[1,] 1.002476 1.021888

  , , 3

    [,1]    [,2]
[1,] 0.9627315 0.9861428

  , , 4

    [,1]    [,2]
[1,] 0.9772631 0.982731

> meanPROP
[1,] 0.1033676 0.2023138 0.2983841 0.3959346

> meanPSI
  , , 1

    [,1]    [,2]
[1,] 0.607627390 0.003630216
[2,] 0.003630216 0.590420068

  , , 2

    [,1]    [,2]
[1,] 0.628055866 -0.007607933
[2,] -0.007607933 0.581837537

  , , 3

    [,1]    [,2]
[1,] 0.61040929 -0.01197769
[2,] -0.01197769 0.58957065
250

\[ \begin{array}{cccc}
[1,] & [2] \\
[1,] & 0.590004792 & 0.008437598 \\
[2,] & 0.008437598 & 0.613756808 \\
\end{array} \]

\[
> \text{meanZfix}
\]

\[
\begin{array}{cccc}
[1,] & 1 & 0 & 0 & 0 \\
[2,] & 1 & 0 & 0 & 0 \\
[3,] & 1 & 0 & 0 & 0 \\
[4,] & 1 & 0 & 0 & 0 \\
[5,] & 0 & 1 & 0 & 0 \\
[6,] & 0 & 1 & 0 & 0 \\
[7,] & 0 & 1 & 0 & 0 \\
[8,] & 0 & 1 & 0 & 0 \\
[9,] & 0 & 1 & 0 & 0 \\
[10,] & 0 & 1 & 0 & 0 \\
[11,] & 0 & 1 & 0 & 0 \\
[12,] & 0 & 1 & 0 & 0 \\
[13,] & 0 & 1 & 0 & 0 \\
[14,] & 0 & 1 & 0 & 0 \\
[15,] & 0 & 0 & 1 & 0 \\
[16,] & 0 & 0 & 1 & 0 \\
[17,] & 0 & 0 & 1 & 0 \\
[18,] & 0 & 0 & 1 & 0 \\
[19,] & 0 & 0 & 1 & 0 \\
[20,] & 0 & 0 & 1 & 0 \\
[21,] & 0 & 0 & 1 & 0 \\
[22,] & 0 & 0 & 1 & 0 \\
[23,] & 0 & 0 & 1 & 0 \\
[24,] & 0 & 0 & 1 & 0 \\
[25,] & 0 & 0 & 1 & 0 \\
[26,] & 0 & 0 & 1 & 0 \\
[27,] & 0 & 0 & 1 & 0 \\
[28,] & 0 & 0 & 1 & 0 \\
[29,] & 0 & 0 & 1 & 0 \\
[30,] & 0 & 0 & 1 & 0 \\
[31,] & 0 & 0 & 1 & 0 \\
[32,] & 0 & 0 & 1 & 0 \\
[33,] & 0 & 0 & 1 & 0 \\
[34,] & 0 & 0 & 1 & 0 \\
[35,] & 0 & 0 & 1 & 0 \\
[36,] & 0 & 0 & 1 & 0 \\
[37,] & 0 & 0 & 1 & 0 \\
[38,] & 0 & 0 & 1 & 0 \\
\end{array}
\]
> error_k
[1] 0.0397217

> error_phi
[1] 0.03881216

> CI_k
[1] 3.326979 4.378621

> CI_phi
[1] 3.582341 4.123259

%========================================================================
%========================================================================

