

UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Identification of Latent Classes in Mixture Models: A Monte Carlo Simulation Study

Permalink

<https://escholarship.org/uc/item/10d652p1>

Author

McDonald, Stefanie R.

Publication Date

2013

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Identification of Latent Classes in Mixture Models:
A Monte Carlo Simulation Study

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Education

by

Stefanie R. McDonald

June 2013

Dissertation Committee:

Dr. George A. Marcoulides, Chairperson

Dr. H. Lee Swanson

Dr. Marsha Ing

Copyright by
Stefanie R. McDonald
2013

The Dissertation of Stefanie R. McDonald is approved:

Committee Chairperson

University of California, Riverside

To my family and friends who walked beside me on this journey, even carrying me part
of the way.

ABSTRACT OF THE DISSERTATION

Identification of Latent Classes in Mixture Models:
A Monte Carlo Simulation Study

by

Stefanie R. McDonald

Doctor of Philosophy, Graduate Program in Education
University of California, Riverside, June 2013
Dr. George A. Marcoulides, Chairperson

Factor mixture modeling is an increasingly popular method used in applied research settings that combines latent class models and common factor models. The method is often used to detect the presence of heterogeneous populations (i.e., latent classes) that differ on one or more population parameters of a factor model in a sample that lacks *a priori* information regarding population membership. Researchers have faced many challenges when using the factor mixture modeling method. The present study employs a Monte Carlo approach to compare the performance of factor mixture modeling with three other tenable methods: Q-Factor Analysis, Marcoulides and Drezner Model, and McDonald Eigen Analysis, in detecting latent classes in a sample. Data were simulated by manipulating six design factors: sample size, number of classes, mixing proportion, class separation, distributional characteristics, and complexity of within-class factor structures. The performance of each method was assessed by determining the proportion of replications for which each method detected the correct number of latent classes (i.e., populations) and Cohen's h effect size for each manipulated design factor.

Overall, the factor mixture model performed best. In particular, the sample size, number of classes, and distributional characteristic of the latent variable significantly impacted the ability of factor mixture model to detect the correct number of classes. The McDonald Eigen Analysis performed second best, overall. The method was only significantly influenced by the number of classes. The other design conditions had a small or negligent influence on the performance. Q-Factor Analysis was influenced most by the complexity of the within-class factor structure and the number of classes. Finally, Marcoulides and Drezner Model was most influenced by the sample size and the number of classes.

From the results, it can be concluded that under certain design conditions, less parameterized and less complex methods perform just as well as factor mixture modeling in detecting latent classes. Researchers can be guided in determining the best method to use based on the characteristics of their data.

Table of Contents

ABSTRACT OF THE DISSERTATION	v
Chapter 1: Introduction	1
Mixture Modeling	1
Statement of Problem	4
Significance of the Study	6
Chapter 2: Review of Heterogeneous Methods	8
Heterogeneity Observed or Unobserved	8
Hypothesized Model	11
Latent Variables in Model	11
Type of Observed Variables	12
Chapter 3: Review of Methods Used in Study	14
Q-Factor Analysis	14
Factor Mixture Model	23
Marcoulides and Drezner Model	31
McDonald Eigen Analysis	34
Chapter 4: Method	38
Manipulated Design Conditions	38
Invariant Design Conditions	43
Data Generation	47
Outcome Measures	53
Software Packages	54

Chapter 5: Results	55
Factor Mixture Model Results	55
Q-Factor Analysis Results	65
Marcoulides and Drezner Model Results	75
McDonald Eigen Analysis Results	85
Comparison of Results	96
Effect Sizes	102
Chapter 6: Discussion	104
Limitations of the Study	107
Suggestions for Future Research	109
References	110
Appendix A	117
Appendix B	120
Appendix C	121
Appendix D	122
Appendix E	123

List of Figures

1	Mixture Model Examples.	2
2	Common Factor Model.	16
3	Common Factor Model.	25
4	Latent Class Model.	26
5	Factor Mixture Model.	28
6	Distance Between and Within Clusters.	32
7	One-Factor Mixture Model.	40
8	Two-Factor Mixture Model.	41
9	Two-Factor Mixture Model with Cross-Loadings.	41
10	Three-Factor Mixture Model.	42

List of Tables

1	Methods Used for Heterogeneous Populations	13
2	Factor Scores	47
3	Factor Scores After Correlation and Non-Normality Applied	50
4	Factor Scores After Correlation, Non-Normality, and Class Separation Applied	51
5	Observed Data Variables Generated Using Factor Scores	52
6	FMM Results - Model 1 with Normal Data	56
7	FMM Results - Model 2 with Normal Data	57
8	FMM Results - Model 2c with Normal Data	58
9	FMM Results - Model 3 with Normal Data	60
10	FMM Results - Model 1 with Non-Normal Data	61
11	FMM Results - Model 2 with Non-Normal Data	62
12	FMM Results - Model 2c with Non-Normal Data	63
13	FMM Results - Model 3 with Non-Normal Data	64
14	QFA Results - Model 1 with Normal Data	66
15	QFA Results - Model 2 with Normal Data	67
16	QFA Results - Model 2c with Normal Data	68
17	QFA Results - Model 3 with Normal Data	70
18	QFA Results - Model 1 with Non-Normal Data	71
19	QFA Results - Model 2 with Non-Normal Data	72
20	QFA Results - Model 2c with Non-Normal Data	73

21	QFA Results - Model 3 with Non-Normal Data	75
22	MDM Results - Model 1 with Normal Data	76
23	MDM Results - Model 2 with Normal Data	77
24	MDM Results - Model 2c with Normal Data	79
25	MDM Results - Model 3 with Normal Data	80
26	MDM Results - Model 1 with Non-Normal Data	81
27	MDM Results - Model 2 with Non-Normal Data	82
28	MDM Results - Model 2c with Non-Normal Data	84
29	MDM Results - Model 3 with Non-Normal Data	85
30	MEA Results - Model 1 with Normal Data	86
31	MEA Results - Model 2 with Normal Data	88
32	MEA Results - Model 2c with Normal Data	89
33	MEA Results - Model 3 with Normal Data	90
34	MEA Results - Model 1 with Non-Normal Data	92
35	MEA Results - Model 2 with Non-Normal Data	93
36	MEA Results - Model 2c with Non-Normal Data	94
37	MEA Results - Model 3 with Non-Normal Data	95
38	Performance Rate by Method	96
39	Performance Rate by Sample Size	97
40	Performance Rate by Number of Classes	98
41	Performance Rate by Mixing Proportion	99
42	Performance Rate by Distance Between Classes	100
43	Performance Rate by Distributional Characteristic	101
44	Performance Rate by Model Complexity	102
45	Effect Sizes of Manipulated Design Variables	103

Chapter 1: Introduction

Researchers are often faced with the challenging task of investigating data from heterogeneous populations for which the sources of heterogeneity are unobserved and *a priori* information regarding population membership is unknown. Although different methodological approaches can be taken to deal with such data, there has been an increased interest in mixture modeling. Mixture modeling is a method designed to deal with data that result from a heterogeneous population in which each population potentially has distinct distributional parameters (McLachlan & Peel, 2000).

One of the earliest attempts involving mixture analysis was carried out by Karl Pearson (1894; 1895). Pearson (1894) fitted a mixture of two normal probability functions to Bay of Naples crab data provided by biologist Walter F. R. Weldon. Weldon (1892) speculated that asymmetry in the histogram of the ratio of forehead to body length of the crabs could indicate evolutionary changes. Pearson (1894) identified two distinct subpopulations within the data. To analyze the mixture data, Pearson used the moment matching technique. This technique posed a significant computational challenge at the time. However, with the processing power provided by the advances of modern day computers and maximum likelihood estimation techniques, interest in the mixture modeling method has been revived. Currently, there is a vast amount of research in mixture modeling spanning various academic fields of study (e.g., agriculture, astronomy, biology, finance, education, economics medicine, psychiatric, and engineering; McLachlan & Peel, 2000).

Mixture Modeling

As basic examples of mixture modeling, consider Figure 1, which presents two univariate data sets. Figure 1a describes the fitted normal distributions for two samples with unequal means and equal variances. Figure 1b describes the data as if it had come

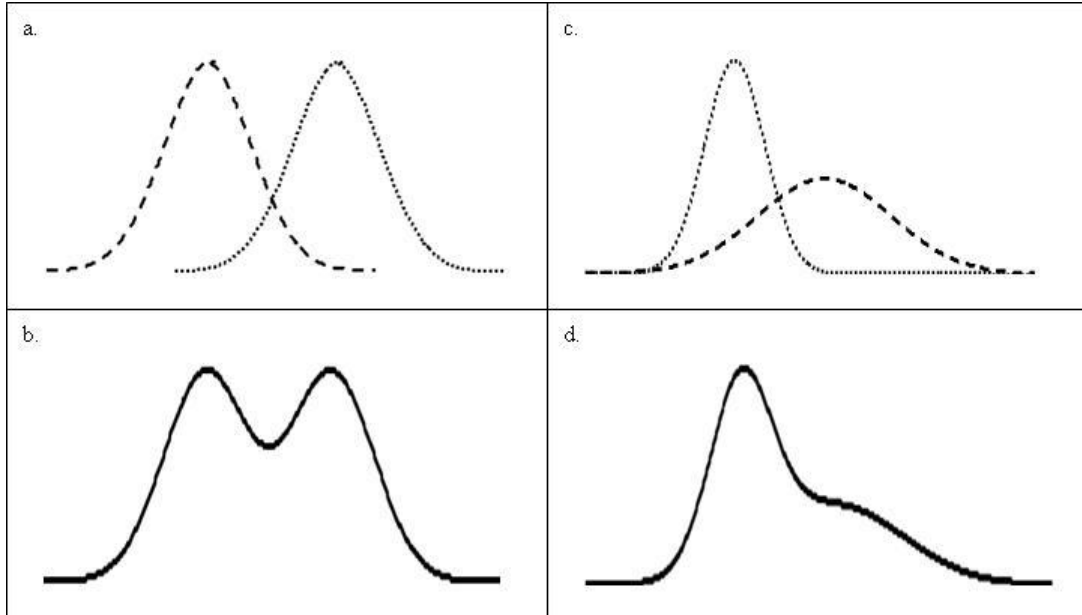


Figure 1. a. Two overlapping normal distributions with unequal means and equal variances. b. Two overlapping normal distributions with unequal means and equal variances depicted as if the data had come from one non-normal distribution. c. Two overlapping normal distributions with unequal means and unequal variances. d. Two overlapping normal distributions with unequal means and unequal variances depicted as if the data had come from one non-normal distribution.

from one population. Similarly, Figure 1c describes the fitted normal distributions for two samples with unequal means and unequal variances, while Figure 1d describes the data as if it had come from one population. In both examples, the actual data had come from two distinct populations; they are examples from heterogeneous data sets. The single non-normal distributions do not fit as well as the mixture of the two normal distributions. Had the data included *a priori* information on the membership of each population, the mean and variance of each population could be readily analyzed and compared. However, population membership was not known *a priori*. Thus, mixture modeling could be called upon to determine the nature of the mixture and the parameter estimates of interest (Gagné, 2006).

Most statistical procedures assume that a sample comes from a homogeneous population (Bauer & Curran, 2004; Yung, 1997). Population homogeneity implies that all individuals in the population come from the same distribution and the relationship amongst the variables are the same across individuals (Bauer & Curran, 2004). The assumption of a homogeneous population is not always met. A sample might consist of individuals that come from different populations with distinct distributions. For example, it is well-known that females suffer more often and more severely from Generalized Anxiety Disorder (GAD) than males. Brown, O’Leary, & Barlow (2001) summarized current prevalence estimates as a 2 to 1 (i.e., female to male) ratio. Suppose a GAD assessment is administered to a random group consisting of females and males. The mean score for females is expected to be higher than the mean score for males. This distributional difference reflects the difference in prevalence rates of the two groups. If the data are analyzed as a single homogeneous population, valuable information regarding each group and differences between the two groups might be lost. To avoid potentially suboptimal analysis, the data should be analyzed using methods that consider a heterogeneous population (i.e., two subpopulations) instead of one homogeneous population.

Subpopulation is a general term indicating a group within a heterogeneous population (Lubke & Muthén, 2005). Population heterogeneity might or might not be observed. Heterogeneity is observed if it is possible to define the subpopulations based on an observed variable (e.g., gender and ethnicity; Lubke & Muthén, 2005). In such a scenario, the different subpopulations are referred to as groups. If the heterogeneity is not observable (e.g., IQ and motivation) the subpopulations are referred to as latent classes or latent clusters¹. For example, suppose an achievement test is given to two distinct groups of students: females and males. The gender group to which each student belongs is directly observable. In such a scenario, the groups are defined and known

¹The terms latent classes and latent clusters are used interchangeably throughout this dissertation.

prior to the analysis and the groups can be analyzed using methods appropriate for multiple-group comparisons. Further suppose a mathematics achievement test is given to students without *a priori* knowledge of the ability of those students. Scores on the achievement test might reflect three types of students: high-ability students, average-ability students, and low-ability students. It is not possible to divide the sample into groups prior to carrying out the analysis because the achievement groups are not directly observable. The subpopulation membership (i.e., latent class) of each student must be inferred from the achievement test scores. In such a scenario, methods for multiple-group comparisons are not applicable. Instead, data should be analyzed using mixture modeling.

The factor mixture model (FMM) is a special type of mixture model that models factor structure heterogeneity of classes by combining latent class models with common factor models (Lubke & Muthén, 2007). Latent class analysis (LCA) is used to divide the data into different subpopulations that represent latent classes (Lubke & Muthén, 2007), while confirmatory factor analysis (CFA) is used to specify a factor structure within each of those subpopulations (see details below). FMM is believed to provide many advantages over other methods used to analyze heterogeneous data, namely: (a) provides a model based approach, which allows for model comparisons, (b) permits the specification of a large number of complex within-class submodels or factor structures (Lubke & Tueller, 2010), (c) accounts for the measurement error in the observed variables (Lubke & Tueller, 2010), and (d) accommodates several latent classes.

Statement of Problem

Despite the perceived advantages of FMM over other methods designed to deal with heterogeneous data, researchers face challenges when using the FMM method. First, FMM is highly parameterized and complex; many opportunities exist for misspecifications (Lubke & Tueller, 2010). In particular, misspecification of the within-

class factor structure (e.g., number of factors or variance-covariance matrices) is of great concern (Lubke & Tueller, 2010). Second, the distributional assumptions of FMM can be quite restrictive, which empirical data often do not meet. Violations of these assumptions (i.e., normally distributed variables, continuous observed variables, linear relationship between observed and latent variables, and uncorrelated errors and latent variables) can potentially be problematic (Bauer & Curran, 2003). For example, regarding normality, Bauer & Curran (2003) found that non-normal continuous measures and non-linear relationships amongst observed and latent variables can lead to estimation of spurious latent clusters. McLachlan & Peel (2000) noted that data from a single population exhibiting non-normal behavior could likely be incorrectly modeled as two populations, resulting in a decision that is not optimal for the data. Furthermore, Rindskopf (2003) commenting on the Bauer and Curran (2003) article showed examples of normal mixture distributions that erroneously appeared to be one normal distribution. Finally, another issue of FMM concerns the Expectation-Maximization (EM) algorithm, which is typically used to find maximum likelihood estimates of FMM parameters. The algorithm has been characterized as slow to converge and often highly dependent on initial conditions and starting values. In particular, the EM algorithm can have difficulty converging to a solution when latent classes exhibit a substantial amount of overlap, or when minimum group sizes are not *a priori* specified.

Although FMM is currently the gold standard approach for investigating factor structure heterogeneity, other methods might prove to be just as tenable in investigating the heterogeneity and overcoming weaknesses and limitations of FMM. Q-Factor Analysis (QFA), Marcoulides and Drezner Model (MDM), and McDonald Eigen Analysis (MEA) are three other approaches that can be used to analyze heterogeneous data although it is not yet clear whether they exhibit different strengths and weaknesses. QFA is a type of exploratory factor analysis (EFA) used to identify relationships amongst individuals and classify individuals on the basis of unobserved latent variables (Stephenson,

1935; Thompson, 1935). MDM is a newer and promising approach for detecting latent classes (Drezner, 1987; Marcoulides & Drezner, 2000). The method is based on spectral decomposition, whereby a similarity matrix consisting of weighted Euclidean distances between individuals is represented in its eigenvalues and eigenvectors. MEA is introduced and developed as part of this dissertation and is another competing method to detect latent classes. MEA is a spectral decomposition method, as well. The method decomposes a similarity matrix consisting of weighted Gower distances between individuals. The gaps between the eigenvalues are examined to determine the appropriate number of classes. All four methods are thoroughly discussed in Chapter 3 of this dissertation.

Significance of the Study

This study seeks to achieve several goals. First, to advance knowledge concerning latent class identification, as no prior literature exists on the use of MEA or MDM compared to currently popular latent class modeling methods available. The study examines to what extent MEA and MDM can be used for latent classification analysis, especially the extent to which either method can overcome any of the weaknesses of FMM under certain design conditions. Although some literature exists comparing QFA to other latent classification techniques, no published literature exists in which QFA is compared directly to the factor mixture model nor are there any simulation studies that evaluates QFA under various design conditions. Conducting this dissertation will also confirm whether QFA can provide advantages over the FMM under certain model conditions. Overall, the study seeks to determine how the FMM compares to less complex heterogeneous modeling approaches in the identification of latent classes. The determination will be made by examining the ability of each of the above mentioned

methods to correctly identify the true number of latent classes within heterogeneous data sets simulated under various design conditions that are known to be both ideal and problematic for FMM. Specifically, the following research questions are addressed:

1. Are QFA, MDM, and MEA viable alternatives to FMM?
2. Under what design conditions does each method underperform or outperform FMM?
3. Under what design conditions does each method perform best?
4. Under what design conditions does each method perform worst?

Chapter 2: Review of Heterogeneous Methods

A review of literature suggests there are a myriad of methods currently used to analyze heterogeneous data. However, the manner of carrying out the analysis and the type of research questions that can be answered can be quite different depending on the method. A review of the commonly used heterogeneous data analysis methods that are most relevant to this dissertation is given below. This review is an extension of the review given by (Lubke & Muthén, 2005).

Four criteria are used to categorize the methods: (a) whether the heterogeneity is observed or unobserved, (b) whether the method includes a hypothesized model, (c) whether the method includes latent variables, and (d) whether the type of observed variables is continuous, categorical, or both. The methods reviewed are: analysis of variance (ANOVA), multivariate analysis of variance (MANOVA), logistic regression (LR), discriminant analysis (DA), multi-group common factor analysis (MG-CFA; Jöreskog, 1971; Sörbom, 1974; Stevens, 2012), K -means clustering (MacQueen, 1967), spectral clustering, taxometry (Lubke & Tueller, 2010; Waller & Meehl, 1998), LCA, latent trait analysis (LTA), and latent profile analysis (LPA).

Heterogeneity Observed or Unobserved

As previously stated, heterogeneity can be observed or unobserved. Heterogeneity is observed if it is possible to define the subpopulations based on the outcome of an observed variable in the data, which serves as a grouping variable. Prior to the analysis, the grouping variable is defined. It is known to which group each individual belongs and the number of individuals in each group. Methods for unobserved heterogeneity take a different approach. Unobserved heterogeneity methods detect a pre-specified number of classes in data by examining the pattern of responses for the observed variables (Lubke

& Muthén, 2005). In contrast to methods for observed heterogeneity, it is not known to which group each individual belongs nor the number of individuals in each group prior to the analysis. Those parameters result from estimation of the model.

ANOVA, MANOVA, DA, LR, and MG-CFA are some of the well-known methods for dealing with observed heterogeneity. Such methods group or classify each individual depending on an observable grouping variable. Researchers use ANOVA when comparing groups on a single observed variable. ANOVA provides a statistical test to determine whether the means of several groups are all equal. MANOVA is an extension of ANOVA and is used when comparing group differences on several observable variables (Raykov & Marcoulides, 2008). Suppose a researcher is interested in the science achievement of students by ethnicity. ANOVA can be used to compare the science achievement means by ethnicity (e.g., science achievement of Asian students vs. science achievement of Black students vs. science achievement of White students). If a researcher were also interested in academic achievement in several areas (e.g., Science, Mathematics, English, etc.) MANOVA can be used to compare the achievement means by ethnicity of all the achievement tests simultaneously.