54. P5_S50_T5_C4

rhat_k
[1] 0.9999068

> rhat_phi
[1] 1.000138

> meanK
[1] 3.8486

> meanPHI
[1] 1.00741

> meanETA
, , 1
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1,1</td>
<td>1.006335</td>
<td>0.9578548</td>
<td>1.025360</td>
<td>1.0183576</td>
<td>0.9667125</td>
<td>1.0224025</td>
</tr>
<tr>
<td>2,1</td>
<td>1.025344</td>
<td>1.0188470</td>
<td>1.006898</td>
<td>0.9849039</td>
<td>0.9921832</td>
<td>0.9751904</td>
</tr>
<tr>
<td>1,2</td>
<td>0.9897847</td>
<td>0.9699908</td>
<td>0.9805308</td>
<td>1.0215055</td>
<td>0.9817315</td>
<td>0.9788525</td>
</tr>
<tr>
<td>2,2</td>
<td>1.0254322</td>
<td>0.9872879</td>
<td>1.0124300</td>
<td>0.9761567</td>
<td>0.9944473</td>
<td>1.0320059</td>
</tr>
<tr>
<td>1,3</td>
<td>0.9980779</td>
<td>1.021814</td>
<td>0.9734724</td>
<td>0.968979</td>
<td>0.9719832</td>
<td>0.9286924</td>
</tr>
<tr>
<td>2,3</td>
<td>0.9973710</td>
<td>1.039098</td>
<td>0.963108</td>
<td>0.945920</td>
<td>0.9931294</td>
<td>0.9367498</td>
</tr>
<tr>
<td>1,4</td>
<td>0.9718396</td>
<td>0.9832103</td>
<td>1.036994</td>
<td>1.011712</td>
<td>0.9777731</td>
<td>0.9995127</td>
</tr>
<tr>
<td>2,4</td>
<td>0.9715388</td>
<td>1.0460555</td>
<td>0.990186</td>
<td>1.008846</td>
<td>0.9770464</td>
<td>1.0321545</td>
</tr>
<tr>
<td>1,5</td>
<td>1.003928</td>
<td>1.0102917</td>
<td>1.0285795</td>
<td>1.001454</td>
<td>0.9688433</td>
<td>0.9942132</td>
</tr>
<tr>
<td>2,5</td>
<td>1.017098</td>
<td>0.9962284</td>
<td>0.9709553</td>
<td>1.019244</td>
<td>0.945823</td>
<td>0.9795592</td>
</tr>
<tr>
<td>1,6</td>
<td>0.990889</td>
<td>0.9421737</td>
<td>0.9961921</td>
<td>0.9047707</td>
<td>0.997057</td>
<td>0.9990838</td>
</tr>
<tr>
<td>2,6</td>
<td>1.001982</td>
<td>0.9927480</td>
<td>0.9920258</td>
<td>0.988942</td>
<td>0.9931294</td>
<td>0.9367498</td>
</tr>
<tr>
<td>1,7</td>
<td>1.0437478</td>
<td>0.9577794</td>
<td>0.9792980</td>
<td>1.010775</td>
<td>0.9616090</td>
<td>0.9162333</td>
</tr>
<tr>
<td>2,7</td>
<td>0.9823834</td>
<td>1.0008221</td>
<td>0.9739275</td>
<td>1.001027</td>
<td>0.9393745</td>
<td>0.9765843</td>
</tr>
<tr>
<td>1,8</td>
<td>1.009322</td>
<td>0.906386</td>
<td>0.9994369</td>
<td>1.022362</td>
<td>0.9822768</td>
<td>1.000606</td>
</tr>
<tr>
<td>2,8</td>
<td>1.0119555</td>
<td>1.003397</td>
<td>0.991537</td>
<td>1.0326345</td>
<td>0.9797912</td>
<td>0.9770433</td>
</tr>
<tr>
<td>1,9</td>
<td>1.0437478</td>
<td>0.9577794</td>
<td>0.9792980</td>
<td>1.010775</td>
<td>0.9616090</td>
<td>0.9162333</td>
</tr>
<tr>
<td>2,9</td>
<td>0.9900938</td>
<td>1.0066582</td>
<td>0.9682055</td>
<td>0.9758166</td>
<td>1.031820</td>
<td>0.21392</td>
</tr>
<tr>
<td>1,10</td>
<td>1.0564558</td>
<td>0.9899628</td>
<td>1.021834</td>
<td>1.0276886</td>
<td>1.013588</td>
<td>1.059399</td>
</tr>
<tr>
<td>2,10</td>
<td>0.990889</td>
<td>0.9421737</td>
<td>0.9961921</td>
<td>0.9047707</td>
<td>0.997057</td>
<td>0.9990838</td>
</tr>
<tr>
<td>1,11</td>
<td>0.9952107</td>
<td>0.9980395</td>
<td>0.9989339</td>
<td>0.9773285</td>
<td>0.9735798</td>
<td>0.9710150</td>
</tr>
<tr>
<td>2,11</td>
<td>1.0349207</td>
<td>1.0008134</td>
<td>0.9292649</td>
<td>1.0225027</td>
<td>0.9914709</td>
<td>0.9804542</td>
</tr>
<tr>
<td>1,12</td>
<td>1.0006379</td>
<td>1.0113580</td>
<td>1.0373237</td>
<td>1.0054589</td>
<td>0.9970368</td>
<td>1.0111258</td>
</tr>
<tr>
<td>2,12</td>
<td>0.9997879</td>
<td>0.9883599</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.9933867</td>
<td>0.9857873</td>
<td>0.9863471</td>
<td>0.9858255</td>
<td>0.9812633</td>
<td>0.9503064</td>
</tr>
<tr>
<td>---</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>2,1</td>
<td>0.9526636</td>
<td>0.9487098</td>
<td>0.955071</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,2</td>
<td>1.0152967</td>
<td>0.9759141</td>
<td>1.047353</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

,, 3

<table>
<thead>
<tr>
<th></th>
<th>0.9796925</th>
<th>1.001600</th>
<th>0.9796934</th>
<th>1.015895</th>
<th>0.9950339</th>
<th>1.035089</th>
<th>1.022424</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,1</td>
<td>1.0346311</td>
<td>0.967208</td>
<td>0.9547935</td>
<td>1.003496</td>
<td>1.0020414</td>
<td>1.033068</td>
<td>0.990481</td>
</tr>
<tr>
<td>2,2</td>
<td>1.008519</td>
<td>0.9955953</td>
<td>0.9817402</td>
<td>0.9661849</td>
<td>1.015750</td>
<td>0.9752481</td>
<td>0.9857024</td>
</tr>
</tbody>
</table>

,, 4

<table>
<thead>
<tr>
<th></th>
<th>0.9770708</th>
<th>1.014828</th>
<th>1.045969</th>
<th>1.0232357</th>
<th>0.993329</th>
<th>1.004656</th>
<th>0.9973678</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,1</td>
<td>0.922928</td>
<td>0.9521267</td>
<td>0.9741976</td>
<td>0.9793419</td>
<td>0.9633983</td>
<td>0.9945132</td>
<td>0.9573348</td>
</tr>
<tr>
<td>2,2</td>
<td>1.036319</td>
<td>0.0081909</td>
<td>0.9182677</td>
<td>0.9773330</td>
<td>0.9916593</td>
<td>1.0008205</td>
<td>0.9955500</td>
</tr>
</tbody>
</table>

,, 5

<table>
<thead>
<tr>
<th></th>
<th>1.0257926</th>
<th>1.001832</th>
<th>1.0096321</th>
<th>1.0059793</th>
<th>0.9830963</th>
<th>0.9627739</th>
<th>1.001301</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,1</td>
<td>0.9574879</td>
<td>0.993043</td>
<td>0.9850251</td>
<td>0.9677148</td>
<td>0.9719698</td>
<td>1.0068813</td>
<td>0.977093</td>
</tr>
<tr>
<td>2,2</td>
<td>1.0275916</td>
<td>1.001502</td>
<td>1.0096321</td>
<td>1.0059793</td>
<td>0.9830963</td>
<td>0.9627739</td>
<td>1.001301</td>
</tr>
</tbody>
</table>

,, 6

<table>
<thead>
<tr>
<th></th>
<th>1.0177023</th>
<th>0.9951258</th>
<th>0.9913092</th>
<th>0.9919466</th>
<th>1.0225060</th>
<th>1.0021121</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2,1</td>
<td>0.9971819</td>
<td>1.0418542</td>
<td>0.9929371</td>
<td>0.9925953</td>
<td>0.9634607</td>
<td>0.9884988</td>
<td></td>
</tr>
<tr>
<td>2,2</td>
<td>0.9666876</td>
<td>0.9865862</td>
<td>1.009371</td>
<td>1.003123</td>
<td>0.9722219</td>
<td>1.0112555</td>
<td>1.0071127</td>
</tr>
</tbody>
</table>

,, 7

<table>
<thead>
<tr>
<th></th>
<th>1.0115007</th>
<th>1.0408670</th>
<th>0.985434</th>
<th>1.011864</th>
<th>0.9912252</th>
<th>0.9966716</th>
<th>0.9514627</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,1</td>
<td>0.9859676</td>
<td>0.9858632</td>
<td>1.009371</td>
<td>1.003123</td>
<td>0.9722219</td>
<td>1.0112555</td>
<td>1.0071127</td>
</tr>
</tbody>
</table>

,, 8

<table>
<thead>
<tr>
<th></th>
<th>0.9865419</th>
<th>1.000343</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2,1</td>
<td>1.0206895</td>
<td>0.968592</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,2</td>
<td>0.9865419</td>
<td>1.000343</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

,, 9

<table>
<thead>
<tr>
<th></th>
<th>0.9858355</th>
<th>0.9713021</th>
<th>0.9926251</th>
<th>0.9787021</th>
<th>0.9783291</th>
<th>0.9573100</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2,1</td>
<td>1.0213096</td>
<td>1.0096795</td>
<td>1.0009892</td>
<td>0.9563406</td>
<td>0.9637820</td>
<td>0.9883558</td>
<td></td>
</tr>
<tr>
<td>2,2</td>
<td>0.9686617</td>
<td>1.000201</td>
<td>0.9739725</td>
<td>1.0438664</td>
<td>0.9883895</td>
<td>0.9966612</td>
<td>1.0391589</td>
</tr>
</tbody>
</table>

,, 10

<table>
<thead>
<tr>
<th></th>
<th>1.008876</th>
<th>0.922005</th>
<th>0.9683765</th>
<th>0.980906</th>
<th>1.0340203</th>
<th>0.9849894</th>
<th>0.9387864</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,1</td>
<td>0.9686617</td>
<td>1.000201</td>
<td>0.9739725</td>
<td>1.0438664</td>
<td>0.9883895</td>
<td>0.9966612</td>
<td>1.0391589</td>
</tr>
<tr>
<td>2,2</td>
<td>1.000343</td>
<td>0.9865419</td>
<td>1.000343</td>
<td>1.000343</td>
<td>0.9722219</td>
<td>1.0112555</td>
<td>1.0071127</td>
</tr>
</tbody>
</table>
> meanALPHA
   [,1]   [,2]
[1,] 1.015359 0.9464643

> meanPROP
[1,] 0.09980214 0.2020755 0.2952048 0.4029176

> meanPSI
   [,1]   [,2]
[1,] 0.594972743 0.004234875
[2,] 0.004234875 0.598231141