DA and LR are methods that can be used when a researcher is most concerned with identifying variables that strongly predict individual group membership. Both DA and LR can classify individuals into two or more known groups. For example, suppose a researcher is interested in knowing what observed variables (e.g., gender, ethnicity, achievement scores, motivation, etc.) contribute to predicting whether a student is classified as resilient or not. DA or LR can be called upon for such analysis. The difference in the two methods lie in the types of observed variables used in the methods, which is discussed in the following sections.

MG-CFA is an extension of the CFA method used for a single homogeneous population (discussed in detail below). The method allows for testing a factor structure theory across known groups. The basic idea in multi-group CFA is to fit factor

models in several groups simultaneously. The factor model fitted within a group is a linear regression model, which relates observed variables to a smaller number of latent variables called factors (Lubke et al., 2003). For example, suppose a researcher wished to test a hypothesized model on self-esteem to determine if the model is a good fit for boys and girls. The researcher could group the data by gender, test the hypothesized factor structure on each group, and then compare the results. In each of these methods discussed, the source of heterogeneity in the data are observed (i.e., grouping variable) and group membership is known *a priori*.

Methods that deal with unobserved heterogeneity include K -means clustering, spectral clustering, LCA, LPA, LTA, and taxometry. K -means clustering is a method that partitions individuals into clusters in a way such that each individual belongs to the cluster with the nearest mean. Spectral clustering involves taking the spectrum (i.e., eigenvalues and eigenvectors) of a similarity matrix between individuals and using the eigens (either the eigenvectors or eigenvalues depending on the method) to detect clusters within the data. LCA, LTA, and LPA are multivariate methods in which the responses on the observed variables are the result of an individual's position on the latent variable or the unobserved variable. LCA, LPA, and LTA are similar methods and are discussed below. Taxometry is used for determining whether relationships among observed variables reflect the existence of a latent taxon by examining the mean differences between two classes, small item covariances within-class, and large item variances between classes (Waller & Meehl, 1998). All of these methods answer the same type of research questions, but the mechanism of how the classes are identified differ. Suppose a researcher is interested in grouping students by their likelihood of passing a standardized test (e.g., very unlikely to pass, unlikely to pass, likely to pass, and very likely to pass) without knowing the students' abilities prior to the analysis. All of the methods for unobserved heterogeneity can deal with this type of research problem.

Hypothesized Model

The second criteria deals with a difference between the unobserved heterogeneity methods, whether a model is hypothesized prior to analysis or not. Model-based procedures imply that a statistical model is postulated for the population from which the data are obtained (Magidson & Vermunt, 2002). More precisely, model based procedures imply that a statistical model is hypothesized regarding the covariance structure for the population from which the data were sampled. Procedures that are not model-based classify individuals based on a somewhat arbitrary criteria that usually involves minimizing the within-class variation and maximizing the between-class variation. An advantage of using a method that involves statistical modeling is that the choice of the cluster criterion is less arbitrary and the approach includes rigorous statistical tests (Magidson & Vermunt, 2002). LCA, LPA, and LTA are model-based methods, while K -means, spectral clustering, and taxometry procedures are not model-based.

Latent Variables in Model

The third criteria deals with latent variables. As previously stated, latent variables are variables that are not directly observable. Instead the variables are inferred from other variables that are directly observed or directly measured. The main idea of latent variables is that the observable phenomena are influenced or caused by underlying and unobserved causes (Bollen, 2002). For example, mental disorders (e.g., depression) can often be thought of as latent variables or latent constructs. Mental disorders underlie the set of observable or measurable symptoms (e.g., fatigue, insomnia, lack of concentration, etc.).

ANOVA, MANOVA, DA, LR, K -means clustering, and spectral clustering do not include latent variables. All of the variables included in those models are observed. However, MG-CFA, LCA, LTA, LPA, and taxometry include latent variables. The latent variable can be independent or dependent, and either categorical or continuous. If the

latent variable is an independent variable, it is conceptualized as the underlying cause of the observed variables. If the latent variable is a dependent variable, it is typically dependent upon another latent variable in the model. In addition, continuous latent variables serve a different purpose compared to categorical latent variables. Instead of modeling the class membership of each individual, the continuous latent variables explain the common variation of the observed variables. MG-CFA, LTA, and taxometry have continuous latent variables, while LCA and LPA have categorical latent variables. CFA models can have independent or dependent latent variables, while the other models only have independent.

Type of Observed Variables

The final criteria deals with the types of observed variables in the model. Observed variables can be categorical, continuous, or both. ANOVA, MANOVA, DA, K -means clustering, spectral clustering, LPA, and taxometry are methods designed to use continuous observed variables. On the other hand, LCA and LTA are designed to use categorical observed variables. LR and MG-CFA are more flexible, in that the methods can handle continuous and categorical observed variables.

Table 1 summarizes each of the methods discussed. Chapter 3 provides a detailed discussion of the four methods (i.e., FMM, QFA, MDM, and MEA) used in this dissertation.

Table 1
Methods Used for Heterogeneous Populations

Classification Method	Heterogeneity	Model Based	Latent Variables	Observed Variables
ANOVA	Observed	No	No	Continuous
MANOVA	Observed	No	No	Continuous
Discriminant Analysis	Observed	No	No	Continuous
Logistic Regression	Observed	No	No	Both
Multi-group CFA	Observed	Yes	Continuous	Both
K -means Clustering	Unobserved	No	No	Continuous
Spectral Clustering	Unobserved	No	No	Both
Latent Class Analysis	Unobserved	Yes	Categorical	Categorical
Latent Profile Analysis	Unobserved	Yes	Categorical	Continuous
Latent Trait Analysis	Unobserved	Yes	Continuous	Continuous
Taxometry	Unobserved	Yes	Continuous	Continuous

Note: The latent variables column indicates whether latent variables are included in the model and if so the type of latent variable.

Chapter 3: Review of Methods Used in Study

This chapter provides an overview of the four latent class methods (i.e., FMM, QFA, MDM, and MEA) examined in this dissertation study. A detailed description of each method is given along with a discussion of how the number of latent classes in each was determined.

Q-Factor Analysis

English psychologist Charles Spearman (1904) is generally credited for the development of factor analysis. Spearman noticed that school children's scores on a wide variety of seemingly unrelated subjects were positively correlated. Spearman theorized that individual ability is a function of g , an underlying general mental ability. Factor analysis was used to explore this theory. Currently, factor analysis is used in a wide variety of substantive areas (e.g., educational, behavioral, social, and medical).

Factor analysis is a procedure for discovering latent (i.e., unobserved) traits, which underlie an individual's score on a set of observed variables. The goal of factor analysis is to explain those observed variables as linear combinations of factors using as few of the factors as possible (Raykov & Marcoulides, 2008). The factors are latent variables or underlying constructs that cause the values of the observed variables.

There are two types of factor analysis: CFA and exploratory factor analysis (EFA). Furthermore, there are several types of EFA and QFA is one of the types. Prior to presenting the details of QFA, the exploratory common factor model is given as described in Raykov & Marcoulides (2008). The model starts by considering a $p \times 1$ vector of observed variables $(y_1, y_2, \dots, y_p)'$, with an associated $p \times 1$ mean vector $(\mu_1, \mu_2, \dots, \mu_p)'$, and a $p \times p$ variance-covariance matrix of observed variables Σ . The factor analysis model expresses each mean-centered observed variable (i.e., $y_i - \mu_i$) as a linear combination of

underlying common factors. The linear combinations are defined as:

$$\begin{aligned}
 y_1 &= \lambda_{11}f_1 + \lambda_{12}f_2 + \dots + \lambda_{1m}f_m + \varepsilon_1 \\
 y_2 &= \lambda_{21}f_1 + \lambda_{22}f_2 + \dots + \lambda_{2m}f_m + \varepsilon_2 \\
 &\quad \vdots \\
 y_p &= \lambda_{p1}f_1 + \lambda_{p2}f_2 + \dots + \lambda_{pm}f_m + \varepsilon_p
 \end{aligned} \tag{1}$$

where f_1, f_2, \dots, f_m are independent and continuous common factors or latent variables, $\lambda_{11}, \lambda_{12}, \dots, \lambda_{ij}$ are factor loadings that serve as weights of the i^{th} observed variable on the j^{th} factor, and $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_p$ are residual error terms. Although unobserved, the factors f_j can be thought of as random variables that cause or influence the observed variables y_i . The factor loadings in the common factor model λ_{ij} convey the importance of the j^{th} common factor on the i^{th} observed variable. The common factor model presented in Figure 2 illustrates the relationship between f_j, y_i, λ_{ij} , and ε_i .

There appears to be a large degree of similarity between the linear regression model and the common factor model. In the linear regression model, observed variables are regressed on a response variable and the strength of the relationships between the observed variables and the response variable are captured in the regression weights. Conceptually, the factor analysis model can be thought of as a linear regression model where the observed variables are regressed on a factor and the strength of the relationships between the observed variables and the factor are captured in the factor loadings. However, there is an important distinction between the two models. The common factor model regresses the observed variables on an unobservable latent factor instead of another observed variable as in linear regression. This distinction makes it impossible to carry out a factor analysis via a regression method in practice.

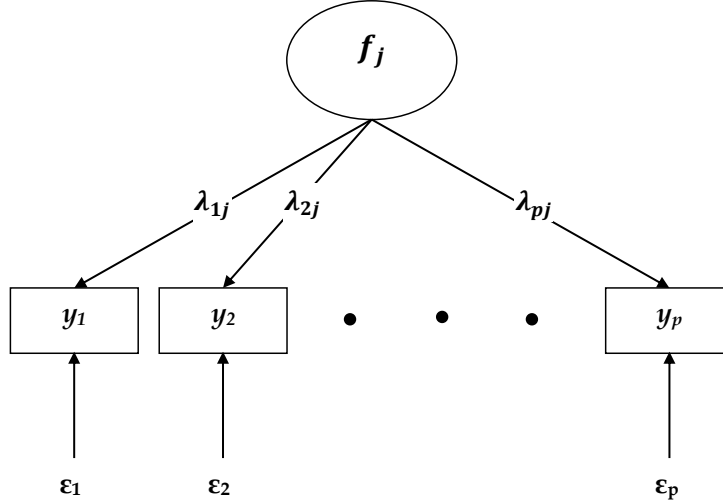


Figure 2. Common Factor Model.

In addition to Equation 1, a set of assumptions is needed to complete the definition of the common factor model. The assumptions define the expectancies and variance-covariance of the factors and residual errors. The assumptions are:

1. $E(f_j) = 0$. The expected values of the factors are 0.
2. $var(f_j) = 1$. The variances of the factors are 1.
3. $E(\varepsilon_i) = 0$. The expected values of the residual errors are 0.
4. $cov(\varepsilon_i, \varepsilon_k) = 0$ where $(i \neq k)$. The covariances of the residual errors are 0.
5. $cov(\varepsilon_i, f_j) = 0$. The covariances of factors and residual errors are 0.
6. $var(\varepsilon_i) = \psi_i$. The variances of residual errors are notated as ψ_i and referred to as specific variances.

The total variance of each observed variable $var(y_i)$ can be divided into two components: the common variance and the unique variance (Hatcher, 2007). The common variance (or the communality) h_i^2 refers to the portion of the total variance in y_i that is accounted for by the model (i.e., the factors). The remaining portion ψ_i is known as the specific or unique variance. The specific variance is not from the influence of f_j . Instead, ψ_i is the variance of the systematic or random error of y_i . The relationship

between the common and unique variances can be written as:

$$\begin{aligned}
\text{var}(y_i) = \sigma_{ii} &= \lambda_{i1}^2 + \lambda_{i2}^2 + \dots + \lambda_{im}^2 + \psi_i & (2) \\
&= (\lambda_{i1}^2 + \lambda_{i2}^2 + \dots + \lambda_{im}^2) + \psi_i \\
&= h_i^2 + \psi_i \\
&= \text{common variance} + \text{specific variance}
\end{aligned}$$

where $i = 1, 2, \dots, p$. Furthermore, the covariances can also be expressed in terms of λ_{ij} as:

$$\text{cov}(y_i, y_k) = \lambda_{i1}\lambda_{k1} + \dots + \lambda_{im}\lambda_{km}. \quad (3)$$

Employing the model assumptions and incorporating Equation 2 and Equation 3, the $p \times p$ variance-covariance matrix Σ can be expressed in compact matrix form as:

$$\Sigma = \Lambda\Lambda' + \Psi \quad (4)$$

where Λ is the $p \times m$ matrix of factor loadings and Ψ is the $p \times p$ matrix of specific variances. Equation 4 is an essential part of the factor analysis model. The emphasis in factor analysis is on modeling Σ across all pairs of observed variables. More specifically, the goal at hand is to reproduce the $\frac{p(p+1)}{2}$ non-redundant elements of the variance-covariance matrix Σ using the factor loadings λ_{ij} and specific variances ψ_i , in such a way that the parameters are in a simplified structure (Rencher & Christensen, 2012).

The factor analysis proceeds by estimating Λ and Ψ . This task is referred to as extracting the factors. Factor extraction is an attempt to disclose the factors or latent variables that explain the relationships amongst the set of observed variables (Raykov & Marcoulides, 2008). There are several techniques that can be used to carry out factor extraction. A few of the well-known techniques are: (a) principal factoring, which is the default in most statistical analysis software packages; (b) unweighted least

squares; (c) generalized least squares; (d) maximum likelihood; (e) alpha factoring; (f) image factoring; and (g) principal axis. If the assumption of multivariate normality is severely violated, researchers suggest using the principal factoring method (Fabrigar et al., 1999). Principal factoring has no distributional assumptions of the data. There are design conditions in this dissertation in which the data are non-normal. Therefore, the principal factoring extraction technique is used.

The extraction process starts by computing the empirical variance-covariance matrix \mathbf{S} from a given set of observed variables in an attempt to find estimators of $\mathbf{\Lambda}$ and $\mathbf{\Psi}$ that will approximate $\mathbf{\Sigma}$ (i.e., Equation 4). The approximation of $\mathbf{\Sigma}$ is given as:

$$\mathbf{S} \approx \hat{\mathbf{\Lambda}}\hat{\mathbf{\Lambda}}' + \hat{\mathbf{\Psi}} \quad (5)$$

where $\hat{\mathbf{\Lambda}}$ is the $p \times m$ matrix estimator of $\mathbf{\Lambda}$ and $\hat{\mathbf{\Psi}}$ is the $p \times p$ matrix estimator of $\mathbf{\Psi}$. The principal factoring technique ignores $\hat{\mathbf{\Psi}}$ and factors \mathbf{S} into $\hat{\mathbf{\Lambda}}\hat{\mathbf{\Lambda}}'$ using spectral decomposition. Spectral decomposition involves decomposing or factoring a variance-covariance or correlation matrix so that it is represented in terms of its eigenvalues and eigenvectors. The spectral decomposition of \mathbf{S} leads to:

$$\mathbf{S} = \mathbf{E}\mathbf{V}\mathbf{E}' \quad (6)$$

where \mathbf{E} is a $p \times p$ matrix of normalized eigenvectors $e_1'e_1, e_2'e_2, \dots, e_p'e_p = 1$ and \mathbf{V} is a $p \times p$ diagonal matrix of eigenvalues v_1, v_2, \dots, v_p . \mathbf{S} is a semi positive definite matrix, having eigenvalues ≥ 0 . Therefore, the matrix of eigenvalues \mathbf{V} can be factored as:

$$\mathbf{V} = \mathbf{V}^{1/2}\mathbf{V}^{1/2}. \quad (7)$$

Factoring matrix \mathbf{V} allows Equation 6 to be expressed as:

$$\mathbf{S} = \mathbf{E}\mathbf{V}^{1/2}\mathbf{V}^{1/2}\mathbf{E}' \quad (8)$$

$$= (\mathbf{E}\mathbf{V}^{1/2})(\mathbf{V}^{1/2}\mathbf{E}') \quad (9)$$

$$= \hat{\mathbf{\Lambda}}\hat{\mathbf{\Lambda}}'. \quad (10)$$

Equation 8 and Equation 9 can be written in vector notation as:

$$\mathbf{S} = v_1 e_1 e_1' + v_2 e_2 e_2' + \dots + v_p e_p e_p' \quad (11)$$

and then expressed as:

$$\mathbf{S} = \begin{bmatrix} \sqrt{v_1} e_1 & \sqrt{v_2} e_2 & \dots & \sqrt{v_p} e_p \end{bmatrix} \begin{bmatrix} \sqrt{v_1} e_1' \\ \sqrt{v_2} e_2' \\ \vdots \\ \sqrt{v_p} e_p' \end{bmatrix} \quad (12)$$

where v_i is the eigenvalue corresponding to the i^{th} variable and e_i is $p \times 1$ eigenvector corresponding to the i^{th} eigenvalue. Equations 10 and 12 present \mathbf{S} in exact form, meaning $\mathbf{\Sigma}$ can be recomposed exactly. However, the exact form is not particularly useful. The exact form of \mathbf{S} consists of as many factors as observed variables (i.e., $p = m$), which is contradictory to the goal of factor analysis in reducing the number of variables into a smaller number of factors. A representation of \mathbf{S} where $m < p$ is desirable. This can be accomplished by considering the contribution of each eigenvalue. The contribution of small eigenvalues to the spectral decomposition is small. In efforts of finding a reduced form of \mathbf{S} , the small eigenvalues of \mathbf{S} are discarded. Now, \mathbf{S} can be defined where $m < p$, and where m is now the number of retained eigenvalues. This

leads to:

$$\mathbf{S} = \left[\begin{array}{c} \sqrt{v_1}e_1 + \sqrt{v_2}e_2 + \dots + \sqrt{v_m}e_m \\ \sqrt{v_1}e_1 \\ \sqrt{v_2}e_2 \\ \vdots \\ \sqrt{v_m}e_m \end{array} \right]. \quad (13)$$

To complete the estimation of $\mathbf{\Sigma}$, the $\hat{\mathbf{\Psi}}$ matrix must be defined. The diagonal elements of $\hat{\mathbf{\Psi}}$ are the differences between the observed variable variances and the communalities $\hat{\lambda}_{ij}^2$. The relationship between the total, common, and specific variances presented in Equation 2 leads to:

$$\begin{aligned} \hat{\psi}_i &= \sigma_{ii} - \hat{\lambda}_{i1}^2 - \hat{\lambda}_{i2}^2 - \dots - \hat{\lambda}_{im}^2 \\ &= \sigma_{ii} - \sum_{j=1}^m \lambda_{ij}^2 \end{aligned} \quad (14)$$

where $\hat{\psi}_i = \text{diag}(\hat{\psi}_1, \hat{\psi}_2, \dots, \hat{\psi}_p)$ is the $p \times p$ diagonal of $\hat{\mathbf{\Psi}}$ and σ_{ii} is the $\text{var}(y_i)$. The diagonal elements are added to Equation 10, resulting in:

$$\mathbf{S} \approx \hat{\mathbf{\Lambda}}\hat{\mathbf{\Lambda}}' + \hat{\mathbf{\Psi}}. \quad (15)$$

The variances on the diagonal of \mathbf{S} are modeled exactly. However, the off-diagonal covariances are only approximate. Herein lies the challenge of factor analysis. Increasing the number of factors improves the approximation of the covariances. Yet, decreasing the number of factors can potentially compromise the goal of factor analysis, which is to reduce the number of variables.

The fit of the factor analysis model describes how well the model fits the observed data (Raykov & Marcoulides, 2008). The fit can be determined by comparing the left side and right side of Equation 15 and computing an error matrix \mathbf{E} . The er-

ror matrix represents the discrepancy between the model estimated variance-covariance matrix Σ and the actual variance-covariance matrix \mathbf{S} . The error matrix is computed as:

$$\mathbf{E} = \mathbf{S} - \Sigma \quad (16)$$

where the $m \times m$ matrix has zeros on the diagonal elements and non-zeros off diagonal. The sum of squared entries in \mathbf{E} is at most equal to the sum of squares of the discarded eigenvalues of \mathbf{S} . If the discarded eigenvalues were small, the elements of \mathbf{E} will be small and the fit good. If the discarded eigenvalues were large, the elements of \mathbf{E} will be large and the fit poor (Rencher & Christensen, 2012).

It is important to note that if a model has more than one factor, the factors can be correlated or uncorrelated. If the factors are correlated, the model is considered oblique. If the factors are not correlated, the model is considered orthogonal. The model definition as provided above is the typical definition of an EFA model, which assumes the factors are not correlated. However, such an assumption is usually not met in many empirical social and behavioral research studies (Raykov & Marcoulides, 2008). For this dissertation the factors do covary. To account for the covariance amongst the factors an oblique rotation was carried out when running the analysis. A rotation can be simply thought of as performing arithmetic to obtain a new set of factor loadings. Furthermore, if the factors are oblique, Σ is adjusted to incorporate the covariances:

$$\Sigma = \Lambda\Phi\Lambda' + \Psi \quad (17)$$

where Φ is the $p \times p$ matrix of factor variances-covariances. The adjustment is also considered in the approximation \mathbf{S} .

A decision must be made as to the number of latent classes. Determination of the number of meaningful classes is the most crucial problem confronting the researcher. Zwick & Velicer (1986) investigated the performance of 5 methods for determining the

number of factors or components to retain under systematically varied design conditions (i.e., sample size, number of variables, number of components, component saturation, equal or unequal numbers of variables for each component, and the presence or absence of unique and complex variables). The methods studied were: (a) Horn's parallel analysis, (b) Velicer's minimum average partial, (c) Cattell's scree test, (d) Bartlett's chi-square test, and (e) Kaiser's eigenvalue greater than 1 rule. Results from the study showed the performance of parallel analysis and minimum average partial methods generally performed best across all design conditions. Therefore, a decision was made to use parallel analysis as the method for determining the number of latent classes in the data.

Typically in the parallel analysis procedure, eigenvalues derived from the correlation matrix are compared with the eigenvalues extracted from random data sets. The random data sets parallel the actual data set with regard to dimensionality, the number of individuals and the number of variables (O'Connor, 2000). Eigenvalues from the data greater than the parallel analysis eigenvalues from the corresponding random data are retained and all others are discarded. In this dissertation, the conventional parallel analysis procedure was modified. Green et al. (2012) proposed a revised method of parallel analysis to deal with the conceptual flaws with the traditional parallel analysis method. First, instead of using completely random data sets, 500 data sets were generated incorporating the factor loadings from the population sampled. Furthermore, only the $k^{th} + 1$ eigenvalue is examined for comparison. For example, if data are generated assuming $k = 3$ factors, then the $k^{th} + 1 = 4^{th}$ eigenvalues from the generated data sets are tested against the $k^{th} + 1$ eigenvalues from the comparison data sets. Eigenvalues from the generated data greater than the comparison data were retained. The number of retained eigenvalues correspond to the number of latent classes.

As previously mentioned there are several types of EFA and QFA is one of them. The different types of EFA refer to what is serving as the rows in the data matrix and what is serving as the columns of the data matrix. The most common

type of factor analysis is known as R-factor analysis and is covered in most multivariate statistics courses. It is also the application of factor analysis that is most common in literature (Burger, 2004). This type of factor analysis, which examines the correlations of variables computed across individuals, attempts to identify the factors that lie behind the interrelationships amongst the variables. Although R-factor analysis is the most common type of factor analysis, some research questions cannot be answered using R-factor analysis (e.g., questions concerning types of people or types of individuals). In such cases, a much broader concept of EFA is needed and QFA might be called upon.

Stephenson (1935) and Thompson (1935) presented QFA as an inversion of conventional R-factor analysis by correlating individuals instead of variables, thus creating a Q-correlation matrix (i.e., rotating the data matrix so that the columns are now the rows and the rows are now the columns). QFA gives information about the similarities and differences of the individuals, instead of the variables. Therefore, factors resulting from a QFA represent clusters or classes of individuals.

The exploratory common factor model presented above is applicable to any type of EFA. The researcher would only need to supply the correct data structure. In this dissertation, data were generated in a conventional $n \times p$ data matrix, with n individuals in the rows of the data matrix and p variables in the columns of the data matrix. However, to analyze the data using QFA, the data were rotated into a $p \times n$ matrix, where the p variables are now the rows of the data matrix and n individuals the columns of the data matrix.

Factor Mixture Model

FMM is a method designed to deal with heterogeneous data, made up of two or more subpopulations that potentially have distinctive distributional parameters (McLachlan & Peel, 2000). As previously stated, FMM combines the common factor model and the latent class model (Lazarsfeld & Henry, 1968; Muthén, 1989). The

common factor model is used to *a priori* specify a factor structure within each subpopulation and the latent class model is used to categorize and divide the entire sample into different subpopulations (Lubke & Muthén, 2005). Different model definitions of FMM have been proposed in the literature (Arminger et al., 1999; Conor & Maas, 1998; Jedidi et al., 1997; Yung, 1997). This dissertation uses the model definition of FMM as proposed by Muthén & Shedden (1999). Although Muthén & Shedden (1999) included covariates as part of their model definition, covariates are not part of this dissertation and are not described.

FMM can be decomposed into two linear regression models (Lubke & Muthén, 2007): observed variables are regressed on within-class factors (factor analysis) and within-class factors are regressed on latent class variables (latent class analysis). As previously stated, there are two types of factor analysis: exploratory and confirmatory. The factor analysis presented in QFA was EFA and the goal was to discover how many factors could be used to explain the relationships amongst a given set of observed individuals (Raykov & Marcoulides, 2008). FMM uses CFA. Instead of discovering the number of factors, CFA is concerned with testing an *a priori* proposed factor structure of the relationship amongst the observed variables (Raykov & Marcoulides, 2008).

The CFA model presented in Equation 18 is similar to the EFA model presented in Equation 1. The model definition is expressed as:

$$y_i = \nu + \Lambda_y \eta_i + \varepsilon_i \tag{18}$$

where y_i is a $p \times 1$ vector of observed variables, ν is a $p \times 1$ vector of intercepts, Λ_y is a $p \times m$ matrix of factor loadings, η_i is a $m \times 1$ vector of continuous latent variables, and ε_i is a $p \times 1$ vector of residual errors for the i^{th} individual. The assumptions of the common factor model as discussed in the QFA hold for the CFA model in the FMM definition, as well. The notation used in the original common factor model presented in Equation 1 is slightly different than the model notation presented in Equation 18. The regression

intercepts ν , which represent the intercepts of within-class regression of the observed variable on the factor are added to the formula. Also, FMM notation conventionally denotes the common factor or latent variable as η_i instead of f_j . Figure 3 presents the adjustment of the notation.

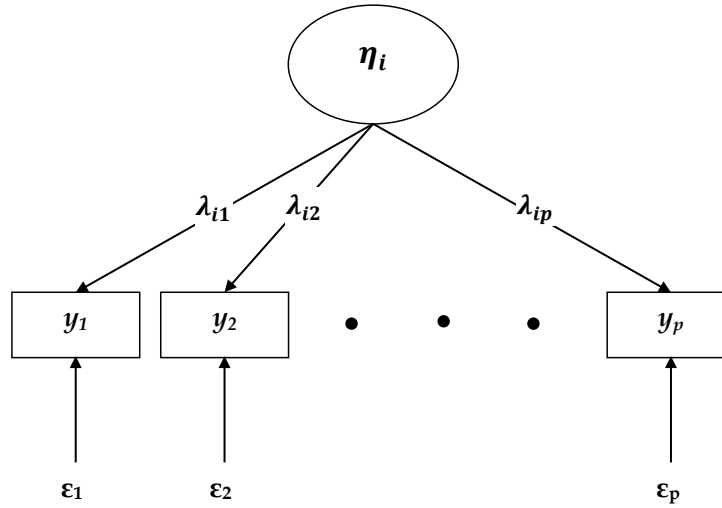


Figure 3. Common Factor Model.

CFA generally assumes that data come from a single homogeneous population and does not allow for comparison of different unknown groups (Raykov & Marcoulides, 2008). This shortcoming is dealt with by extending the common factor model to incorporate LCA in its framework. The basic idea underlying the latent class model is that some of the parameters of the hypothesized model differ across subpopulations. The subpopulations form the categories or classes of a categorical latent variable (Vermunt & Magidson, 2002). The latent class model organizes individuals into meaningful groups by using the observed variable responses (Nylund et al., 2007). The observed variables are related to each other only through the categorical latent variable C . The latent class model is presented in Figure 4.

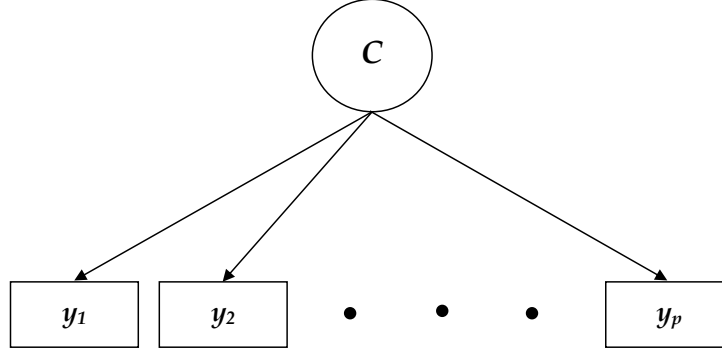


Figure 4. Latent Class Model.

The latent classes are modeled using a $k \times 1$ vector of latent categorical variables $(c_i = c_{i1}, c_{i2}, \dots, c_{ik})'$ that follow the multinomial distribution (Muthén & Shedden, 1999). The multinomial distribution is simply a generalization of the well-known binomial distribution. Binomial variables are discrete variables that count how often a particular event occurs in a fixed number of events. In such a scenario, there are only two categories, either an event occurs or it does not occur (e.g., a student in a classroom is classified as resilient or non-resilient). Instead of only two outcomes, the multinomial variable allows for any number of discrete outcomes (e.g., a student in a classroom is classified as low-resilient, medium-resilient, or high-resilient). In particular to FMM, the classification of an individual to a particular latent class can be thought of as an event. Each individual is categorized into one of the latent classes based on the individuals scores on the observed variables. The vector C is therefore defined as:

$$c_{ik} = \begin{cases} 1, & \text{if individual } i \text{ belongs to class } k \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

where c_{ik} is a $k \times 1$ vector of latent categorical variables and k is the number of latent classes. The common factor model is extended with the latent class model by regressing the within-class factors η_i on the class variable C . The extended common factor model

is now given as:

$$y_{ik} = \nu + \Lambda_{yk}\eta_{ik} + \varepsilon_{ik} \quad (20)$$

$$\eta_i = A c_i + \zeta_i \quad (21)$$

where A is a $m \times k$ matrix of intercepts for each latent class and ζ_i is an $m \times 1$ vector of the i^{th} individuals residual factor score not explained by the class membership. The subscript k is added to any parameter that is allowed to vary across classes and to the means and variance-covariance that may have class specific distributions (Lubke & Muthén, 2005). Figure 4 extends the common factor model by adding the latent classes. The result is presented in Figure 5. This is now considered the FMM. Although the model in Figure 5 illustrates a very simple within-class factor structure consisting of p observed variables loading on one within-class continuous factor, FMM permits a researcher to hypothesize a full range of simple or complex factor structures within-class.

Estimation of the model involves finding estimates for separate distributional parameters for all k classes, along with $k - 1$ mixing proportions that quantify the proportion of the data that is from each of the distributions or subpopulations (Gagné, 2006). Parameters of FMM are estimated by maximizing a likelihood function. For a given set of data, the method of maximum likelihood selects values of the model parameters that gives the observed data the greatest probability of occurring. The joint likelihood function for the k classes is given as (Gagné, 2004):

$$L = \prod_{i=1}^N \left(\sum_{j=1}^J \pi_j^* (2\pi)^{-\frac{p}{2}} |\hat{\Sigma}_j|^{-\frac{1}{2}} e^{(-.5)(y_i - \mu_j)' \hat{\Sigma}_j^{-1} (y_i - \mu_j)} \right) \quad (22)$$

where the model-implied mean vector for each class is:

$$\mu = \nu_j + \Lambda_j \eta_j \quad (23)$$

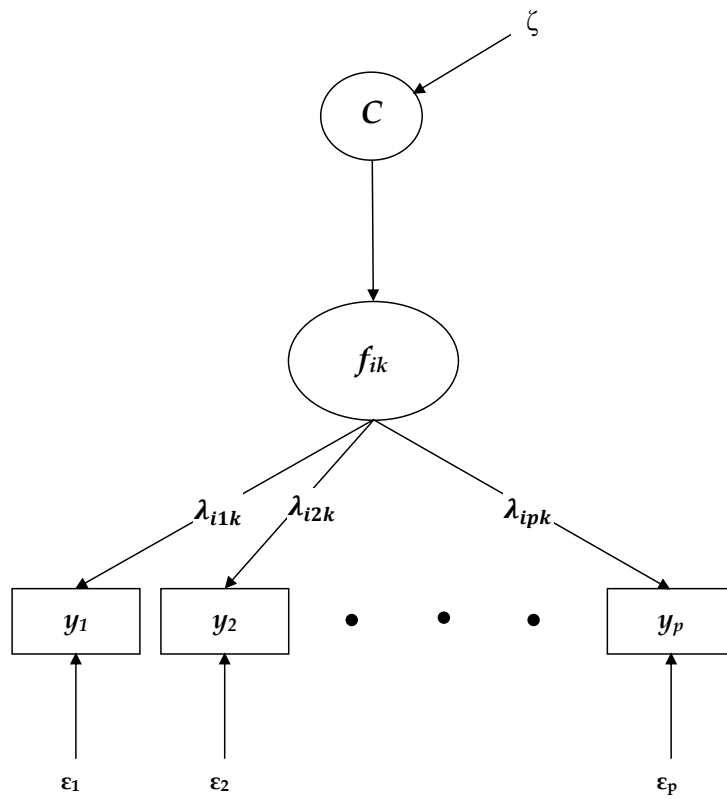


Figure 5. Factor Mixture Model.

and the model-implied variance-covariance for each class is defined as:

$$\hat{\Sigma}_j = \Lambda_j \Phi_j \Lambda_j' + \Psi_j \quad (24)$$

The estimated parameters are: (a) $m \times 1$ vector of factor means η , (b) $p \times m$ matrix of factor loadings Λ , (c) $m \times m$ matrix of factor variance-covariance Φ , (d) $p \times p$ variance-covariance matrix of residuals Ψ , (e) $p \times 1$ vector of intercepts ν , and (f) $k - 1$ mixing proportions π (π^* is the mathematical constant that is part of the normal distribution formula).

Despite the usefulness and flexibility of the FMM, one unresolved issue is how to accurately determine the number of latent classes in the study population (Nylund et al., 2007). Different fit criteria can be used to decide on the number of latent classes. Fit refers to the magnitude of the discrepancy between the data and the hypothesized model. Akaike's Information Criterion (AIC; Akaike, 1981), Bayesian Information Criterion (BIC; Schwarz, 1978), and the Adjusted BIC (aBIC; Sclove, 1987) are the most commonly used criteria for evaluating mixture models (Nylund et al., 2007; Lubke & Muthén, 2005).

The AIC, BIC, and aBIC are statistics based on the likelihood function (Gagné, 2006). The AIC is defined as:

$$AIC = -2 \log L + 2k \quad (25)$$

where k is the number of parameters in the model and L is the maximized value of the likelihood function for the estimated model Akaike (1987). Increasing the number of free parameters in a model improves the fit. Therefore, the AIC includes a penalty for this increase. However, this penalty is less than what is imposed in the BIC. The BIC is defined as:

$$BIC = -2 \log L + k \log(n) \quad (26)$$

where L is the maximum value of the likelihood function for the estimated model, k is the number of parameters to be estimated, and n is the number of individuals. The aBIC replaces the sample size n in the BIC equation with:

$$n^* = (n + 2)/24, \tag{27}$$

and fully expressed as:

$$aBIC = -2 \log L + k \log n^* \tag{28}$$

where again, L is the likelihood function for the estimated model, k is the number of parameters to be estimated, and n is the sample size. The AIC, BIC, and aBIC fit statistics are not standardized and are not interpreted for a given model; rather the focus is on relative size. For models estimated using the same data set, the model with the lowest fit statistic is the preferred model.

Researchers have reported inconsistent results on the performance of the fit indices. Some studies have reported the aBIC is the superior fit statistic (Tofighi & Enders, 2008; Yang, 2006); while other studies have reported the BIC is superior (Nylund et al., 2007; Magidson & Vermunt, 2004; Hagenaars & McCutcheon, 2002; Roeder & Wasserman, 1997). Previous studies have found the AIC to overestimate the correct number of latent classes in the FMM (Gilles & Gilda, 1996; Söromenho, 1994). In this dissertation the decision on which fit statistic to report was made after carrying out a pilot study. The AIC, BIC, and aBIC results were compared. The BIC performed better for one and two class models, while the aBIC performed better for three and four class models. Overall, the BIC performed best. Both fit statistics performed better than the AIC in all design conditions. Therefore, the BIC is reported in this dissertation.

Marcoulides and Drezner Model

MDM is a newer and promising approach for detecting latent classes or clusters in data (Marcoulides & Drezner, 2000). The general idea underlying the MDM approach lies in spectral clustering, whereby latent classes might be detected using eigenvalues and eigenvectors of a matrix derived by computing the distance between individuals. Spectral clustering has become quite popular over the past few years; such algorithms have been used successfully in various applications (Ng et al., 2001) and several spectral clustering algorithms have been published (Verma & Meila, 2003). The MDM is based on an algorithm originally developed by Drezner (1987). The original procedure was an approach to find a simple, yet effective solution for designing the layout of a facility. However, the generalization of the procedure allows it to be potentially useful whenever a researcher wishes to partition data into latent classes.

The method begins by assuming n individuals are located in an p -dimensional space, where p is the number of observed variables. A weight between each pair of individuals in the p -dimensional space is given as:

$$w_{ij} = \frac{1}{D_{ij}^P} \quad (29)$$

and

$$w_{ii} = 0, w_{ij} > 0, \text{ and } w_{ij} = w_{ji} \quad (30)$$

where P is the power parameter that varies according to the measurement design and equal to 1 in this dissertation, w_{ij} is a weight that represents the importance of the proximity between the i^{th} and j^{th} individuals, and D is the p -dimensional distance between the i^{th} and j^{th} individuals. The proximity can be thought of as a measure of similarity (although dissimilarity can be used, as well). The larger the weight the more similar the individuals on the observed variables and the smaller the weight the more dissimilar the individuals. Equation 29 shows the relationship between the weight and

the distance. If the distance between the i^{th} and j^{th} individuals is large, the denominator is large and effectively the weight is small. Likewise, if the p -dimensional distance between the i^{th} and j^{th} individuals is small, the denominator is small and effectively the weight is large. As an example consider Figure 6, which presents a scatter plot of three clusters. The distance between an individual in Cluster 1 and an individual in Cluster 3 is greater than the distance between two individuals within Cluster 3. The weights between the individuals within a cluster will be larger than the weights of individuals between clusters.