> meanPSI
   [,1]   [,2]
[1,] 0.9627186 0.976580
[2,] 0.9794468 0.9961006
\[
\begin{align*}
\text{[,1]} & \quad 0.6103779 \quad 0.0115621 \\
\text{[,2]} & \quad 0.0115621 \quad 0.5666974 \\
\text{[,3]} & \\
\text{[,4]} & \\
\text{[,1]} & \quad 0.59055378 \quad -0.01082234 \\
\text{[,2]} & \quad -0.01082234 \quad 0.60610554 \\
\text{[,3]} & \\
\text{[,4]} & \\
\text{[,1]} & \quad 0.619707635 \quad -0.008997817 \\
\text{[,2]} & \quad -0.008997817 \quad 0.581318529 \\
\text{[,3]} & \\
\text{[,4]} & \\
> \text{meanZfix} \\
\begin{array}{cccc}
\text{[,1]} & \text{[,2]} & \text{[,3]} & \text{[,4]} \\
\text{[1,]} & 1 & 0 & 0 & 0 \\
\text{[2,]} & 1 & 0 & 0 & 0 \\
\text{[3,]} & 1 & 0 & 0 & 0 \\
\text{[4,]} & 1 & 0 & 0 & 0 \\
\text{[5,]} & 0 & 1 & 0 & 0 \\
\text{[6,]} & 0 & 1 & 0 & 0 \\
\text{[7,]} & 0 & 1 & 0 & 0 \\
\text{[8,]} & 0 & 1 & 0 & 0 \\
\text{[9,]} & 0 & 1 & 0 & 0 \\
\text{[10,]} & 0 & 1 & 0 & 0 \\
\text{[11,]} & 0 & 1 & 0 & 0 \\
\text{[12,]} & 0 & 1 & 0 & 0 \\
\text{[13,]} & 0 & 1 & 0 & 0 \\
\text{[14,]} & 0 & 1 & 0 & 0 \\
\text{[15,]} & 0 & 0 & 1 & 0 \\
\text{[16,]} & 0 & 0 & 1 & 0 \\
\text{[17,]} & 0 & 0 & 1 & 0 \\
\text{[18,]} & 0 & 0 & 1 & 0 \\
\text{[19,]} & 0 & 0 & 1 & 0 \\
\text{[20,]} & 0 & 0 & 1 & 0 \\
\text{[21,]} & 0 & 0 & 1 & 0 \\
\text{[22,]} & 0 & 0 & 1 & 0 \\
\text{[23,]} & 0 & 0 & 1 & 0 \\
\text{[24,]} & 0 & 0 & 1 & 0 \\
\text{[25,]} & 0 & 0 & 1 & 0 \\
\text{[26,]} & 0 & 0 & 1 & 0 \\
\text{[27,]} & 0 & 0 & 1 & 0 \\
\text{[28,]} & 0 & 0 & 1 & 0 \\
\text{[29,]} & 0 & 0 & 1 & 0 \\
\text{[30,]} & 0 & 0 & 0 & 1 \\
\end{array}
\end{align*}
\]
> error_k
[1] 0.006928203

> error_phi
[1] 0.02177773

> CI_k
[1] 3.329947 4.367253

> CI_phi
[1] 3.571878 4.125322

#==========================================
#==========================================

Procedure 89: DU3_S15_T5_C3

rhat_k
[1] 0.9999579

> rhat_phi
[1] 0.999942

> meanK
[1] 3.4804
> meanPHI
[1] 0.993738

> meanETA
,, 1

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.036401</td>
<td>0.9845132</td>
<td>1.009968</td>
<td>1.008844</td>
<td>0.9961716</td>
<td>1.0279914</td>
<td>1.032708</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.005410</td>
<td>1.0233489</td>
<td>0.993797</td>
<td>1.023074</td>
<td>1.0302444</td>
<td>0.9963456</td>
<td>1.016548</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.0048185</td>
<td>0.9575859</td>
<td>0.9951811</td>
<td>0.9885589</td>
<td>0.9727229</td>
<td>0.9858003</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9956318</td>
<td>1.0249138</td>
<td>1.0344522</td>
<td>1.0010917</td>
<td>1.0250716</td>
<td>1.0157339</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>[,14]</th>
<th>[,15]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.0135876</td>
<td>1.0000068</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9846002</td>
<td>0.9623524</td>
</tr>
</tbody>
</table>

,, 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>0.999634</td>
<td>1.0207815</td>
<td>1.009462</td>
<td>1.0128822</td>
<td>0.9663163</td>
<td>1.0216261</td>
<td>0.9938220</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.002035</td>
<td>0.9872403</td>
<td>1.024656</td>
<td>0.9838088</td>
<td>1.0058920</td>
<td>0.9699677</td>
<td>0.9911099</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>0.940674</td>
<td>0.962187</td>
<td>0.9973457</td>
<td>1.0426814</td>
<td>1.013042</td>
<td>0.9957472</td>
<td>1.023051</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9938695</td>
<td>1.0460427</td>
<td>0.9974247</td>
<td>0.9856744</td>
<td>1.024600</td>
<td>1.0231820</td>
<td>1.010792</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>[,15]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>0.9838262</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.0148097</td>
</tr>
</tbody>
</table>

,, 3

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>0.9896354</td>
<td>0.9997936</td>
<td>0.9838119</td>
<td>0.9672128</td>
<td>0.989629</td>
<td>1.0116011</td>
<td>0.9976149</td>
</tr>
<tr>
<td>[2,]</td>
<td>1.0185057</td>
<td>0.9843123</td>
<td>0.9819058</td>
<td>0.9913111</td>
<td>0.979444</td>
<td>0.9698568</td>
<td>1.0101293</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.0088331</td>
<td>0.9844598</td>
<td>1.0027046</td>
<td>0.9733792</td>
<td>1.0038755</td>
<td>0.9845948</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9777182</td>
<td>0.9375149</td>
<td>0.9821667</td>
<td>1.0288696</td>
<td>0.9670698</td>
<td>1.0057966</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>[,14]</th>
<th>[,15]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>0.9649523</td>
<td>0.9989557</td>
</tr>
<tr>
<td>[2,]</td>
<td>0.9621066</td>
<td>1.0288140</td>
</tr>
</tbody>
</table>

> meanALPHA

,, 1

<table>
<thead>
<tr>
<th></th>
<th>[,1]</th>
<th>[,2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>1.021397</td>
<td>0.9899873</td>
</tr>
</tbody>
</table>
[, 2