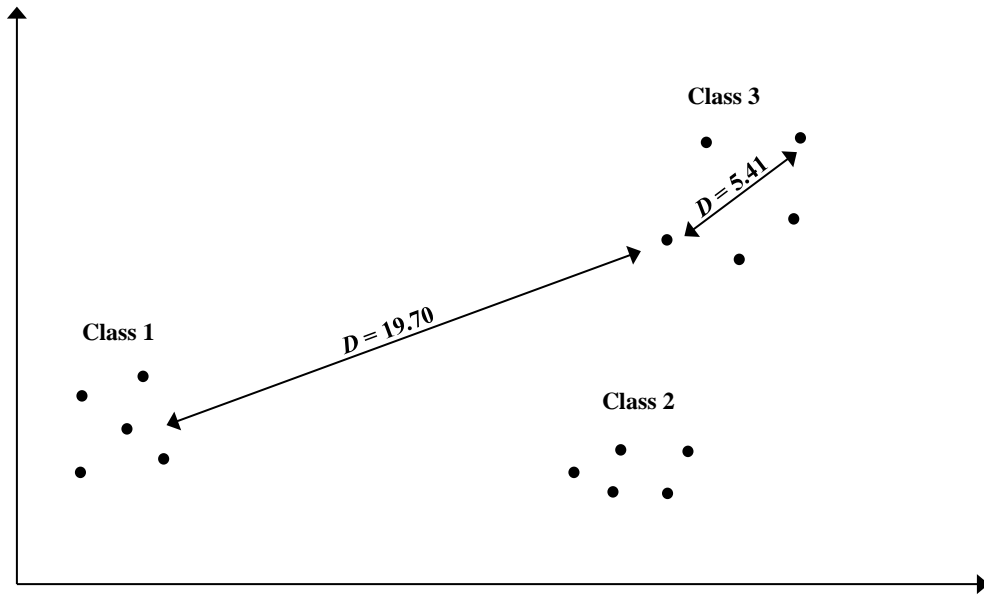


Figure 6. Distance Between and Within Clusters.

The model is formulated as a mathematical programming problem:

$$\min f(X) = \sum_{i,j=1}^n w_{ij} d_{ij}^2 \quad (31)$$

where X is a $p \times 1$ vector of observed scores for the i^{th} and j^{th} individuals on the latent variable and d_{ij}^2 is the squared Euclidean distance between the i^{th} and j^{th} individuals. The minimization of $f(X)$ is achieved when $d_{ij}^2 = 0$ for all pairs of individuals. However,

that solution is not of interest. To prevent the trivial solution, the denominator is added and the function is now expressed as:

$$\min f(X) = \frac{\sum_{i,j=1}^n w_{ij} d_{ij}^2}{\sum_{i,j=1}^n d_{ij}^2} \quad (32)$$

The optimal solution of $f(X)$ as given in Marcoulides & Drezner (2000) is found by creating a $n \times n$ matrix \mathbf{S} , where n is the number of individuals. The matrix \mathbf{S} is specified as:

$$s_{ij} = \begin{cases} \sum_j w_{ij} & \text{if } i = j \\ -w_{ij} & \text{if } i \neq j \end{cases} \quad (33)$$

and then taking the second smallest eigenvector associated with the second smallest eigenvalue of \mathbf{S} . The smallest eigenvalue of \mathbf{S} is $\lambda = 0$, with associated eigenvector of $e = (1, 1, 1, \dots, 1)'$; this solution is not of interest, therefore, the second smallest eigenvalue and associated eigenvector are used.

The method proceeds by standardizing the elements of the eigenvector. The elements are standardized using the following formula:

$$s_i = \frac{v_i - v_{min}}{v_{max} - v_{min}} \quad (34)$$

where s_i is the standardized element, v_i is the original element of the eigenvector, v_{min} is the smallest element in the eigenvector, and v_{max} is the largest element in the eigenvector. Standardizing the eigenvector retains the rank ordering of the original eigenvector. The standardized vector is then sorted in increasing order such that $s_{(i)} \leq s_{(i+1)}$. The standardization forces $s_{(1)} = 0$ and $s_{(n)} = 1$. Next, the $i = 1, \dots, n - 1$ gaps between two consecutive values of the standardized vector are computed using the formula:

$$\Delta_i = s_{(i+1)} - s_{(i)} \quad (35)$$

and then sorted in decreasing order:

$$\Delta_{(1)} \geq \Delta_{(2)} \geq \Delta_{(3)} \cdots \tag{36}$$

The largest term $\Delta_{(1)}$ divides the n individuals into two classes. The first k Δ 's divide the n individuals into $k + 1$ classes with the smallest gap between any two classes maximized.

The number of classes can be determined by examining the percentage contribution of each incremental Δ . The vector s_i is standardized; therefore, the accumulation of all gaps is exactly 1 or 100%. Accordingly, a researcher can select the first k Δ 's that sum to the percentage desired. For example, 90% is achieved when the cumulative value of the selected k Δ 's reaches .90. Although this approach is preferable, it is not realistic to examine each data set in a large simulation study. For this dissertation an alternative, yet similar approach is taken. The gaps are examined and a determination is made as to where the values of the gaps begin to level out. Different cut-off values give different results. A pilot study was carried out testing various gap size values ranging from 3% to 7%. The value of 5% gave the best results and was used for the entire study. Therefore, when the gap between two successive Δ 's is $< 5\%$ the other Δ 's were discarded.

McDonald Eigen Analysis

As with MDM, the MEA model is based on the spectral decomposition of a similarity matrix. However, there are 3 key differences between the two methods. In particular, the MEA: (a) uses the Gowers similarity measure, (b) weighs the distance by the range of the variable, and (c) takes a different strategy in deciding on the number of latent classes. MEA can easily incorporate a mix of data levels (i.e., nominal, ordinal, interval, and ratio). The inclusion of nominal and ordinal variables allows for covariates and *a priori* group membership related variables to be included. For example, suppose a researcher is testing a theory related to mathematics achievement. Researchers have

already established there exists group differences in mathematics achievement based on gender and ethnicity. Adding the *a priori* group membership information regarding ethnicity and gender into the clustering analysis could prove to be quite useful in detecting the correct number of latent classes. Also, the method accommodates missing data without listwise deletion. However, for this dissertation, covariates and missing data are not considered. Therefore, those two characteristics of the method are not assessed.

The procedure begins by assuming n individuals are located in a p -dimensional space, where p is the number of observed variables. An $n \times n$ matrix of similarities between each pair of individuals is computed. The matrix expresses the similarity of the i^{th} and j^{th} individuals on a given set of p variables. The similarity between the individuals are determined by the Coefficient of Similarity (Gower, 1971) and is defined as:

$$s_{ij} = \frac{\sum_{k=1}^p \delta_{ijk} d_{ijk}}{\sum_{k=1}^p \delta_{ijk}} \quad (37)$$

where $\delta_{ijk} = 1$ for ordinal, interval and ratio variables. However, if the k^{th} variable is nominal, then $\delta_{ijk} = 1$ if the i^{th} or j^{th} individuals are present. If both the i^{th} and j^{th} individuals are missing, then $\delta_{ijk} = 0$. Furthermore, for nominal variables $d_{ijk} = 1$ if $i = j$, otherwise $d_{ijk} = 0$. For ordinal, interval, or ratio variables:

$$d_{ijk} = 1 - \frac{|x_{ik} - x_{jk}|}{r_k} \quad (38)$$

where r_k is the range of the k^{th} variable. For interval variables d_{ijk} ranges between 0 and 1, inclusive. A value of 0 indicates the i^{th} and j^{th} individuals are extremely different and a value of 1 indicates the i^{th} and j^{th} individuals are the same.

The method proceeds by computing the $n \times n$ degree matrix \mathbf{G} . The degree matrix is defined as a diagonal matrix with the diagonal elements consisting of the sum of the corresponding column of the similarity matrix \mathbf{S} :

$$\mathbf{G}_{ij} = \begin{cases} \sum_j s_{ij} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (39)$$

The degree of \mathbf{S} provides information on an individual's connection or relationship with other individuals. For example, if an individual has a degree of 0, then the individual is not similar to any other individuals (i.e., the similarity between the individual and any other individual is 0). Next, a Laplacian matrix \mathbf{L} is computed. \mathbf{L} contains information on the degree and the similarity of each individual and is computed as:

$$\mathbf{L} = \mathbf{G} - \mathbf{S}. \quad (40)$$

\mathbf{L} has many properties and is the main tool used for spectral clustering. Spectral graph theory is a field that is dedicated to the study of Laplacian matrices. In particular to this dissertation, \mathbf{L} can be used to find the number of latent classes in the data. Matrix perturbation theory indicates the stability of the eigenvectors is determined by the gap of the eigenvalues (i.e., eigengaps) of a Laplacian matrix. When an eigenvector's eigenvalue is near other eigenvalues, the subspace spanned by the eigenvectors corresponding to a cluster of eigenvalues may be insensitive to perturbations in the elements of the matrix (Li & Liu, 2007). Therefore, the stability of the latent classes can be found by examining the eigengaps of \mathbf{L} .

The eigenvalues of \mathbf{L} are computed and sorted in ascending order: $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_n$. The eigengaps are then computed by finding the difference between two successive eigenvalues:

$$EG(\mathbf{L}) = \lambda_{i+1} - \lambda_i. \quad (41)$$

The ordered eigenvalues are examined to find where the largest gap exists. The corresponding eigengap represents the number of latent classes. For example, suppose the following ordered eigengaps were computed: 40, 43, 50, 110, and 125. The largest gap is between the 3rd and 4th eigenvalues. Therefore, there are 3 latent classes in the data.

Chapter 4: Method

The major aim of this dissertation was to compare the performances of FMM, QFA, MDM, and MEA under various controlled design conditions. The best way to achieve this aim was to use Monte Carlo simulations as an investigation tool. Monte Carlo involves the use of random sampling techniques and computer simulations to approximate solutions to mathematical problems especially in terms of a range of values each of which has a computed probability of being the solution (Fan et al., 2002). In this dissertation, the Monte Carlo technique was used to generate sample data with known population parameters. The ability of each method in detecting the correct number of classes under different design conditions was then examined.

This chapter describes the methodology of the Monte Carlo simulation study. In particular, the following topics are discussed: manipulated design conditions, invariant design conditions, data generation procedure, outcome measures used in evaluating the performance of the methods, and software packages used to generate data and results for each method.

Manipulated Design Conditions

Seven design variables were manipulated: method, sample size, number of latent classes, mixing proportion, class separation, distributional assumption, and model complexity. A description of each design variable (aside from method, which was discussed thoroughly in Chapter 3) is given below.

Sample size. Three levels of sample size (N) were studied: 180, 540, and 1080 individuals. The sample sizes stated were for the whole samples (i.e., not the sample sizes for the classes). A review of the literature of other Monte Carlo simulation studies in mixture modeling showed these sample sizes are similar to what other researchers used as small, medium, and large samples (Tofighi & Enders, 2008).

Number of latent classes. Four levels of number of latent classes (k) were assessed: 1, 2, 3, and 4 classes. Simulation studies using $k = 1$ and $k = 2$ classes have been repeated extensively in mixture modeling and can serve as a link to prior literature. Few empirical studies in literature have studied populations with $k \geq 3$.

Mixing proportion. Two levels of mixing proportion (π) were used: balanced (i.e., π_i 's are equal) and imbalanced (i.e., $\pi_1 = .70$). The mixing proportion is the probability of a specific class occurring in the population. Furthermore, the mixing proportion dictates how the whole sample size was divided into class sizes. At the π_1 balanced level, each class had the same number of individuals equal to N/k . At the π_2 imbalanced level, the first class had $.70 \times N$ individuals and the other classes were equal and had $(.30 \times N)/(k - 1)$ individuals.

Class separation. Four levels of class separation ($\Delta\eta$) were tested: 1.5, 2.0, 2.5, and 3.0. The separation of classes was measured by Mahalanobis Distance (MD), a multivariate distance statistic computed as:

$$MD = \sqrt{(d' \mathbf{S}^{-1} d)} \quad (42)$$

where d is a vector of univariate standardized mean differences for the classes of interest and \mathbf{S} is the variance-covariance matrix of the within-class latent factors. Incorporating prior study information, the four levels of separation between classes were established as detailed above.

Distributional characteristic. Two levels of distributional characteristics were examined: normally distributed factor scores (i.e., normal data) and non-normally distributed factor scores (i.e., non-normal data). The normal design condition had a skewness and kurtosis both equal to 0.0. The non-normal design condition had a skewness equal to 1.0 and kurtosis equal to 3.0. These values are large enough to cause a

significant deviation from the normal distribution. Yet, the values are small enough to fall within the range of skewness and kurtosis encountered by behavioral and social sciences researchers (Micceri, 1989).

Model complexity. Four data generating models with increasing complexity were tested: a one-factor model with 5 indicators (referred to as Model 1), a two-factor model with 10 indicators (referred to as Model 2), a two-factor model with 10 indicators and 4 cross loadings (referred to as Model 2c), and a three-factor model with 15 indicators (referred to as Model 3). The models are presented in Figure 7 through Figure 10. Model 1, Model 2, and Model 2c are similar to other models used in mixture modeling simulation studies. Model 3 has not been studied and was added to determine the performance of the methods with the additional complexity provided (i.e., another latent factor and more observed variables).

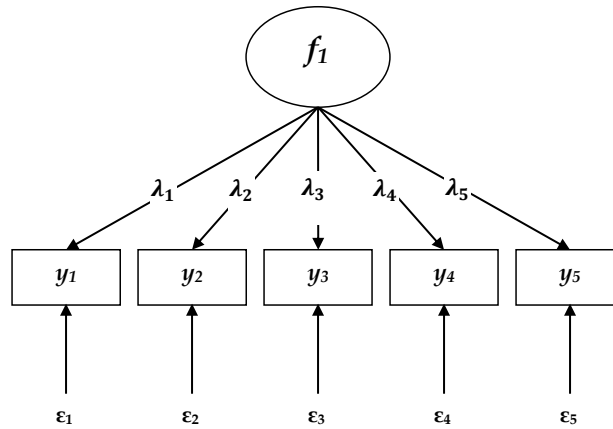


Figure 7. One-Factor Mixture Model.

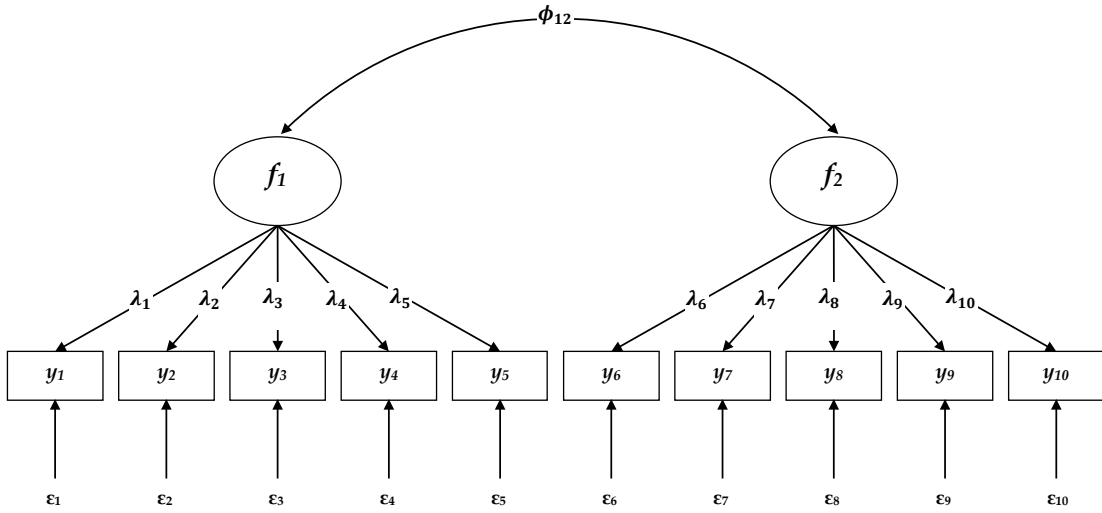


Figure 8. Two-Factor Mixture Model.

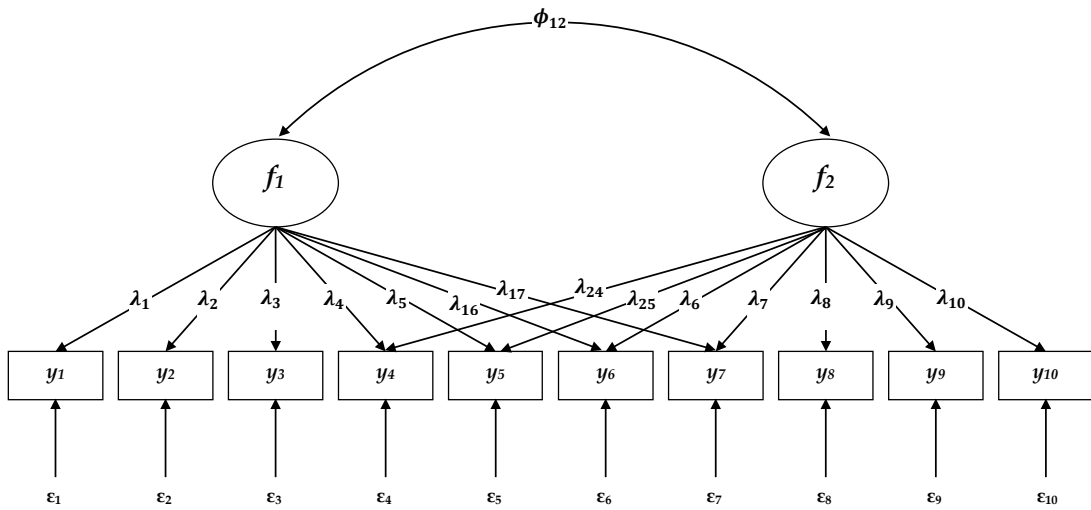


Figure 9. Two-Factor Mixture Model with Cross-Loadings.

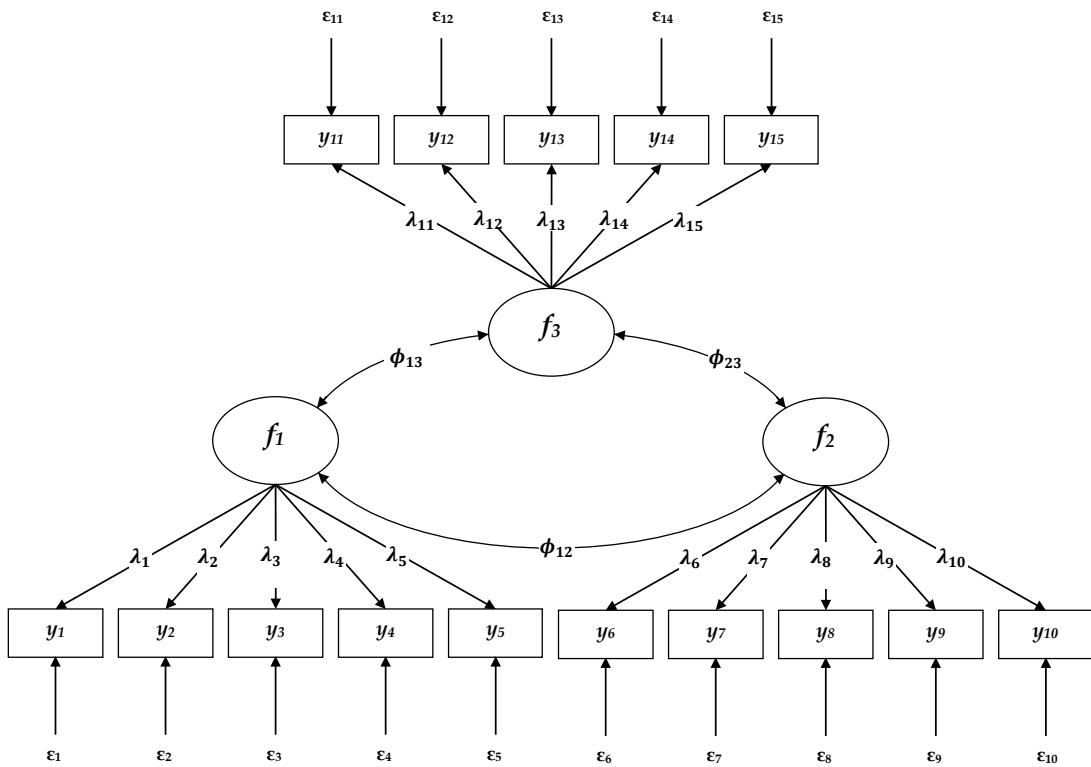


Figure 10. Three-Factor Mixture Model.

Invariant Design Conditions

Four design conditions were not manipulated. First, the factor structure was invariant across classes within a specific model. Second, factor loadings λ were equal to .80 when the observed variable was not cross-loaded and equal to .40 when the variable was cross-loaded. Third, residual errors of the observed variables ε were distributed as $N(\mu = 0.00, \sigma^2 = 0.36)$ if the observed variable was not cross-loaded. If the observed variable was cross-loaded, then the residual errors were distributed as $N(\mu = 0.0, \sigma^2 = 0.19)$. The residual errors were not correlated. Finally, all factor variances ϕ_{jj} were equal to 1.00 and factor covariances ϕ_{jk} were equal to 0.60. Setting the factor variances equal to 1.00 made the variance-covariance matrices equal to correlation matrices.

As previously stated, the variance-covariance matrix of the observed variables is specified as:

$$\Sigma = \Lambda \Phi \Lambda' + \Psi. \quad (43)$$

Implied variance-covariance matrices for each model were computed using the factor loading matrix, the factor variance-covariance matrix, and the specific variance matrix:

$$\Sigma_1 = \begin{bmatrix} 1.00 & & & & \\ .64 & 1.00 & & & \\ .64 & .64 & 1.00 & & \\ .64 & .64 & .64 & 1.00 & \\ .64 & .64 & .64 & .64 & 1.00 \end{bmatrix}$$

$$\Lambda_1 = \begin{bmatrix} .80 \\ .80 \\ .80 \\ .80 \\ .80 \end{bmatrix} \quad \Phi_1 = [1.00] \quad \Psi_1 = \begin{bmatrix} .36 & & & & \\ .00 & .36 & & & \\ .00 & .00 & .36 & & \\ .00 & .00 & .00 & .36 & \\ .00 & .00 & .00 & .00 & .36 \end{bmatrix}$$

Data Generation

Five major steps were taken to generate the mixture data: (1) factor scores were generated, (2) specified inter-correlation was imposed on the factor scores, (3) factor scores were transformed to non-normal, if applicable, (4) class separation was created, and finally (5) observed data were generated. Each of the steps is described below. The steps of the data generation process are discussed using one of the design conditions as an example. Model 2 with $k = 2$, $N = 1080$ ($N = 540$ per class), $\Delta\eta = 3.0$, and non-normal data is used to aid in explaining the details of the process.

The process started by generating data for the within-class factor scores η_1 and η_2 of the first class. Specific to the example, two vectors of $N = 540$ standard normal factor scores were generated. The factor scores were distributed as $N(\mu = 0.0, \sigma^2 = 1.0)$ with skewness and kurtosis both equal to 0.0. Also, the factor scores were generated to be independent (i.e. $\rho = .0$). Table 2 presents the descriptive statistics and the correlation matrix of the factor scores from the sample generated.

Table 2
Factor Scores

	<i>N</i>	<i>M</i>	<i>SD</i>	<i>Skew</i>	<i>Kurtosis</i>
η_1	540	0.00	1.00	0.00	0.01
η_2	540	0.00	1.00	0.00	0.00

$$r = \begin{bmatrix} & \eta_1 & \eta_2 \\ \eta_1 & 1.00 & \\ \eta_2 & .00 & 1.00 \end{bmatrix}$$

The next step involved transforming the factor scores to have the appropriate correlation. If the design condition calls for normal data, then the correlation matrix previously specified for each model would be appropriate. However, if the design condition

calls for non-normal data, then an adjustment is required. The procedure to generate correlated and non-normal data must use an intermediate correlation matrix. Interaction between correlations and non-normality causes the sample correlations to deviate from the intended population correlations (Fan et al., 2002). Using an intermediate correlation resolves this issue. Vale & Maurelli (1983) presented a procedure for computing the intermediate correlation matrix by demonstrating the following relationship:

$$r_{12} = \rho(b_1b_2 + 3b_1d_2 + 9d_1d_2) + \rho^2(2c_1c_2) + \rho^3(6d_1d_2) \quad (44)$$

where r_{12} is the intended correlation between two variables and ρ is the intermediate correlation between the same two variables. The coefficients a , b , c , and d are needed to specify the level of skewness and kurtosis for the non-normal data. The coefficients are found in the Power Method Weights table (Fleishman, 1978). The following coefficient values were needed to transform each of the two variables to have a positive skewness of 1.0 and kurtosis of 3.0: $a = -c = -0.128396714$, $b = 0.832216343$, and $d = 0.048032053$. All elements of the formula were known except ρ . An iterative approach (i.e., Newton-Raphson) was implemented to solve for ρ . The intermediate correlations were then assembled into a correlation matrix. For each model, the intermediate correlations were computed as:

$$\rho_{Model1} = [1.00]$$

$$\rho_{Model2=Model3} = \begin{bmatrix} 1.00 & \\ .61 & 1.00 \end{bmatrix}$$

$$\rho_{Model4} = \begin{bmatrix} 1.00 & & \\ .61 & 1.00 & \\ .61 & .61 & 1.00 \end{bmatrix}$$

Kaiser & Dickman (1962) presented a matrix decomposition procedure that imposes a specified correlation matrix on a set of random normal variables to create multivariate normal variables. The transformation formula is given as:

$$\hat{\mathbf{Z}}_{(p \times N)} = \mathbf{F}_{(p \times p)}(\hat{\mathbf{Y}})_{(p \times N)} \quad (45)$$

where p is the number of variables, N is the sample size, $\hat{\mathbf{Y}}$ is a $p \times N$ data matrix of uncorrelated standard normal variables, \mathbf{F} is the $p \times p$ matrix of factor pattern coefficients that results from applying principal component analysis to the correlation matrix (or intermediate correlation matrix), and $\hat{\mathbf{Z}}$ is the resulting $p \times N$ data matrix (Kaiser & Dickman, 1962). Specific to this dissertation, $\hat{\mathbf{Z}}$ is the matrix of correlated factor scores, N is the class sample size, and p is the number of within-class factors.

The process continues by transforming the correlated factor scores from normally distributed to non-normally distributed, if the design condition calls for non-normal data. Fleishman's Power Transformation (Fleishman, 1978) uses a polynomial transformation that converts a normally distributed variable to a non-normally distributed variable with a specified level of skewness and kurtosis. The polynomial is given as:

$$Y = a + bZ + cZ^2 + dZ^3 \quad (46)$$

where the coefficients a , b , c , and d are the same coefficients used in computing the intermediate correlation matrix, Z is the standard normal factor score, and Y is the resulting transformed non-normal variable with the specified skewness and kurtosis. Table 3 presents descriptive statistics and the correlation matrix of the factor scores after imposing the appropriate correlation, skewness, and kurtosis.

Table 3
Factor Scores After Correlation and Non-Normality Applied

	<i>N</i>	<i>M</i>	<i>SD</i>	<i>Skew</i>	<i>Kurtosis</i>
η_1	540	0.00	1.00	1.00	3.02
η_2	540	0.00	1.00	1.01	3.00

$$r = \begin{bmatrix} & \eta_1 & \eta_2 \\ \eta_1 & 1.00 & \\ \eta_2 & .60 & 1.00 \end{bmatrix}$$

Now the sample of factor scores have the appropriate level of non-normality (skewness=1.0 and kurtosis=3.0) and with the correct factor correlations ($r_{11} = r_{22} = 1.0$ and $r_{12} = .60$). The entire process outlined above was repeated for each class that the design condition called for. In particular to the example, the process was repeated to generate another sample for the second class.

The next step was to separate the classes at the specified $\Delta\eta$ level. The factor scores in the appropriate classes were linearly transformed to create the appropriate class separation by using the formula:

$$\eta'_i = \eta_i * SD + M \tag{47}$$

where η'_i is the transformed factor score, η_i is the original factor score, M is the value needed to transform the mean of the vector of factor scores and SD is the value needed to transform the standard deviation of the vector of factor scores. The value of M needed to produce the correct $\Delta\eta$ was found by trial and error using Equation 42. Various values of d were tested while holding constant \mathbf{S} . Therefore, the class separation was based entirely upon the factor means. In the example, the first class of $N = 540$ factor scores were left unchanged. The factor scores of the second class of $N = 540$ were linearly

transformed. It was found that a value of $d = 1.55$ was needed to achieve $MD = 3.0$. After the data for the second class were generated, $SD = 1$ was multiplied to each factor score and $M = 1.55$ was added to each factor score. If the design condition called for $k = 3$ or $k = 4$, the process would be repeated. Table 4 presents the correlated factor scores after the two classes of non-normal data were created. Also, the table presents the $\Delta\eta$ between the two classes.

Table 4
Factor Scores After Correlation, Non-Normality, and Class Separation Applied

		N	M	SD	$Skew$	$Kurtosis$
Class 1	η_1	540	0.00	1.00	1.00	3.02
	η_2	540	0.00	1.00	1.00	3.02
Class 2	η_1	540	1.55	1.00	1.01	3.00
	η_2	540	1.55	1.00	1.01	3.00

$$\Delta\eta = 3.00$$

The final step was to generate the observed data using the formula:

$$y_i = \Lambda_y \eta_i + \varepsilon_i \quad (48)$$

where y_i is the observed data value computed by multiply each factor score η_i with the appropriate factor loading Λ , and then adding the residual error ε_i . The factor loadings and residual errors were specified in the Invariant Design Conditions section and the factor scores were generated in the previous steps. In particular to the example, data for 10 indicator variables were generated. Table 5 presents the descriptive statistics after computing data for the observed variables using Equation 48.

Table 5
Observed Data Variables Generated Using Factor Scores

		<i>N</i>	<i>M</i>	<i>SD</i>	<i>Skew</i>	<i>Kurtosis</i>
Class 1	Y_1	540	0.00	1.00	1.00	2.98
	Y_2	540	0.00	1.00	1.00	3.00
	Y_3	540	0.00	1.00	0.99	3.00
	Y_4	540	0.00	1.00	1.00	3.00
	Y_5	540	0.00	1.00	1.00	3.01
	Y_6	540	-0.01	1.00	1.12	3.00
	Y_7	540	0.00	1.00	1.00	3.00
	Y_8	540	0.00	1.00	1.00	3.10
	Y_9	540	0.00	1.00	1.00	3.00
	Y_{10}	540	0.01	1.00	1.00	3.00
Class 2	Y_1	540	1.24	1.00	1.00	3.01
	Y_2	540	1.24	1.00	1.00	3.00
	Y_3	540	1.24	1.00	0.97	3.00
	Y_4	540	1.23	1.00	1.00	3.00
	Y_5	540	1.23	1.00	1.00	3.01
	Y_6	540	1.24	1.00	1.03	3.00
	Y_7	540	1.26	1.00	1.00	3.00
	Y_8	540	1.24	1.01	1.00	3.00
	Y_9	540	1.24	1.00	1.00	3.02
	Y_{10}	540	1.24	1.00	1.00	3.00

It is important to note that the design conditions generated at the latent factor level (i.e., non-normality, number of classes, class size, class separation, and mixing proportion) manifested to the observed variable level, as well.

All of the manipulated design variables were completely crossed, resulting in 3,072 (i.e., 4 methods \times 3 sample sizes \times 4 number of classes \times 2 mixing proportions \times 4 class separations \times 2 distributional characteristics \times 4 models) distinct combinations. Although there is no generally agreed upon recommendation for the number of samples in a Monte Carlo simulation study, 500 data sets per condition is consistent with other

research studies found in literature. For each design condition, 500 data sets were generated, giving a total of 384,000 data sets generated for this dissertation (each method used the same data sets).

Outcome Measures

Two outcome measures were used to evaluate how well each method performed: the performance rate and Cohen’s h effect size. The performance rate is simply defined as the proportion of replications for which the true number of latent classes was identified. It serves as an evaluation of how well each method performed under each of the design conditions. The proportion was computed as:

$$\text{Performance Rate} = p = \frac{X_{success}}{X_{total}} \quad (49)$$

where $X_{success}$ is the number of replications for which the method detected the correct number of latent classes and X_{total} is the total number of replications (i.e., 500). This approach is similar to other simulation studies aimed at detecting the correct number of classes (Tofighi & Enders, 2008; Nylund et al., 2007).

Furthermore, Cohen’s h effect size (Cohen, 1988) was reported for each manipulated design condition. Cohen’s h provides information on detecting the difference between two proportions. The proportions were transformed using the following formula:

$$\phi = 2(\arcsin\sqrt{p}) \quad (50)$$

where \arcsin is the inverse trigonometric function of the \sin function. The transformations are needed to address the issue of the dependency between p and its variance $p(1 - p)$. The effect size is computed as:

$$h = |\phi_1 - \phi_2| \quad (51)$$

where ϕ_1 corresponds to the first group in the comparison and ϕ_2 corresponds to the second group in the comparison. In particular to this dissertation, the effect sizes were computed for each manipulated design variable, using the smallest and largest values of p .

The traditional operational definition provided by Cohen 1988 was used to qualify the magnitude of h and p . The following criteria were used:

$$\begin{aligned} &\cong 0.00 && \text{Trivial, no practical importance} \\ &\cong 0.20 && \text{Small, negligible practical importance} \\ &\cong 0.50 && \text{Moderate, moderate practical importance} \\ &\cong 0.80 && \text{Large, crucial practical importance} \end{aligned} \tag{52}$$

Software Packages

This dissertation used three software packages: *Mplus* (v6.12), R (v2.15.1), and SAS (v9.2). The coding written to simulate the data and summarize the results are included in the Appendix. The datasets were generated using SAS and SAS IML and following the method outlined by Fan (2002). FMM results were generated by calling *Mplus* through the *MplusAutomation* (v0.5-3) package in R. QFA results were computed using SAS Base. MDM and MEA methods were coded in SAS Base and SAS IML specifically for this dissertation.

Chapter 5: Results

This chapter presents the results of the simulation study as detailed in the prior chapter. The results are organized in 40 tables (i.e., Table 6 through Table 45). First, the results are presented by method, model, and distributional characteristic². Next, a comparison of the methods by design condition is presented. Finally, Cohen's h effect sizes are presented for each design variable.

Factor Mixture Model Results

Table 6 shows the proportion of replications for which FMM correctly identified the true number of latent classes in Model 1 with normal data. The performance rate had a positive relationship with the sample size. As N increased p generally increased, as well. The average p was .812 when $N = 180$, and increased by 15.8% to .940 when $N = 1080$. The number of classes had a negative relationship with the performance. As k increased p decreased. The average p was .999 when $k = 1$, but decreased by over 34% to .657 when $k = 4$. Although the mixing proportion had a differential effect on the performance rate the effect was weak. The difference between the average performance rate at π_1 and the average performance rate at π_2 was only .045. When π did affect the performance rate, FMM performed better at π_1 . Finally, the distance between factor means and the average performance rate had a positive relationship. The average p increased by about 25.0% from .784 when $\Delta\eta = 1.5$ to .976 when $\Delta\eta = 3.0$.

²Although $\Delta\eta$ and π are not applicable when $k = 1$, those results were added to each block in the tables so that $k = 1$ can be easily compared to $k = 2$, $k = 3$, and $k = 4$ in all design conditions.

Table 6
FMM Results - Model 1 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.998	1.000	1.000	.998	1.000	1.000
	2	.996	.998	1.000	.986	.998	.998
	3	.538	.782	.890	.522	.644	.802
	4	.300	.500	.724	.268	.356	.512
2.0	1	.998	1.000	1.000	.998	1.000	1.000
	2	.998	1.000	1.000	.990	.998	1.000
	3	.770	.906	1.000	.656	.796	.970
	4	.490	.550	.768	.332	.540	.564
2.5	1	.998	1.000	1.000	.998	1.000	1.000
	2	.998	1.000	1.000	.992	1.000	1.000
	3	.928	.994	1.000	.680	.968	.998
	4	.534	.836	.998	.526	.542	.882
3.0	1	.998	1.000	1.000	.998	1.000	1.000
	2	1.000	1.000	1.000	.998	1.000	1.000
	3	.988	.994	1.000	.924	.994	1.000
	4	.850	.992	1.000	.738	.976	.998

Table 7 shows the proportion of replications for which FMM correctly identified the number of classes for Model 2 with normal data. The overall results were very similar to Model 1 using normal data. There was a general positive relationship between sample size and the performance rate. When $N = 180$ the average p was .726. When $N = 540$ the average p increased to .752, and finally, when $N = 1080$ the performance increased to .798. The number of classes had a negative relationship with the performance rate. At $k = 1$ and $k = 2$, p was consistently high, with an average of .996. However, the average p drops off significantly for $k = 3$ and $k = 4$, with an average of .522. The mixing proportion had a trivial effect on the performance rate. The range between π_1 and π_2

was only .093. Lastly, the distance between factor means and the performance rate had a small positive relationship. The average detection rate increased by just under 10% from .721 when $\Delta\eta = 1.5$ to .791 when $\Delta\eta = 3.0$.

Table 7
FMM Results - Model 2 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.992	.998	1.000	.894	.996	.998
	3	.396	.550	.594	.358	.484	.584
	4	.252	.458	.546	.246	.440	.512
2.0	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.994	1.000	1.000	.952	.998	1.000
	3	.450	.564	.640	.300	.530	.598
	4	.408	.512	.606	.380	.448	.544
2.5	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.994	1.000	1.000	.982	1.000	1.000
	3	.500	.598	.822	.472	.546	.652
	4	.412	.544	.688	.396	.460	.558
3.0	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.998	1.000	1.000	.988	1.000	1.000
	3	.546	.598	.952	.486	.548	.658
	4	.432	.564	.724	.406	.514	.574

Table 8 shows the proportion of replications in which the correct number of classes was correctly identified for Model 2c with normal data. The overall pattern of results for the performance rate was similar to the pattern of results for Model 1 and Model 2 with normal data. The ability of FMM to detect latent classes was influenced by the sample size. When $N = 180$ the average p was .826. When $N = 540$ the average p increased to .882, and when $N = 1080$ the average p was .920. The number of classes had an impact on the ability of FMM to detect classes. FMM performed consistently

well when $k = 1$ or $k = 2$, with the average p ranging from .994 to 1.000. However, the average p was not as high when $k = 3$ and $k = 4$. The average performance rate decreased to .786 when $k = 3$ and .725 when $k = 4$. The overall pattern was a negative relationship between number of classes and detection rate. The mixing proportion had a marginal impact on how well FMM was able to perform. The average performance rate was .963 at π_1 and decreased to .789 at π_2 . Finally, the class separation played a role in how well FMM detected latent classes. The average p increased from .824 to .922 when $\Delta\eta$ increased from 1.5 to 3.0.

Table 8
FMM Results - Model 2c with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.994	1.000	1.000	.910	.998	1.000
	3	.474	.802	.986	.414	.584	.968
	4	.424	.606	.898	.302	.516	.890
2.0	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.994	1.000	1.000	.962	.998	1.000
	3	.590	.876	.994	.540	.616	.974
	4	.456	.666	.994	.432	.566	.956
2.5	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.996	1.000	1.000	.988	1.000	1.000
	3	.620	.898	.996	.570	.764	.986
	4	.586	.864	.998	.518	.814	.966
3.0	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.998	1.000	1.000	.990	1.000	1.000
	3	.798	.962	1.000	.612	.838	.992
	4	.588	.956	.998	.548	.884	.974

Table 9 shows the proportion of replications for which FMM correctly identified the true number of classes in Model 3 when the data were normal. There was a positive relationship between sample size and performance. When $N = 180$ the average p was .676 and increased to .756 when $N = 1080$. The number of classes had an influence on the performance rate, as well. At the $k = 1$ level the average performance was 1.000. However, when $k = 4$ the average detection rate decreased to .380. The mixing proportion had a trivial influence on how well FMM performed. When the mixing proportion was π_1 the average p was .760; the average p decreased to .674 when π_2 . When a difference in p existed, π_1 tended to perform better. Finally, the latent class separation played a role in how well FMM was able to detect latent classes in Model 3. The average p increased from .679 to .751 when the class separation increased from $\Delta\eta = 1.5$ to $\Delta\eta = 3.0$, respectively. As with the other models, there was a positive relationship between class separation and the performance rate. The difference between the two mixing proportions was largest for Model 3 when compared to the other models.