          [,1]   [,2]
[1,]  0.9601938 1.025939

[, 3

          [,1]   [,2]
[1,]  1.02178 1.00007

> meanPROP

          [,1]   [,2]   [,3]
[1,]  0.1692805 0.3346982 0.4960213

> meanPSI

[, 1

          [,1]   [,2]
[1,]  0.58483772 -0.01320339
[2,] -0.01320339  0.59899893

[, 2

          [,1]   [,2]
[1,]  0.4371731  0.1426441
[2,]  0.1420550  0.4331753

[, 3

          [,1]   [,2]
[1,]  0.59614888 -0.01354471
[2,] -0.01354471  0.62852534

> meanZfix

          [,1]   [,2]   [,3]
[1,]  1 0 0
[2,]  1 0 0
[3,]  0 1 0
[4,]  0 1 0
[5,]  0 1 0
[6,]  0 1 0
[7,]  0 1 0
[8,]  0 0 1
> error_k
[1] 0.02101555

> error_phi
[1] 0.009022909

> CI_k
[1] 2.49223 4.46857

> CI_phi
[1] 2.982337 3.978463

%========================================================================
%========================================================================

Procedure 98: DU4_S15_T5_C3

rhat_k
[1] 0.9999007

> rhat_phi
[1] 0.9999219

> meanK
[1] 3.4074

> meanPHI
[1] 0.9840321

> meanETA
, , 1

[1,] 1.0272450 1.020736 0.9864407 0.9853022 0.9658070 0.9687562
[2,] 0.9646652 0.993011 0.9538680 0.9799278 1.0063210 0.9863247 1.0101377
[1,] 1.006831 1.0038153 1.011372 1.0187483 0.9994773 0.9827179 0.9865972
```
[2,]  1.023747  0.9854705  1.006860  0.9873642  1.0109773  1.0488311  1.0247580
  [,15]
[1,]  1.0028341
[2,]  0.9781219

, , 2

[1,]  1.0223979  0.9958772  0.9939507  1.0494799  1.0237678  1.021572  0.9825644
[2,]  0.9785959  1.0030730  0.9782459  0.9707114  1.024553  0.9616314

[1,]  0.9872353  1.0408433  0.9922023  0.9893781  0.9934045  1.0227580  1.0306439
[2,]  0.9875899  0.9986042  1.0061397  0.9930368  0.9827545  0.9986492  0.9895358

[,15]
[1,]  1.0361874
[2,]  0.9807423

, , 3

[1,]  1.0347746  0.9871951  1.001897  0.9891016  0.9865856  1.043622  1.034405
[2,]  0.9785495  0.9599064  1.009992  1.0220796  0.9543157  1.015461  1.019593

[1,]  0.993571  1.005359  1.002876  0.9685905  1.005048  1.0348538  1.0179140
[2,]  1.012411  1.007932  1.004819  0.9789130  1.030444  0.9483374  0.9840232

[,15]
[1,]  1.033266
[2,]  1.006101

> meanALPHA

, , 1

[,1]  [,2]
[1,]  1.029086  1.011007

, , 2

[,1]  [,2]
[1,]  1.030017  1.010277

, , 3

[,1]  [,2]
[1,]  1.002162  1.060358

> meanPROP
```
[,1] [,2] [,3]
[1,] 0.1648269 0.3327274 0.5024457

> meanPSI

[,1] [,2]
[1,] 0.600874096 0.003284949
[2,] 0.003284949 0.604148816

[,1] [,2]
[1,] 0.4526718 0.1544584
[2,] 0.1605413 0.4566121

[,1] [,2]
[1,] 0.6118609898 0.0005390495
[2,] 0.0005390495 0.5954747983

> meanZfix

[,1] [,2] [,3]
[1,] 1 0 0
[2,] 1 0 0
[3,] 0 1 0
[4,] 0 1 0
[5,] 0 1 0
[6,] 0 1 0
[7,] 0 1 0
[8,] 0 0 1
[9,] 0 0 1
[10,] 0 0 1
[11,] 0 0 1
[12,] 0 0 1
[13,] 0 0 1
[14,] 0 0 1
[15,] 0 0 1

> error_k
[1] 0.002309401

> error_phi
Procedure 137: DU3_S200_T3_C3

rhat_k
[1] 0.9999894

> rhat_phi
[1] 0.9999715

> meanK
[1] 3.5882

> meanPHI
[1] 0.9961969

> meanETA
, , 1

[1,] 0.9613220 0.9654400 1.021047 0.9764654 1.068600 0.9859306 1.0050611
[2,] 0.9949373 0.9751458 0.998532 1.0408889 1.019402 0.9681612 0.9873712

[1,] 0.9799797 0.9936516 1.007768 0.9851112 0.9961359 0.9723867 0.9826146
[2,] 1.0256940 0.9539756 0.986535 0.9708871 0.9916934 0.9692888 1.0232532

[1,] 0.9799797 0.9936516 1.007768 0.9851112 0.9961359 0.9723867 0.9826146
[2,] 1.0015630 0.9531264 1.015940 0.9564446 1.002572 0.9544296 1.0250170

[1,] 1.0389212 0.9998079 1.0098183 0.9466945 0.9862237 1.034006 0.9872632
[2,] 0.9635369 1.0107580 0.9600485 0.9422519 0.9873848 1.023402 1.0347616

[1,] 0.994869 0.9792913 0.9501173 1.026452 0.9786621 1.009299 0.9911674
[2,] 1.040499 0.9832672 0.9711371 1.026432 1.0002375 1.027484 1.0197127

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9823280</td>
<td>0.9846284</td>
<td>0.9967281</td>
<td>0.9812153</td>
<td>0.9827952</td>
<td>0.9817284</td>
</tr>
<tr>
<td>2</td>
<td>0.9975262</td>
<td>1.0078808</td>
<td>1.0057320</td>
<td>0.9512865</td>
<td>0.9882674</td>
<td>0.9873621</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9948682</td>
<td>1.0175590</td>
<td>0.9846090</td>
<td>0.9999939</td>
<td>1.0049950</td>
<td>0.9817284</td>
</tr>
<tr>
<td>2</td>
<td>1.0454584</td>
<td>0.9955488</td>
<td>1.0117740</td>
<td>1.0275840</td>
<td>0.9862574</td>
<td>0.9873621</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9711691</td>
<td>1.0137380</td>
<td>1.0073774</td>
<td>1.0065117</td>
<td>0.9945234</td>
<td>0.9817284</td>
</tr>
<tr>
<td>2</td>
<td>1.0175248</td>
<td>1.0064873</td>
<td>1.0761330</td>
<td>1.0049950</td>
<td>0.9882674</td>
<td>0.9873621</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9759162</td>
<td>0.9907145</td>
<td>1.0175590</td>
<td>1.0057320</td>
<td>0.9512865</td>
<td>1.0307353</td>
</tr>
<tr>
<td>2</td>
<td>1.0175248</td>
<td>1.0064873</td>
<td>1.0761330</td>
<td>1.0049950</td>
<td>0.9882674</td>
<td>0.9873621</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9635426</td>
<td>1.0039101</td>
<td>1.0133530</td>
<td>1.0307253</td>
<td>1.0348390</td>
<td>1.0417987</td>
</tr>
<tr>
<td>2</td>
<td>1.0091680</td>
<td>1.0055420</td>
<td>1.0485142</td>
<td>0.9841480</td>
<td>0.9300373</td>
<td>0.9795376</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.0030744</td>
<td>0.9802177</td>
<td>1.0408190</td>
<td>1.0011390</td>
<td>0.9155404</td>
<td>0.9631662</td>
</tr>
<tr>
<td>2</td>
<td>0.9635426</td>
<td>1.0039101</td>
<td>1.0133530</td>
<td>1.0307253</td>
<td>1.0348390</td>
<td>1.0417987</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9864415</td>
<td>0.9798663</td>
<td>0.9847859</td>
<td>0.9638081</td>
<td>1.0230316</td>
<td>1.0009184</td>
</tr>
<tr>
<td>2</td>
<td>0.9875150</td>
<td>0.9926656</td>
<td>1.0036070</td>
<td>0.9763382</td>
<td>0.9607476</td>
<td>0.9847274</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9499962</td>
<td>0.9191698</td>
<td>0.9891433</td>
<td>1.0433690</td>
<td>1.0586040</td>
<td>1.0159604</td>
</tr>
<tr>
<td>2</td>
<td>0.9873390</td>
<td>0.9648747</td>
<td>0.9946111</td>
<td>0.9909615</td>
<td>1.0098860</td>
<td>1.0266800</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.0110981</td>
<td>1.0223930</td>
<td>1.0022190</td>
<td>1.0242450</td>
<td>0.9916956</td>
<td>1.0121990</td>
</tr>
<tr>
<td>2</td>
<td>0.9930998</td>
<td>1.0328910</td>
<td>1.0110230</td>
<td>1.0091240</td>
<td>1.0443260</td>
<td>0.9454762</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.0145200</td>
<td>0.1272090</td>
<td>0.9507841</td>
<td>1.0236215</td>
<td>1.0105380</td>
<td>1.0159538</td>
</tr>
<tr>
<td>2</td>
<td>0.1070390</td>
<td>0.1903210</td>
<td>0.1024562</td>
<td>0.1036907</td>
<td>0.1032110</td>
<td>0.9857990</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9645738</td>
<td>0.9876405</td>
<td>1.0126339</td>
<td>0.9141981</td>
<td>0.9867668</td>
<td>1.0066870</td>
</tr>
<tr>
<td>2</td>
<td>0.9862453</td>
<td>1.0273995</td>
<td>0.9806153</td>
<td>0.9382292</td>
<td>0.9991979</td>
<td>1.0021130</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9904193</td>
<td>0.9751972</td>
<td>0.9921930</td>
<td>1.0195609</td>
<td>0.9765341</td>
<td>0.9991810</td>
</tr>
<tr>
<td>2</td>
<td>0.9718931</td>
<td>1.0135581</td>
<td>1.0041440</td>
<td>0.9956866</td>
<td>1.0069078</td>
<td>0.9719338</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9961766</td>
<td>1.0064140</td>
<td>0.9909357</td>
<td>0.9762536</td>
<td>0.9412682</td>
<td>0.1030858</td>
</tr>
<tr>
<td>2</td>
<td>0.9319232</td>
<td>1.0360690</td>
<td>1.0788850</td>
<td>0.8959296</td>
<td>0.9388970</td>
<td>0.9914395</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.0153849</td>
<td>0.9853533</td>
<td>1.0218000</td>
<td>1.0247970</td>
<td>1.0126862</td>
<td>0.9651655</td>
</tr>
<tr>
<td>2</td>
<td>0.9948649</td>
<td>0.9792556</td>
<td>0.9950140</td>
<td>1.0226370</td>
<td>1.0016220</td>
<td>0.9876864</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9975627</td>
<td>0.9912693</td>
<td>0.9544510</td>
<td>1.0346780</td>
<td>1.0767300</td>
<td>0.9636605</td>
</tr>
<tr>
<td>2</td>
<td>1.0052034</td>
<td>0.9586234</td>
<td>1.0407400</td>
<td>1.0031400</td>
<td>0.9857775</td>
<td>0.9900373</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9914400</td>
<td>1.0647661</td>
<td>0.9720958</td>
<td>0.9914120</td>
<td>1.0038760</td>
<td>0.9728382</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>1.0023804</td>
<td>1.0223552</td>
<td>1.0063060</td>