Table 9
FMM Results - Model 3 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.994	1.000	1.000	.796	1.000	1.000
	3	.346	.450	.504	.204	.368	.490
	4	.318	.372	.468	.256	.338	.392
2.0	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.996	1.000	1.000	.856	1.000	1.000
	3	.440	.484	.506	.378	.460	.500
	4	.320	.380	.494	.270	.340	.432
2.5	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.996	1.000	1.000	.922	1.000	1.000
	3	.458	.530	.874	.432	.482	.596
	4	.332	.386	.502	.288	.354	.442
3.0	1	1.000	1.000	1.000	1.000	1.000	1.000
	2	.998	1.000	1.000	.976	1.000	1.000
	3	.466	.540	1.000	.434	.512	.718
	4	.340	.390	.592	.290	.378	.446

Table 10 displays the proportion of replications for which FMM correctly identified the true number of latent classes in Model 1 with non-normal data. The performance rate had a positive relationship with the sample size. As N increased p increased, as well. The average p was .112 when $N = 180$, and increased over 4-fold to .555 when $N = 1080$. The number of classes had a negative relationship with the performance. As k increased the detection rate diminished. The average p was .495 when $k = 1$, but decreased approximately 70% to .150 when $k = 4$. Although the mixing proportion had a differential effect on the performance rate the effect was weak. The difference between the average p at π_1 and the average p at π_2 was only .052. When π did affect

the performance rate, FMM performed better when π_1 . Finally, the distance between factor means and the average performance rate had a positive relationship. The average p increased by over 45% from .259 when $\Delta\eta = 1.5$ to .376 when $\Delta\eta = 3.0$.

Table 10
FMM Results - Model 1 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.248	.502	.738	.248	.497	.738
	2	.158	.330	.510	.034	.234	.490
	3	.024	.126	.478	.014	.084	.444
	4	.016	.064	.086	.008	.062	.078
2.0	1	.248	.502	.738	.248	.497	.738
	2	.178	.442	.646	.046	.266	.612
	3	.062	.230	.506	.020	.152	.464
	4	.020	.104	.322	.014	.090	.302
2.5	1	.248	.502	.738	.248	.497	.738
	2	.182	.446	.674	.062	.288	.634
	3	.128	.342	.518	.022	.212	.476
	4	.022	.110	.490	.014	.104	.454
3.0	1	.248	.502	.738	.248	.497	.738
	2	.286	.534	.742	.064	.376	.670
	3	.136	.418	.726	.046	.224	.572
	4	.026	.134	.504	.020	.104	.462

Table 11 shows the proportion of replications for which FMM correctly identified the number of classes for Model 2 with non-normal data. There was a general positive relationship between sample size and the performance rate. When $N = 180$ the average p rate was .216. When $N = 540$ the average p increased to .291, and finally, when $N = 1080$ the average p increased again to .351. The number of classes had a negative relationship with the average performance rate. An increase in k caused a decrease in p . The average p decreased from .618 when $k = 1$ to .111 when $k = 4$. The

mixing proportion had a marginal effect on the performance rate. The range of average performance rates between π_1 and π_2 was .157 and FMM favored the balanced design condition. Lastly, the relationship between factor means and the ability of FMM to detect latent classes was small and positive. The average detection rate increased by just under 22% from .262 when $\Delta\eta = 1.5$ to .319 when $\Delta\eta = 3.0$.

Table 11
FMM Results - Model 2 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.606	.612	.636	.606	.612	.636
	2	.074	.138	.528	.058	.114	.138
	3	.028	.080	.228	.010	.036	.128
	4	.000	.042	.224	.000	.034	.138
2.0	1	.606	.612	.636	.606	.612	.636
	2	.142	.188	.568	.100	.166	.442
	3	.028	.128	.290	.012	.106	.272
	4	.002	.062	.256	.000	.046	.146
2.5	1	.606	.612	.636	.606	.612	.636
	2	.200	.250	.578	.108	.204	.500
	3	.032	.130	.348	.014	.116	.300
	4	.002	.078	.312	.000	.056	.156
3.0	1	.606	.612	.636	.606	.612	.636
	2	.264	.334	.592	.116	.272	.544
	3	.046	.144	.464	.032	.116	.420
	4	.004	.092	.520	.000	.070	.414

Table 12 shows the proportion of replications in which FMM detected the correct number of classes for Model 2c with non-normal data. The ability of FMM to detect latent classes was marginally influenced by the sample size. When $N = 180$ the average p was .338. When $N = 540$ the average p increased to .374, and when $N = 1080$ the average p was .453. The number of classes had an influence on the ability of FMM to

detect classes. As the number of classes increased the performance rate decreased. The average performance rate decreased from .653 when $k = 1$ to .210 when $k = 4$. The overall pattern was a negative relationship between k and p . The mixing proportion had a marginal impact on how well FMM was able to perform. The average p was .487 at π_1 and decreased to .290 at π_2 . Finally, the class separation played a role in how well FMM detected latent classes. The average p increased from .311 to .469 when separation increased from $\Delta\eta = 1.5$ to $\Delta\eta = 3.0$.

Table 12
FMM Results - Model 2c with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.640	.652	.668	.640	.652	.668
	2	.118	.436	.582	.072	.238	.534
	3	.088	.166	.262	.034	.162	.198
	4	.024	.138	.200	.022	.130	.150
2.0	1	.640	.652	.668	.640	.652	.668
	2	.126	.472	.656	.078	.284	.578
	3	.152	.192	.396	.104	.172	.288
	4	.052	.184	.320	.022	.180	.220
2.5	1	.640	.652	.668	.640	.652	.668
	2	.264	.594	.732	.154	.510	.664
	3	.228	.358	.522	.114	.254	.364
	4	.060	.350	.504	.026	.194	.354
3.0	1	.640	.652	.668	.640	.652	.668
	2	.286	.648	.804	.248	.516	.668
	3	.266	.500	.536	.144	.304	.512
	4	.136	.476	.506	.096	.198	.486

Table 13 shows the proportion of replications in which FMM correctly identified the number of classes for Model 3 with non-normal data. The ability of FMM to detect latent classes was marginally influenced by the sample size. When $N = 180$ the average

p was .192. When $N = 540$ the average p increased to .273, and when $N = 1080$ the average p was .332. In design conditions where N influenced the performance of FMM, the relationship was positive. The number of classes had a negative influence on the ability of FMM to detect the correct number of classes. The average p decreased from .622 at $k = 1$ to .074 when $k = 4$. The mixing proportion had a small impact on how well FMM was able to perform. The average performance rate was .336 at π_1 and decreased to .196 at π_2 . Finally, the class separation played a negligible role in how well FMM detected latent classes. The average p increased from .234 to .289 when separation increased from $\Delta\eta = 1.5$ to $\Delta\eta = 3.0$.

Table 13
FMM Results - Model 3 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.562	.620	.684	.562	.620	.684
	2	.052	.102	.476	.046	.066	.332
	3	.022	.080	.252	.014	.040	.172
	4	.000	.010	.176	.000	.002	.122
2.0	1	.562	.620	.684	.562	.620	.684
	2	.102	.122	.482	.096	.106	.390
	3	.028	.096	.292	.014	.062	.236
	4	.002	.036	.258	.000	.012	.130
2.5	1	.562	.620	.684	.562	.620	.684
	2	.126	.196	.498	.100	.150	.420
	3	.044	.116	.322	.014	.086	.270
	4	.002	.052	.264	.000	.044	.140
3.0	1	.562	.620	.684	.562	.620	.684
	2	.130	.248	.504	.104	.214	.456
	3	.074	.130	.426	.022	.110	.366
	4	.004	.060	.268	.000	.048	.152

Q-Factor Analysis Results

Table 14 presents the proportion of replications for which the correct number of latent classes was identified for Model 1 using normal data. The performance of QFA was slightly influenced by the sample size. The average performance was .338 when $N = 180$, .361 when $N = 540$, and .369 when $N = 1080$. As the sample size increased, the average performance rate slight increased. The number of classes had an influence on the performance rate of QFA. At the $k = 1$ level the average performance was .541. When $k = 2$ the average performance decreased to .371. The average performance was .271 when $k = 3$, and finally when $k = 4$ the average performance of QFA was .241. Although the range in average performance rates between $k = 1$ and $k = 4$ was .300, the range between $k = 2$ to $k = 4$ was much smaller with a range of .130. The mixing proportion was inconsequential in how well QFA performed. At the π_1 level the average performance was .353; while the average performance was .359 when π_2 . Finally, the class separation did not play not a role in how well QFA detects latent classes. The average performance rates ranged from .347 to .362.

Table 14
QFA Results - Model 1 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.520	.550	.578	.520	.550	.578
	2	.352	.362	.378	.320	.346	.348
	3	.238	.260	.274	.238	.268	.274
	4	.200	.204	.228	.218	.246	.282
2.0	1	.520	.550	.578	.520	.550	.578
	2	.356	.378	.392	.320	.362	.380
	3	.252	.266	.274	.250	.272	.282
	4	.212	.226	.246	.243	.262	.296
2.5	1	.520	.550	.578	.520	.550	.578
	2	.362	.388	.398	.360	.378	.394
	3	.264	.272	.288	.264	.284	.292
	4	.222	.248	.264	.244	.262	.206
3.0	1	.520	.550	.578	.520	.550	.578
	2	.364	.397	.404	.364	.380	.420
	3	.270	.274	.294	.270	.294	.294
	4	.226	.256	.261	.247	.274	.212

Table 15 shows the proportion of replications in which the correct number of classes was correctly identified for Model 2 under the normal data condition. Similar to Model 1, the performance of QFA was marginally influenced by the sample size. When $N = 180$ the average performance was .315. When $N = 540$ the average performance was .324 and when $N = 1080$ the average performance was .330. The number of classes was the most significant factor to influence the performance rate. At the $k = 1$ level the average performance was .657. When $k = 2$ the average performance was .269. The average performance was .194 when $k = 3$, and finally decreased to .176 when $k = 4$. The performance generally decreased as the number of classes increased. The mixing proportion had a trivial influence on how well QFA performed. When the mixing propor-

tion was π_1 the average performance rate was .329; the average performance decreased slightly to .319 at the π_2 level. When a difference in performance based on the mixing proportion existed, the balance design condition tended to perform better. Finally, the class separation did not play a role in how well QFA detected latent classes. The average performance increased from .316 to .33. The increase in average performance from $\Delta\eta = 1.5$ to $\Delta\eta = 3.0$.

Table 15
QFA Results - Model 2 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.664	.652	.654	.664	.652	.654
	2	.246	.272	.282	.252	.254	.258
	3	.178	.188	.196	.162	.174	.178
	4	.156	.180	.196	.130	.162	.168
2.0	1	.664	.652	.654	.664	.652	.654
	2	.256	.274	.286	.254	.262	.266
	3	.184	.196	.206	.174	.182	.206
	4	.158	.182	.198	.156	.162	.168
2.5	1	.664	.652	.654	.664	.652	.654
	2	.260	.284	.296	.256	.262	.266
	3	.188	.198	.212	.176	.204	.216
	4	.176	.186	.200	.158	.164	.180
3.0	1	.664	.652	.654	.664	.652	.654
	2	.272	.284	.302	.258	.276	.282
	3	.194	.204	.226	.180	.206	.224
	4	.184	.200	.218	.164	.180	.196

Table 16 displays the proportion of replications for which the correct number of classes was correctly identified for Model 2c using normal data. When $N = 180$ the average performance was .325. When $N = 540$ the average performance was .344 and when $N = 1080$ the average performance was .361. The number of classes had a marginal

influence on the performance of QFA. At the $k = 1$ level the average performance was .616 and decreased to .216 when $k = 4$. The mixing proportion did not have an impact on how well QFA performed. The average performance rate was .344 when the mixing proportion was balanced and increased slightly to .343 when the mixing proportion was imbalanced. Finally, the class separation played a trivial role in how well QFA performed. The range of average performance rates was only .330 to .357.

Table 16
QFA Results - Model 2c with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.616	.648	.652	.616	.648	.652
	2	.254	.274	.284	.266	.276	.290
	3	.192	.206	.220	.192	.192	.216
	4	.202	.210	.218	.168	.192	.234
2.0	1	.616	.648	.652	.616	.648	.652
	2	.272	.288	.302	.270	.284	.316
	3	.196	.210	.238	.196	.210	.228
	4	.208	.212	.220	.206	.224	.250
2.5	1	.616	.648	.652	.616	.648	.652
	2	.288	.310	.318	.274	.288	.318
	3	.204	.224	.250	.202	.212	.230
	4	.208	.214	.248	.210	.226	.256
3.0	1	.616	.648	.652	.616	.648	.652
	2	.288	.314	.348	.322	.326	.348
	3	.218	.228	.264	.206	.226	.234
	4	.210	.238	.248	.220	.230	.262

Table 17 shows the proportion of replications in which the correct number of classes was correctly identified for Model 3 using normal data. The overall results were very similar to the other models using normal data. The relationship between the average performance rate and the sample size was marginal. The range between the

average performance rates when $N = 180$ and $N = 1080$ was .036. However, it appears there was a tendency for the performance rate to increase as the sample sized increased. The number of classes had the largest impact on the detection rate. At the $k = 1$ level the average performance was .461. When $k = 2$ the average performance was .409. The average performance was .204 when $k = 3$, and finally, when $k = 4$ the average performance was .168. As the number of classes increased the ability of QFA to detect latent classes decreased. The mixing proportion had an inconsequential influence on how well QFA performed. When the mixing proportion was π_1 the average performance was .312; while the average performance was .309 when π_2 . Finally, the class separation played an insignificant role in how well QFA detected latent classes in Model 3. The performance rates ranged from .301 when $\Delta\eta = 1.5$ to .319 when $\Delta\eta = 3.0$. However, when there was a difference in performance, generally design conditions with a larger separation performed best.

Table 17
QFA Results - Model 3 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.444	.468	.470	.444	.468	.470
	2	.384	.402	.424	.372	.392	.406
	3	.170	.190	.204	.182	.198	.206
	4	.138	.154	.174	.128	.138	.186
2.0	1	.444	.468	.470	.444	.468	.470
	2	.392	.416	.438	.382	.400	.414
	3	.174	.202	.218	.190	.204	.210
	4	.142	.162	.176	.144	.150	.208
2.5	1	.444	.468	.470	.444	.468	.470
	2	.398	.418	.452	.386	.406	.422
	3	.198	.210	.226	.200	.210	.218
	4	.152	.170	.190	.148	.176	.208
3.0	1	.444	.468	.470	.444	.468	.470
	2	.412	.418	.460	.388	.412	.424
	3	.200	.214	.228	.212	.218	.220
	4	.152	.170	.206	.160	.182	.212

Table 18 shows the proportion of replications for which the correct number of classes was correctly identified for Model 1 using non-normal data. The performance of QFA model was marginally impacted by sample size. When $N = 180$ the average performance was .343. When $N = 540$ the average performance was .363 and when $N = 1080$ the average performance was .376. The number of classes had the largest effect on the performance rate of QFA. The range between the highest performing level $k = 1$ and the lowest performing level $k = 4$ was .290. The mixing proportion had a trivial influence on how well QFA performed. When the mixing proportion was π_1 the average performance was .364; the average performance was .357 when π_2 . QFA tended to perform best under the π_1 design condition. Finally, the class separation played a

very small role in how well QFA detected the latent classes. The average performance for each $\Delta\eta$ ranged from .353 to .370, increasing as the distance between latent classes increased.

Table 18
QFA Results - Model 1 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.522	.546	.566	.522	.546	.566
	2	.346	.374	.386	.312	.352	.354
	3	.248	.268	.268	.250	.254	.280
	4	.246	.244	.255	.234	.245	.278
2.0	1	.522	.546	.566	.522	.546	.566
	2	.352	.374	.388	.324	.354	.380
	3	.258	.270	.296	.254	.270	.296
	4	.222	.246	.256	.245	.282	.214
2.5	1	.522	.546	.566	.522	.546	.566
	2	.360	.376	.394	.326	.368	.382
	3	.258	.280	.306	.262	.270	.304
	4	.244	.258	.272	.266	.286	.218
3.0	1	.522	.546	.566	.522	.546	.566
	2	.370	.380	.426	.330	.378	.384
	3	.278	.292	.308	.280	.286	.312
	4	.254	.282	.286	.272	.249	.254

Table 19 presents the proportion of replications for which the correct number of classes was correctly identified for Model 2 using non-normal data. The sample size played a larger role in the ability of QFA to detect latent classes than in Model 1. When $N = 180$ the average performance was .233. The average detection rate increased by over 46% at $N = 540$, where the average performance was .341. The number of classes had a marginal positive effect on the performance rate of QFA. As the number of classes increased, the performance rate increased. However, the average performance at

$k = 2$, $k = 3$, and $k = 4$ was quite similar. The performance decreased from .724 when $k = 1$ to .137 when $k = 4$. The mixing proportion did not play a role in how well QFA performed. The difference in performance was only .002. Finally, the class separation played a very small role in how well QFA performed. The average performance rate had a positive relationship with the class separation. As the distance increased, the average performance rate increased as well. When $\Delta\eta = 1.5$ the average performance rate was .279 and increased to .295 when $\Delta\eta = 3.0$.

Table 19
QFA Results - Model 2 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		180	540	1080	180	540	1080
1.5	1	.664	.722	.796	.644	.722	.796
	2	.090	.152	.190	.090	.140	.220
	3	.072	.102	.152	.088	.120	.176
	4	.064	.128	.152	.0920	.146	.166
2.0	1	.664	.722	.796	.644	.722	.796
	2	.076	.152	.210	.092	.146	.222
	3	.092	.130	.170	.094	.124	.186
	4	.094	.132	.158	.102	.158	.166
2.5	1	.664	.722	.796	.644	.722	.796
	2	.098	.154	.214	.094	.146	.226
	3	.098	.130	.182	.102	.130	.190
	4	.098	.142	.162	.108	.164	.176
3.0	1	.664	.722	.796	.644	.722	.796
	2	.116	.164	.244	.001	.174	.234
	3	.100	.144	.190	.128	.164	.192
	4	.120	.146	.180	.114	.132	.194

Table 20 shows the proportion of replications for which the correct number of classes was identified for Model 2c using non-normal data. QFA did not perform as well with Model 2c as the other models using non-normal data. The performance of

the QFA model was positively impacted by sample size. When $N = 180$ the average performance was .035. When $N = 540$ the average performance was .093 and when $N = 1080$ the average performance was .185. The number of classes had a trivial effect on the performance rate at design conditions $k > 1$. When $k = 1$ the average detection rate was .710. However, from $k = 2$ to $k = 4$ the average performance ranged from .190 to .180, with the rate decreasing as k increased. The mixing proportion did not have an influence on how well QFA performed. When the mixing proportion was π_1 the average performance was .312; and the performance was .315 when π_2 . The class separation played a minor role in how well QFA detected latent classes in Model 2c. The average performance rate went from .295 when $\Delta\eta = 1.5$ to .332 when $\Delta\eta = 3.0$.

Table 20
QFA Results - Model 2c with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.644	.690	.796	.644	.690	.796
	2	.078	.178	.234	.102	.158	.204
	3	.080	.094	.204	.098	.144	.196
	4	.104	.168	.228	.130	.190	.218
2.0	1	.644	.690	.796	.644	.690	.796
	2	.092	.192	.242	.148	.182	.222
	3	.098	.180	.230	.128	.146	.224
	4	.120	.156	.218	.110	.202	.214
2.5	1	.644	.690	.796	.644	.690	.796
	2	.158	.204	.262	.112	.196	.270
	3	.120	.158	.224	.168	.204	.240
	4	.106	.204	.232	.160	.186	.224
3.0	1	.644	.690	.796	.644	.690	.796
	2	.198	.224	.266	.174	.204	.270
	3	.140	.210	.246	.170	.198	.250
	4	.140	.208	.242	.128	.188	.244

Table 21 shows the proportion of replications in which the correct number of classes was correctly identified for Model 3 using non-normal data. QFA did not perform as well with Model 3 as the other models using non-normal data. The performance of the QFA model was positively impacted by sample size. When $N = 180$ the average performance was .035. When $N = 540$ the average performance was .093 and when $N = 1080$ the average performance was .185. The number of classes had a small impact on the performance rate. When $k = 1$ the average detection rate was .159. When $k = 4$ the average performance ranged decreased to .073, with the rate decreasing as k increased. The mixing proportion did not have an influence on how well QFA performed. When the mixing proportion was π_1 the average performance was .104 and the average performance was .105 when π_2 . The class separation played a minor but positive role in how well QFA detected latent classes in Model 3. The average performance rates went from .096 when $\Delta\eta = 1.5$ to .113 when $\Delta\eta = 3.0$.

Table 21
QFA Results - Model 3 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.044	.138	.296	.044	.138	.296
	2	.026	.078	.154	.012	.072	.166
	3	.030	.064	.140	.040	.070	.128
	4	.014	.056	.108	.030	.054	.106
2.0	1	.044	.138	.296	.044	.138	.296
	2	.042	.078	.172	.040	.098	.174
	3	.030	.070	.142	.032	.072	.140
	4	.018	.066	.110	.016	.064	.108
2.5	1	.044	.138	.296	.044	.138	.296
	2	.044	.074	.178	.042	.100	.176
	3	.040	.080	.150	.040	.086	.156
	4	.032	.078	.134	.016	.080	.134
3.0	1	.044	.138	.296	.044	.138	.296
	2	.024	.082	.202	.054	.094	.186
	3	.052	.084	.162	.040	.094	.158
	4	.032	.080	.136	.026	.108	.138

Marcoulides and Drezner Model Results

The ability of MDM to detect latent classes under each of the design conditions is presented in Table 22 through Table 29. Table 22 displays the proportion of replications for which MDM correctly identified the true number of latent classes in Model 1 when the data are normal. The performance rate had a positive relationship with the sample size. As N increased the detection rate increased, as well. The average detection rate was .203 when $N = 180$. The detection rate increased to .284 when $N = 540$, and then finally, increased to .354 when $N = 1080$. The number of classes had the strongest relationship

with the performance rate of MDM. As k increased the detection rate decreased, as well. The average detection rate was .335 when $k = 1$, increased slightly to .386 when $k = 2$, then decreased to .287 when $k = 3$, and finally to .112 when $k = 4$. The mixing proportion had a weak effect on the ability of MDM to detect latent classes. The range in performance rates between π_1 and π_2 was only .019. Finally, the distance between the latent classes played a small role in the performance. The performance of MDM tended to increase as $\Delta\eta$ increased, but the increase was flat.

Table 22
MDM Results - Model 1 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.156	.342	.508	.156	.342	.508
	2	.310	.382	.394	.328	.376	.382
	3	.168	.236	.254	.196	.254	.318
	4	.044	.046	.086	.040	.044	.052
2.0	1	.156	.342	.508	.156	.342	.508
	2	.316	.382	.412	.342	.380	.414
	3	.184	.238	.314	.248	.318	.360
	4	.060	.088	.140	.092	.096	.120
2.5	1	.156	.342	.508	.156	.342	.508
	2	.338	.386	.426	.358	.382	.446
	3	.188	.282	.348	.300	.356	.368
	4	.086	.120	.172	.130	.144	.158
3.0	1	.156	.342	.508	.156	.342	.508
	2	.380	.410	.448	.366	.428	.486
	3	.214	.302	.360	.328	.380	.382
	4	.092	.128	.208	.142	.180	.226

Table 23 presents the proportion of replications for which MDM identified the correct number of latent classes in Model 2 when the data were normal. The pattern of results was similar to what was found for Model 1. The relationship between the

performance rate and sample size was positive. The average performance increased from .203 to .372 as the sample size increased from $N = 180$ to $N = 1080$. The number of classes had a moderate positive relationship with the performance rate. Generally, as k increased the detection rate decreased. The average detection rate was .383 when $k = 1$, increased slightly to .396 when $k = 2$ and then decreased from .242 to .141 as the number of classes increased from $k = 3$ to $k = 4$. The relationship between performance and mixing proportion was trivial. The range in average performance rates was only .020. Lastly, the relationship between factor means and the performance rate was small. The average detection rate increased from .267 when $\Delta\eta = 1.5$ to .317 when $\Delta\eta = 3.0$.

Table 23
MDM Results - Model 2 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.178	.414	.558	.178	.414	.558
	2	.332	.368	.388	.352	.366	.420
	3	.120	.168	.270	.124	.202	.304
	4	.048	.092	.172	.086	.120	.164
2.0	1	.178	.414	.558	.178	.414	.558
	2	.350	.376	.414	.364	.392	.426
	3	.126	.208	.288	.220	.260	.308
	4	.052	.142	.182	.090	.144	.178
2.5	1	.178	.414	.558	.178	.414	.558
	2	.358	.398	.422	.388	.406	.446
	3	.134	.226	.296	.230	.274	.336
	4	.054	.150	.182	.144	.156	.182
3.0	1	.178	.414	.558	.178	.414	.558
	2	.364	.398	.488	.392	.432	.460
	3	.180	.254	.356	.258	.332	.340
	4	.148	.160	.206	.152	.162	.218

Table 24 presents the proportion of replications for which MDM correctly identified the true number of classes in Model 2c when the data were normal. The overall pattern of results was similar to the pattern of results for Model 1 and Model 2 with normal data. When $N = 180$ the average performance rate was .195 and increased to .363 when $N = 1080$. The relationship between the ability of MDM to detect latent classes and the number of classes was consistent with the other models. While $k = 1$ the average performance rate was .351 and increased slightly to .385 when $k = 2$. The average performance rate then decreased to .138 when $k = 4$. The mixing proportion had a trivial impact on how well MDM performed. The average performance rate was .268 when the mixing proportion was balanced and decreased to .295 when the mixing proportion was imbalanced. In some design conditions, π_2 outperformed π_1 . However, the difference in average performances between the two levels was not significant. Finally, the class separation played a small role in how well MDM detected latent classes. The average performance increased from .258 to .304 as $\Delta\eta$ increased from 1.5 to 3.0.

Table 24
MDM Results - Model 2c with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.154	.370	.530	.154	.370	.530
	2	.312	.320	.382	.338	.358	.412
	3	.098	.192	.262	.214	.236	.262
	4	.042	.114	.156	.064	.154	.168
2.0	1	.154	.370	.530	.154	.370	.530
	2	.332	.344	.394	.348	.398	.432
	3	.142	.236	.296	.214	.242	.324
	4	.052	.130	.158	.122	.156	.174
2.5	1	.154	.370	.530	.154	.370	.530
	2	.342	.378	.440	.358	.406	.462
	3	.162	.244	.320	.228	.316	.328
	4	.086	.132	.176	.132	.156	.192
3.0	1	.154	.370	.530	.154	.370	.530
	2	.358	.392	.456	.364	.420	.484
	3	.186	.274	.320	.286	.318	.380
	4	.088	.152	.198	.144	.170	.206

Table 25 shows the proportion of replications for which MDM correctly identified the true number of classes in Model 3 when the data were normal. The sample size had a positive relationship with the performance rate, as one increased the other increased. The average performance rate improved from .199 to .389 as the sample size increased from $N = 180$ to $N = 1080$. When $k = 1$ and $k = 2$ the average performance rate was approximately .400 and declined to .138 when $k = 4$. The mixing proportion had a trivial impact on how well MDM performed. The average performance rate was .289 when the mixing proportion was balanced and increased to .302 when the mixing proportion was

imbalanced. In some design conditions, π_2 outperformed π_1 . In other design conditions, the opposite was true. Finally, the class separation played just a minor role in the ability of MDM to detect latent classes.

Table 25
MDM Results - Model 3 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.185	.434	.644	.185	.434	.644
	2	.324	.376	.408	.372	.376	.408
	3	.112	.188	.270	.126	.222	.298
	4	.044	.098	.180	.042	.100	.170
2.0	1	.185	.434	.644	.185	.434	.644
	2	.344	.386	.426	.374	.392	.424
	3	.120	.206	.272	.156	.224	.304
	4	.050	.100	.182	.110	.136	.172
2.5	1	.185	.434	.644	.185	.434	.644
	2	.346	.400	.428	.378	.408	.440
	3	.150	.206	.274	.198	.260	.316
	4	.078	.164	.182	.132	.150	.172
3.0	1	.185	.434	.644	.185	.434	.644
	2	.350	.402	.434	.386	.428	.442
	3	.186	.234	.372	.218	.270	.318
	4	.144	.174	.214	.154	.170	.198

Table 26 displays the proportion of replications for which MDM correctly identified the true number of latent classes in Model 1 when the data were non-normal. The performance rate had a positive relationship with the sample size. As N increased the detection rate increased, as well. The average detection rate was .213 when $N = 180$. The average detection rate increased to .273 when $N = 540$, and then finally, increased to .337 when $N = 1080$. The number of classes had a strong relationship with the performance rate. As k increased the detection rate declined. The average detection rate

ranged from .327 when $k = 1$ to .102 when $k = 4$. The mixing proportion had a weak effect on the detection rate. The range between the average performance rate of π_1 and the average performance rate of π_2 was only .019 and favored the imbalanced condition. Finally, the distance between factor means and the performance rate was somewhat flat; the detection increased from .252 to .298 as $\Delta\eta$ increased.

Table 26
MDM Results - Model 1 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.206	.328	.446	.206	.328	.446
	2	.362	.382	.416	.386	.404	.440
	3	.126	.206	.264	.116	.234	.296
	4	.036	.080	.096	.042	.094	.100
2.0	1	.206	.328	.446	.206	.328	.446
	2	.368	.402	.418	.392	.414	.456
	3	.142	.214	.288	.218	.266	.310
	4	.056	.082	.104	.070	.116	.136
2.5	1	.206	.328	.446	.206	.328	.446
	2	.380	.414	.464	.406	.418	.538
	3	.190	.234	.288	.242	.270	.324
	4	.076	.090	.122	.072	.126	.142
3.0	1	.206	.328	.446	.206	.328	.446
	2	.400	.422	.510	.412	.420	.554
	3	.212	.274	.294	.248	.304	.332
	4	.114	.118	.160	.112	.134	.176

Table 27 presents the proportion of replications for which MDM identified the correct number of latent classes in Model 2 when the data were non-normal. The performance rate had a positive relationship with the sample size. As N increased the detection rate increased, as well. The average detection rate was .184 when $N = 180$. The average detection rate increased to .282 when $N = 540$, and then finally, increased

to .371 when $N = 1080$. The number of classes had a moderate negative relationship with the performance rate. As k increased the detection rate decreased. The average detection rates ranged from .531 to .079. The mixing proportion had a weak effect on the detection rate. The range between the average performance rates of π_1 and π_2 was only .012; the imbalanced condition had a higher rate. Finally, the relationship between the factor means and the performance rate was trivial; as $\Delta\eta$ increased from 1.5 to 3.0 the detection increased from .262 to .297.

Table 27
MDM Results - Model 2 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.354	.556	.684	.354	.556	.684
	2	.210	.356	.394	.266	.360	.384
	3	.042	.100	.210	.086	.154	.192
	4	.014	.024	.128	.004	.068	.112
2.0	1	.354	.556	.684	.354	.556	.684
	2	.268	.364	.422	.278	.362	.392
	3	.054	.114	.222	.088	.156	.262
	4	.018	.046	.144	.008	.070	.146
2.5	1	.354	.556	.684	.354	.556	.684
	2	.294	.374	.424	.286	.372	.400
	3	.058	.118	.222	.098	.156	.264
	4	.018	.052	.152	.040	.080	.154
3.0	1	.354	.556	.684	.354	.556	.684
	2	.300	.424	.440	.320	.382	.416
	3	.070	.132	.272	.110	.160	.312
	4	.024	.064	.158	.088	.096	.178

Table 28 shows the proportion of replications for which MDM correctly identified the true number of classes in Model 2c when the data were non-normal. The overall pattern of results was similar to the pattern of results for Model 1 and Model 2 with non-

normal data. There was a positive relationship between the sample size and the ability of MDM to detect latent classes. When $N = 180$ the average performance rate was .199 and increased to .361 when $N = 1080$. The sample size had a positive relationship with the performance rate, as one increased the other increased. The average performance improved from .199 to .361 as the sample size increased from $N = 180$ to $N = 1080$. The range of the performance rates based on the number of classes was more varied with this model. The average performance rate for $k = 1$ was .451 and declined to .088 when $k = 4$. The mixing proportion had a trivial impact on how well MDM performed. The average performance rate was .274 when the mixing proportion was balanced and increased to .285 when the mixing proportion was imbalanced. In some design conditions, π_2 outperformed π_1 . In other conditions, π_1 had a better performance rate. Finally, the class separation played a minor role in the ability of MDM to detect latent classes, with average performances ranging from .264 to .295 as $\Delta\eta$ increased.

Table 28
MDM Results - Model 2c with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.282	.462	.610	.282	.462	.610
	2	.374	.390	.392	.262	.372	.430
	3	.072	.138	.240	.136	.174	.254
	4	.004	.042	.116	.036	.072	.130
2.0	1	.282	.462	.610	.282	.462	.610
	2	.382	.392	.410	.338	.392	.438
	3	.086	.152	.252	.136	.180	.266
	4	.036	.068	.120	.038	.094	.138
2.5	1	.282	.462	.610	.282	.462	.610
	2	.386	.400	.440	.342	.402	.446
	3	.092	.160	.258	.138	.186	.268
	4	.036	.070	.126	.054	.098	.154
3.0	1	.282	.462	.610	.282	.462	.610
	2	.386	.402	.460	.380	.412	.452
	3	.094	.178	.286	.158	.204	.286
	4	.056	.084	.144	.094	.126	.178

Table 29 shows the proportion of replications for which MDM correctly identified the true number of classes in Model 3 when the data were non-normal. There was a positive relationship between the sample size and the ability of MDM to detect latent classes. When $N = 180$ the average performance rate was .181 and increased to .385 when $N = 1080$. The range of the performance rate based on the number of classes was more varied with this model. The average performance rate for $k = 1$ was .603 and declined to .072 when $k = 4$. The mixing proportion had a trivial impact on how well MDM performed. The average performance rate was .281 when the mixing proportion was balanced and increased to .286 when the mixing proportion was imbalanced. In some design conditions, π_2 outperformed π_1 . In other design conditions, the method

performed better when π_1 . Finally, the class separation played a minor role in the ability of MDM to detect latent classes, with average performances that increased from .261 to .306 as $\Delta\eta$ increased from 1.5 to 3.0.

Table 29
MDM Results - Model 3 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.418	.634	.756	.418	.634	.756
	2	.176	.322	.402	.198	.284	.362
	3	.014	.082	.192	.024	.120	.202
	4	.006	.022	.086	.008	.056	.102
2.0	1	.418	.634	.756	.418	.634	.756
	2	.178	.338	.414	.228	.310	.394
	3	.022	.114	.196	.034	.128	.220
	4	.014	.038	.138	.016	.058	.138
2.5	1	.418	.634	.756	.418	.634	.756
	2	.250	.356	.422	.278	.332	.428
	3	.062	.116	.226	.054	.132	.246
	4	.024	.052	.154	.022	.072	.168
3.0	1	.418	.634	.756	.418	.634	.756
	2	.334	.358	.480	.280	.348	.444
	3	.062	.116	.228	.090	.138	.286
	4	.040	.054	.156	.036	.094	.174

McDonald Eigen Analysis Results

Table 30 presents the proportion of replications for which MEA was able to detect the true number of latent classes for Model 1 using normal data. The performance of MEA was influenced by the sample size. The average p was .324 when $N = 180$, .378 when $N = 540$, and increased to .441 when $N = 1080$. As the sample size increased, the average performance rate increased, as well. The number of classes had an effect

on the ability of MEA to detect latent classes. As the number of classes increased, the performance rate steadily declined. At the $k = 1$ level the average p was .639. When $k = 2$ the average p decreased to .468. The average p was .277 when $k = 3$, and finally when $k = 4$ the average p of MEA was .141. The mixing proportion had a very small impact on how well MEA performed. At the π_1 level the average p was .396; while the average p was .366 when π_2 . When there was a difference in performance, the balanced design conditions tended to have a higher rate. Finally, the class separation only played a minor role in how well MEA detected the latent classes. The average p ranged from .396 to .445, with the performance generally improving as $\Delta\eta$ increased.

Table 30
MEA Results - Model 1 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.590	.634	.692	.590	.634	.692
	2	.378	.510	.546	.326	.360	.434
	3	.174	.202	.302	.140	.214	.262
	4	.102	.120	.140	.062	.124	.126
2.0	1	.590	.634	.692	.590	.634	.692
	2	.378	.452	.532	.348	.402	.454
	3	.190	.294	.372	.174	.236	.366
	4	.114	.120	.168	.078	.122	.144
2.5	1	.590	.634	.692	.590	.634	.692
	2	.422	.504	.566	.394	.428	.528
	3	.240	.328	.378	.188	.256	.398
	4	.122	.162	.204	.080	.140	.164
3.0	1	.590	.634	.692	.590	.634	.692
	2	.522	.554	.592	.492	.526	.586
	3	.280	.342	.430	.198	.292	.388
	4	.126	.194	.286	.108	.146	.224

Table 31 shows the proportion of replications for which MEA was able to detect the true number of latent classes for Model 2 using normal data. Similar to Model 1, the performance of MEA was marginally influenced by the sample size and increased as the sample size increased. When $N = 180$ the average p was .347. When $N = 540$ the average p increased to .385 and when $N = 1080$ the average p increased again to .449. Although the performance consistently increased, the range between $N = 180$ and $N = 1080$ was only .064. The number of classes was the most significant factor to influence the performance rate. At the $k = 1$ level the average p was .607. When $k = 2$ the average p decreased to .556. The average p was .244 when $k = 3$, and finally, decreased to .168 when $k = 4$. The performance generally decreased as the number of classes increased. The mixing proportion had a minor influence on how well MEA performed. When the mixing proportion was π_1 the average performance was .402; the average p decreased slightly to .385 at the π_2 level. When a difference in performance based on the mixing proportion existed, the balance design condition tended to perform best. Finally, the class separation did not play a significant role in how well MEA detected latent classes. The average performance increased from .375 to .411 as $\Delta\eta$ increased from 1.5 to 3.0.

Table 31
MEA Results - Model 2 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.546	.604	.670	.546	.604	.670
	2	.492	.520	.620	.464	.500	.558
	3	.202	.238	.274	.178	.218	.242
	4	.126	.146	.194	.074	.130	.180
2.0	1	.546	.604	.670	.546	.604	.670
	2	.500	.528	.642	.484	.526	.578
	3	.206	.244	.274	.198	.230	.264
	4	.126	.160	.244	.114	.144	.220
2.5	1	.546	.604	.670	.546	.604	.670
	2	.536	.544	.680	.522	.542	.594
	3	.228	.246	.284	.202	.232	.282
	4	.128	.166	.246	.124	.154	.238
3.0	1	.546	.604	.670	.546	.604	.670
	2	.558	.580	.680	.538	.562	.596
	3	.232	.284	.322	.220	.260	.284
	4	.146	.168	.262	.134	.164	.242

Table 32 shows the proportion of replications for which MEA was able to detect the true number of latent classes for Model 2c using normal data. The sample size had a marginal positive effect on the ability of MEA to detect the true number of latent classes. When $N = 180$ the average p was .351 and increased to .455 when $N = 1080$. The number of classes had the most influence on the performance of MEA. At the $k = 1$ level the average p was .628 and decreased to .168 when $k = 4$. The mixing proportion had a very small impact on how well MEA performed. The average p was .409 when the mixing proportion was balanced and decreased slightly to .391 when the mixing proportion was imbalanced. When the mixing proportion played a role in the ability of MEA, the balanced condition tended to perform better than the imbalanced

condition. However, there were some design conditions where the imbalanced condition performed better. Finally, the class separation played a minor role in how well MEA performed. The average p ranged from .381 to .421. As the distance increased, the overall performance slightly increased in most design conditions.