<td>1.0002130</td>
<td>0.9923774</td>
<td>0.9749815</td>
</tr>
<tr>
<td>2</td>
<td>0.9805217</td>
<td>0.9838499</td>
<td>0.9736659</td>
<td>1.0219332</td>
<td>0.9923774</td>
<td>0.9749815</td>
</tr>
<tr>
<td>1</td>
<td>1.0510700</td>
<td>0.9822396</td>
<td>1.0089940</td>
<td>0.9632373</td>
<td>1.0150150</td>
<td>1.0006332</td>
</tr>
<tr>
<td>2</td>
<td>0.9815453</td>
<td>0.9994999</td>
<td>1.0353999</td>
<td>0.9273927</td>
<td>1.0202643</td>
<td>0.9988739</td>
</tr>
<tr>
<td>1</td>
<td>1.0510700</td>
<td>0.9822396</td>
<td>1.0089940</td>
<td>0.9632373</td>
<td>1.0150150</td>
<td>1.0006332</td>
</tr>
<tr>
<td>2</td>
<td>0.9815453</td>
<td>0.9994999</td>
<td>1.0353999</td>
<td>0.9273927</td>
<td>1.0202643</td>
<td>0.9988739</td>
</tr>
<tr>
<td>1</td>
<td>1.0006218</td>
<td>1.0018470</td>
<td>1.0440550</td>
<td>1.0477979</td>
<td>0.9799237</td>
<td>1.0130543</td>
</tr>
<tr>
<td>2</td>
<td>0.9897007</td>
<td>1.0018470</td>
<td>1.0440550</td>
<td>1.0477979</td>
<td>0.9799237</td>
<td>1.0130543</td>
</tr>
</tbody>
</table>
[1,] 1.016897 0.9979264 0.9808116 0.9903262 1.0360721 1.035343 0.9908097
[2,] 1.015400 0.9936542 0.9909475 1.0088629 0.9895367 1.033853 1.0421123
[1,] 0.9941579 0.9751727 0.9369508 0.9944841 1.0011315 0.995383 1.021218
[2,] 1.0126916 0.9840487 0.9894445 0.9960625 1.0196817 0.9890502
[1,] 0.9849084 1.021364 0.9991123 1.000114 0.9673203 0.9809032 0.9561486
[2,] 1.0325338 1.009215 0.9874058 1.028937 1.0447917 1.0196817 0.9890502
[1,] 0.9960118 1.017960 0.9581286 1.070603 1.000113 0.9917344 1.012364
[2,] 0.9950685 1.073471 0.9851734 1.044642 0.9693717 0.9945132 1.018066
[1,] 0.995629 1.010586 0.9891637 0.9739178 0.9971484 1.0230424 1.0056870
[2,] 1.018738 0.9649564 0.9862801 1.0121675 0.9525814 0.9849418 0.9949903
, , 3
[1,] 0.9786372 0.9974394 1.005149 1.0031648 1.003355 0.9879837 0.9891277
[2,] 0.9992693 0.9945909 1.044493 0.9930261 1.008935 1.0247811 1.0060500
[1,] 1.018437 1.0107101 1.100523 0.9819045 0.9956765 1.0165640 1.019093
[2,] 0.921015 0.9949378 1.002189 1.0530584 1.0410470 0.9989879 1.011408
[1,] 1.146371 0.0101701 1.0091101 1.0031648 1.003355 1.005149 1.0071403
[2,] 0.995383 1.0365640 0.9803321 1.008935 1.008935 1.008935 1.008935
[1,] 0.9804610 0.9628135 0.9632391 1.008209 1.0281318 1.018182 0.9976225
[2,] 0.9798164 0.9533139 1.015146 0.973122 0.9982306 1.003955 1.0060904
[1,] 1.0115809 1.0056400 1.021706 0.9803321 1.0095666 1.0074744 1.0140202
[2,] 0.9876802 0.9930787 1.023078 1.0208653 0.9877286 0.9814799 0.9830454
[1,] 1.0257940 0.9812133 1.0348611 0.9404307 1.0513528 1.0036268
[2,] 0.9701601 0.9736917 0.9471043 1.0629751 0.9765161 0.9774565
[1,] 1.0077541 0.9885509 0.9782284 0.9672866 1.0027019 1.010973 0.9991647
[2,] 0.9947438 0.9874994 1.0100963 0.9885889 0.9965222 1.008697 0.988466
[1,] 0.9991698 0.9972213 0.9786418 0.9844859 0.9900962 1.002929 1.050781
[2,] 1.0041693 0.9885377 0.9858643 0.9742943 0.9897280 1.044742 1.036897
<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
<th>Value 7</th>
<th>Value 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.008906</td>
<td>1.0346228</td>
<td>0.9669920</td>
<td>0.974947</td>
<td>0.9871339</td>
<td>1.006207</td>
<td>1.0084085</td>
<td></td>
</tr>
<tr>
<td>0.9694535</td>
<td>0.9725247</td>
<td>0.9976065</td>
<td>1.031924</td>
<td>1.032903</td>
<td>0.9803297</td>
<td>0.9878077</td>
<td></td>
</tr>
<tr>
<td>1.0329166</td>
<td>0.9615658</td>
<td>0.9787505</td>
<td>1.005025</td>
<td>1.012910</td>
<td>1.0018115</td>
<td>1.0489105</td>
<td></td>
</tr>
<tr>
<td>0.9866392</td>
<td>1.027047</td>
<td>1.044762</td>
<td>1.022258</td>
<td>1.0161955</td>
<td>0.9910602</td>
<td>0.9856099</td>
<td></td>
</tr>
<tr>
<td>1.010389</td>
<td>0.9671107</td>
<td>0.9996140</td>
<td>0.9856353</td>
<td>0.9915515</td>
<td>1.0201316</td>
<td>0.9503989</td>
<td></td>
</tr>
<tr>
<td>0.9751911</td>
<td>1.042883</td>
<td>1.003732</td>
<td>1.013793</td>
<td>1.0148397</td>
<td>1.0044231</td>
<td>1.0178567</td>
<td></td>
</tr>
<tr>
<td>1.0216008</td>
<td>1.031925</td>
<td>0.991237</td>
<td>1.011849</td>
<td>0.9769731</td>
<td>0.9770401</td>
<td>0.9896802</td>
<td></td>
</tr>
<tr>
<td>0.9692328</td>
<td>0.9798049</td>
<td>0.9663214</td>
<td>1.0116273</td>
<td>0.9870514</td>
<td>0.9808455</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0017983</td>
<td>0.9609216</td>
<td>0.9918889</td>
<td>1.0066595</td>
<td>0.9876787</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9555739</td>
<td>1.02888</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9608892</td>
<td>1.000416</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1690477</td>
<td>0.3365164</td>
<td>0.4944359</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> meanALPHA

, , 1

[1,] 0.9827005 1.000951

, , 2

[1,] 0.9555739 1.02888

, , 3

[1,] 0.9608892 1.000416

> meanPROP

[1,] 0.1690477 0.3365164 0.4944359

> meanPSI

, , 1
[,1]    [,2]
[1,] 0.605117568 0.006333358
[2,] 0.006333358 0.586723962

[,1]    [,2]
[1,] 0.603264113 0.003082976
[2,] 0.003082976 0.592419832

[,1]    [,2]
[1,] 0.592394863 -0.007109465
[2,] -0.007109465 0.615340006

> meanZfix

[,1]    [,2]    [,3]
[1,] 1 0 0
[2,] 1 0 0
[3,] 1 0 0
[4,] 1 0 0
[5,] 1 0 0
[6,] 1 0 0
[7,] 1 0 0
[8,] 1 0 0
[9,] 1 0 0
[10,] 1 0 0
[11,] 1 0 0
[12,] 1 0 0
[13,] 1 0 0
[14,] 1 0 0
[15,] 1 0 0
[16,] 1 0 0
[17,] 1 0 0
[18,] 1 0 0
[19,] 1 0 0
[20,] 1 0 0
[21,] 1 0 0
[22,] 1 0 0
[23,] 1 0 0
[24,] 1 0 0
[25,] 1 0 0
[26,] 1 0 0
[27,] 1 0 0
[28,] 1 0 0
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[29,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[30,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[31,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[32,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[33,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[34,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[35,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[36,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[37,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[38,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[39,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[40,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[41,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[42,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[43,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[44,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[45,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[46,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[47,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[48,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[49,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[50,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[51,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[52,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[53,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[54,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[55,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[56,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[57,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[58,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[59,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[60,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[61,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[62,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[63,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[64,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[65,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[66,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[67,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[68,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[69,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[70,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[71,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[72,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[73,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[74,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[75,]</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>76,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>77,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>78,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>79,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>80,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>81,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>82,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>83,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>84,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>85,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>86,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>87,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>88,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>89,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>90,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>91,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>92,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>93,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>94,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>95,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>96,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>97,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>98,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>99,</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>100,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>101,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>102,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>103,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>104,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>105,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>106,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>107,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>108,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>109,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>110,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>111,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>112,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>113,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>114,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>115,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>116,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>117,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>118,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>119,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>120,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>121,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>122,</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
[170,] 0 0 1
[171,] 0 0 1
[172,] 0 0 1
[173,] 0 0 1
[174,] 0 0 1
[175,] 0 0 1
[176,] 0 0 1
[177,] 0 0 1
[178,] 0 0 1
[179,] 0 0 1
[180,] 0 0 1
[181,] 0 0 1
[182,] 0 0 1
[183,] 0 0 1
[184,] 0 0 1
[185,] 0 0 1
[186,] 0 0 1
[187,] 0 0 1
[188,] 0 0 1
[189,] 0 0 1
[190,] 0 0 1
[191,] 0 0 1
[192,] 0 0 1
[193,] 0 0 1
[194,] 0 0 1
[195,] 0 0 1
[196,] 0 0 1
[197,] 0 0 1
[198,] 0 0 1
[199,] 0 0 1
[200,] 0 0 1

> error_k
[1] 0.02471059

>

> error_phi
[1] 0.01181331

> CI_k
[1] 3.332135 3.844265

> CI_phi
[1] 3.451259 3.725141

>