Table 32
MEA Results - Model 2c with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.588	.618	.678	.588	.618	.678
	2	.484	.542	.598	.484	.528	.556
	3	.178	.222	.278	.146	.212	.262
	4	.086	.160	.230	.076	.126	.206
2.0	1	.588	.618	.678	.588	.618	.678
	2	.510	.542	.646	.496	.530	.592
	3	.212	.254	.284	.166	.220	.272
	4	.122	.164	.236	.092	.136	.218
2.5	1	.588	.618	.678	.588	.618	.678
	2	.512	.546	.660	.508	.536	.604
	3	.222	.260	.312	.194	.244	.272
	4	.122	.202	.244	.102	.164	.220
3.0	1	.588	.618	.678	.588	.618	.678
	2	.526	.566	.680	.508	.546	.620
	3	.280	.304	.328	.244	.294	.318
	4	.134	.204	.258	.128	.164	.244

Table 33 shows the proportion of replications for which MEA was able to detect the true number of latent classes for Model 3 using normal data. The overall results were very similar to the other models using normal data. The relationship between the average performance rate and the sample size was positive. The range between the average p when $N = 180$ and $N = 1080$ was .106. The number of classes had the largest impact on the ability of MEA to detect latent classes. At the $k = 1$ level the average

p was .607. When $k = 2$ the average p decreased to .521. The average p dropped to .238 when $k = 3$, and finally, when $k = 4$ the average p was .170. As the number of classes increased the ability of MEA to detect latent classes decreased. The mixing proportion had an inconsequential influence on how well MEA performed. When the mixing proportion was π_1 the average p was .396; while the average p slightly decreased to .372 when π_2 . Finally, the class separation played an insignificant role in how well MEA detected latent classes in Model 3. The performance rates ranged from .364 when $\Delta\eta = 1.5$ to .404 when $\Delta\eta = 3.0$. However, when there was a difference in performance, generally design conditions with a larger separation performed best.

Table 33
MEA Results - Model 3 with Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.546	.604	.670	.546	.604	.670
	2	.470	.524	.574	.442	.484	.532
	3	.180	.228	.262	.166	.218	.252
	4	.120	.122	.220	.054	.122	.126
2.0	1	.546	.604	.670	.546	.604	.670
	2	.482	.528	.576	.444	.488	.550
	3	.182	.228	.266	.174	.220	.254
	4	.122	.174	.258	.112	.152	.200
2.5	1	.546	.604	.670	.546	.604	.670
	2	.488	.534	.588	.478	.508	.552
	3	.194	.248	.322	.188	.244	.278
	4	.138	.214	.268	.114	.158	.226
3.0	1	.546	.604	.670	.546	.604	.670
	2	.498	.546	.628	.482	.532	.576
	3	.252	.270	.348	.194	.268	.284
	4	.158	.220	.280	.124	.162	.232

Table 34 shows the proportion of replications for which the correct number of classes was correctly identified for Model 1 using non-normal data. The sample size had an impact on how well MEA was able to identify the correct number of classes. When $N = 180$ the average p was .311. When $N = 540$ the average p increased to .358 and when $N = 1080$ the average p was .418. Although there was not a strong relationship between N and p , the performance rate tended to increase as the sample size increased. The number of classes had the largest effect on the performance rate of MEA. The range between the highest performing level $k = 1$ and the lowest performing level $k = 4$ was .337. The mixing proportion had a trivial influence on how well MEA performed. When the mixing proportion was π_1 the average p was .371; the average p was .354 when π_2 . MEA tended to perform better under the π_1 design condition. Finally, the class separation played a small role in how well MEA detected latent classes. The average performance for each $\Delta\eta$ ranged from .339 to .390, generally p increased as the distance between latent classes increased.

Table 34
MEA Results - Model 1 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.424	.484	.578	.424	.484	.578
	2	.424	.474	.530	.394	.460	.516
	3	.208	.260	.292	.184	.246	.262
	4	.102	.164	.226	.086	.124	.212
2.0	1	.424	.484	.578	.424	.484	.578
	2	.458	.524	.538	.444	.468	.532
	3	.240	.280	.292	.218	.248	.288
	4	.122	.168	.228	.116	.126	.214
2.5	1	.424	.484	.578	.424	.484	.578
	2	.484	.526	.576	.466	.524	.542
	3	.248	.286	.318	.244	.272	.308
	4	.124	.198	.250	.116	.128	.240
3.0	1	.424	.484	.578	.424	.484	.578
	2	.520	.552	.622	.518	.550	.572
	3	.294	.304	.378	.272	.304	.314
	4	.158	.236	.264	.126	.164	.242

Table 35 shows the proportion of replications for which MEA correctly identified the number of classes for Model 2 using non-normal data. When $N = 180$ the average p was .330. The average detection rate increased by over 44% at $N = 540$, where the average p was .405. The number of classes had a negative effect on the performance rate. As the number of classes increased, the performance rate decreased. The average p was .547 at $k = 1$ and decreased to .162 at $k = 4$. The mixing proportion played a trivial role in how well MEA performed. The difference in performance between π_1 and π_2 was only .018. Finally, the class separation had a small influence on the performance

of MEA. The average performance rate generally had a positive relationship with the class separation. When $\Delta\eta = 1.5$ the average p was .375 and increased to .377 when $\Delta\eta = 3.0$.

Table 35
MEA Results - Model 2 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.546	.604	.670	.546	.604	.670
	2	.492	.520	.620	.464	.500	.558
	3	.202	.238	.274	.178	.218	.242
	4	.126	.146	.194	.074	.130	.180
2.0	1	.494	.526	.560	.494	.526	.560
	2	.488	.514	.544	.442	.498	.526
	3	.224	.238	.278	.210	.226	.246
	4	.122	.166	.182	.090	.144	.172
2.5	1	.494	.526	.560	.494	.526	.560
	2	.490	.514	.562	.460	.506	.526
	3	.228	.262	.284	.212	.256	.268
	4	.162	.166	.232	.122	.164	.198
3.0	1	.494	.526	.560	.494	.526	.560
	2	.504	.552	.568	.480	.528	.560
	3	.242	.284	.292	.214	.268	.284
	4	.164	.178	.270	.126	.168	.214

Table 36 shows the proportion of replications for which MEA correctly identified the true number of classes for Model 2c using non-normal data. The sample size had a very small positive influence on the ability of MEA to detect latent classes. When $N = 180$ the average p was .312. When $N = 540$ the average p was .359 and when $N = 1080$ the average p was .411. As the sample size increased, the performance of MEA improved. The number of classes had a moderate effect on the average performance rate. When $k = 1$ the average detection rate was .511. However, at $k = 4$ the average p

decreased to .164. There was a consistent decline in the performance as the number of classes in the data were increased. The mixing proportion was not very helpful to MEA in detecting latent classes. When the mixing proportion was π_1 the average performance was .371 and the average performance was .351 when the mixing proportion was π_2 . The class separation played a minor role in how well MEA detected latent classes in Model 2c. The average performance rate increased from .340 when $\Delta\eta = 1.5$ to .381 when $\Delta\eta = 3.0$.

Table 36
MEA Results - Model 2c with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.446	.514	.574	.446	.514	.574
	2	.472	.514	.554	.444	.488	.524
	3	.180	.228	.268	.152	.214	.242
	4	.096	.122	.260	.086	.102	.142
2.0	1	.446	.514	.574	.446	.514	.574
	2	.488	.516	.564	.458	.498	.534
	3	.240	.248	.288	.176	.248	.264
	4	.102	.166	.260	.088	.112	.188
2.5	1	.446	.514	.574	.446	.514	.574
	2	.498	.522	.580	.470	.508	.552
	3	.242	.278	.298	.220	.260	.284
	4	.120	.170	.262	.092	.164	.224
3.0	1	.446	.514	.574	.446	.514	.574
	2	.506	.552	.590	.506	.520	.582
	3	.258	.282	.298	.250	.266	.286
	4	.152	.206	.268	.134	.176	.240

Table 37 shows the proportion of replications in which the correct number of classes was correctly identified for Model 3 using non-normal data. The performance of the MEA model was positively impacted by sample size. When $N = 180$ the average

p was .306. When $N = 540$ the average p was .348 and when $N = 1080$ the average p increased to .396 when $k = 4$. The number of classes had a moderate effect on the performance rate. When $k = 1$ the average detection rate was .490 and steadily decreased to .140 when $k = 4$. The mixing proportion had a minor influence on how well MEA performed. When the mixing proportion was π_1 the average performance was .359. The average performance was .341 when π_2 . The relationship between class separation and the performance rate was small and decreased from .328 when $\Delta\eta = 1.5$ to .364 when $\Delta\eta = 3.0$.

Table 37
MEA Results - Model 3 with Non-Normal Data

$\Delta\eta$	k	π_1			π_2		
		N			N		
		180	540	1080	180	540	1080
1.5	1	.430	.492	.548	.430	.492	.548
	2	.448	.526	.578	.432	.486	.538
	3	.220	.242	.288	.200	.230	.248
	4	.106	.122	.164	.088	.108	.158
2.0	1	.430	.492	.548	.430	.492	.548
	2	.482	.526	.584	.440	.488	.544
	3	.236	.244	.296	.200	.238	.254
	4	.112	.142	.168	.104	.116	.166
2.5	1	.430	.492	.548	.430	.492	.548
	2	.482	.528	.584	.462	.522	.562
	3	.240	.244	.326	.206	.244	.260
	4	.112	.144	.192	.108	.118	.168
3.0	1	.430	.492	.548	.430	.492	.548
	2	.486	.558	.596	.464	.524	.572
	3	.244	.282	.328	.220	.250	.312
	4	.148	.152	.232	.112	.152	.172

Comparison of Results

Table 38 summarizes p by method across all design conditions. Overall, FMM performed best. FMM correctly identified the correct number of classes in 56.2% of the replications. FMM had the greatest variance in performance, with rates ranging from .000 to 1.000. MEA was the second best performing method and correctly identified the true number of classes in 37.4% of the replications. The performance of MEA was more consistent than FMM. The range of performance rates was only .638 for MEA. QFA correctly identified 30.1% of the replications and MDM correctly identified 28.3% of the replications. The performances of QFA and MDM were more varied than MEA and had p ranges of .752 and .784, respectively.

Table 38
Performance Rate by Method

	M	SD	Min	Max
FMM	.562	.347	.000	1.000
QFA	.301	.191	.012	.796
MDM	.283	.165	.004	.756
MEA	.374	.181	.054	.692

Table 39 summarizes the performance rate by sample size. All of the methods were impacted by the sample size. With each method, the performance rate increased as the sample size increased. FMM was impacted the most by the sample size and had a difference of .222 between $N = 180$ and $N = 1080$. QFA was impacted the least and only had a difference of .066 between $N = 180$ and $N = 1080$. FMM had more variance of the performance based on the sample size, while the other models performances were more consistent.

Table 39
Performance Rate by Sample Size

	<i>N</i>	<i>M</i>	<i>SD</i>	<i>Min</i>	<i>Max</i>
FMM	180	.456	.374	.000	1.000
	540	.551	.346	.002	1.000
	1080	.678	.281	.078	1.000
QFA	180	.268	.192	.012	.664
	540	.302	.190	.054	.722
	1080	.334	.185	.106	.796
MDM	180	.197	.122	.004	.418
	540	.285	.150	.022	.634
	1080	.367	.173	.052	.756
MEA	180	.327	.174	.054	.590
	540	.370	.176	.102	.634
	1080	.425	.181	.126	.692

Table 40 shows the performance rate by the number of classes. With each method, the performance rate decreased as the number of classes increased. The range between the average performance rate of $k = 1$ and $k = 4$ for FMM was 0.450. The range was 0.371 for QFA, 0.317 for MDM, and 0.403 for MEA. FMM and MEA were influenced most by the number of classes.

Table 40
Performance Rate by Number of Classes

	<i>k</i>	<i>M</i>	<i>SD</i>	<i>Min</i>	<i>Max</i>
FMM	1	.798	.219	.248	1.000
	2	.658	.366	.034	1.000
	3	.441	.309	.010	1.000
	4	.349	.286	.000	1.000
QFA	1	.555	.179	.044	.796
	2	.271	.122	.012	.578
	3	.194	.067	.030	.312
	4	.184	.061	.026	.296
MDM	1	.425	.173	.154	.756
	2	.383	.058	.176	.554
	3	.215	.086	.014	.382
	4	.109	.055	.004	.226
MEA	1	.563	.076	.424	.692
	2	.519	.059	.326	.680
	3	.253	.048	.140	.430
	4	.160	.052	.054	.286

Table 41 presents the performance of each method by mixing proportion. The mixing proportion did not significantly influence any of the methods. Although the balanced design condition tended to perform best with FMM, QFA, and MEA, the difference in performance between the two level was negligible for QFA. Also, FMM was most impacted by the mixing proportion. The performance difference between the two levels is approximately 5%.

Table 41
Performance Rate by Mixing Proportion

		<i>M</i>	<i>SD</i>	<i>Min</i>	<i>Max</i>
FMM	Balanced	.584	.341	.000	1.000
	Imbalanced	.539	.353	.000	1.000
QFA	Balanced	.303	.193	.014	.796
	Imbalanced	.300	.189	.012	.796
MDM	Balanced	.275	.168	.004	.756
	Imbalanced	.291	.161	.004	.756
MEA	Balanced	.384	.178	.086	.692
	Imbalanced	.364	.184	.054	.692

Table 42 presents the performance rate by class separation. FMM was the only method that was significantly influenced by the distance between classes. Although the relationship between class distance and average performance rate was positive with QFA, MDM, and MEA, the increase was modest.

Table 42
Performance Rate by Distance Between Classes

	$\Delta\eta$	M	SD	Min	Max
FMM	1.5	.507	.359	.000	1.000
	2.0	.544	.347	.000	1.000
	2.5	.581	.341	.000	1.000
	3.0	.615	.336	.000	1.000
QFA	1.5	.290	.196	.012	.796
	2.0	.298	.192	.016	.796
	2.5	.305	.190	.016	.796
	3.0	.312	.186	.024	.796
MDM	1.5	.261	.171	.004	.756
	2.0	.276	.165	.008	.756
	2.5	.289	.162	.018	.756
	3.0	.306	.158	.024	.756
MEA	1.5	.354	.188	.054	.692
	2.0	.366	.182	.078	.692
	2.5	.379	.178	.080	.692
	3.0	.396	.174	.108	.692

Table 43 presents the performance rate of each method by the distributional characteristic of the latent variables. FMM and QFA were most impacted by the distribution of the data. The ability of FMM to detect latent classes declined by more than 50% when the data were non-normal and the performance of QFA declined by more than 26%. The other two methods, MEA and MDM, were not as influenced by the distributional characteristic. Although the normal design condition tended to perform better the difference was not significant.

Table 43
Performance Rate by Distributional Characteristic

		<i>M</i>	<i>SD</i>	<i>Min</i>	<i>Max</i>
FMM	Normal	.808	.248	.204	1.000
	Non-Normal	.316	.242	.000	.804
QFA	Normal	.336	.161	.128	.664
	Non-Normal	.266	.211	.012	.796
MDM	Normal	.287	.141	.040	.644
	Non-Normal	.279	.186	.004	.756
MEA	Normal	.390	.198	.054	.692
	Non-Normal	.358	.162	.086	.622

Table 44 shows the performance of each method by model complexity. FMM performed best with Model 2c. Although it appears counterintuitive that FMM performed best with the most complex model, the variance-covariance matrix shows the cross-loadings in Model 2c reduced the amount of variation in the cross-loaded observed variables. A comparison of Model 2 and Model 2c variance-covariance matrices makes this clear. Without considering the cross-loaded model, the ability of FMM to detect latent classes diminished as the model became more complex. The performance of MEA did not vary significantly by model. The range of performance was only .013. QFA performed best with Model 1, the least complex model and the performance was negatively impacted as the model became more complex. Finally, the performance of MDM did not vary significantly by model, although the performance rate tended to increase slightly as the model became more complex. However, the difference between the model with best and worst performances was only 4.5%.

Table 44
Performance Rate by Model Complexity

		<i>M</i>	<i>SD</i>	<i>Min</i>	<i>Max</i>
FMM	Model 1	.600	.356	.008	1.000
	Model 2	.523	.342	.000	1.000
	Model 2c	.632	.324	.022	1.000
	Model 3	.491	.351	.000	1.000
QFA	Model 1	.363	.123	.200	.578
	Model 2	.306	.229	.064	.796
	Model 2c	.328	.208	.078	.796
	Model 3	.207	.147	.012	.470
MDM	Model 1	.277	.135	.036	.554
	Model 2	.285	.171	.004	.684
	Model 2c	.281	.152	.004	.610
	Model 3	.290	.196	.006	.756
MEA	Model 1	.372	.179	.062	.692
	Model 2	.377	.183	.074	.680
	Model 2c	.380	.185	.076	.680
	Model 3	.367	.179	.054	.670

Effect Sizes

To summarize the results, Table 45 indicates FMM was most impacted by the distributional characteristic of the variables and the number of classes. QFA was most impacted by the number of classes and the model. MDM was most impacted by the number of classes and the sample size. MEA was most impacted by the number of classes. The largest effects out of all the design conditions were distributional characteristic and number of classes on FMM. The smallest effects were mixing proportion on QFA and distributional characteristic on MDM.

Table 45
Effect Sizes of Manipulated Design Variables

Design Variable	FMM	QFA	MDM	MEA
Model	0.285	0.348*	0.027	0.028
Sample Size	0.450*	0.145	0.381*	0.203
Number of Classes	0.947**	0.795**	0.749**	0.874**
Mixing Proportion	0.09	0.006	0.036	0.042
Distributional Characteristic	1.041**	0.152	0.018	0.065
Class Separation	0.219	0.047	0.1	0.088

Notes. *Moderate effect size. **Large effect size.

Chapter 6: Discussion

This dissertation intended to answer questions regarding the tenability of QFA, MDM, and MEA in detecting latent classes in data from heterogeneous populations. Data were simulated by manipulating several design variables: sample size, number of latent classes, mixing proportion, class separation, distributional assumption, and model complexity. The performance of each method was assessed for each design condition by finding the proportion of generated data sets for which the method was able to identify the true number of latent classes. Overall, the results show that although FMM performed best, the other methods are viable in successfully identifying latent classes.

The results for FMM were consistent with what other researchers have established in mixture model simulation studies. When data were non-normal, FMM performance was often quite poor, especially when the number of classes increased. In regards to the class separation and mixing proportion, prior literature (Lubke & Neale, 2008; Yung, 1997) showed a small multivariate distance (i.e., $MD = 1.5$) can be used when classes are of equal size. However, a larger value of MD was needed when the class sizes are not of equal size. A simulation study by Lubke & Tueller (2010) showed that $MD > 2.0$ was not problematic for FMM if class sizes are equal. The study also showed $MD > 3.0$ was a large effect for FMM. Under the $MD = 3.0$ data condition FMM consistently detected the correct number of latent classes. Distributional characteristics have also been studied. Bauer & Curran (2003) and McLachlan & Peel (2000) found that non-normal continuous measures can lead to estimation of spurious latent classes. None of the results for FMM were contrary to what was already established by other mixture model researchers.

QFA has potential as a viable method in detecting latent classes. Some researchers have been quite critical of using the method in determining clusters in data. One issue in particular concerning this dissertation is the ratio of individuals to vari-

ables. Gorsuch (1983) and Cattell (1978) suggested that the use of the factor analysis method is justified only when the number of columns is considerably fewer than the number of rows. In particular to a Q-data matrix, the number of individuals should be considerably fewer than the number of variables. Otherwise, the stability of the solution is compromised (Gorsuch, 1983).

There are two categories of general recommendations involving the amount of data needed to carry out a factor analysis: the absolute number of individuals (i.e., rows of data) and the individual-to-variable ratio (variable-to-individual ratio for QFA). Although researchers do not agree on the minimum desirable level of sample size or the ratio to use, there are several recommendations. The minimum desirable number of rows ranges from 100 to 500. The maximum number of rows this dissertation considered was only 15. Furthermore, the individual-to-variable recommended ratio ranges from 2 : 1 (2 rows for every 1 column of data) to 20 : 1 (20 rows for every 1 column of data). In this dissertation, the number of columns always exceeded the number of rows. The dimensionality of the data used in this dissertation do not fall within the recommended minimum sample size or ratio. However, the results do not show that QFA was significantly influenced by the sample size.

MDM is a promising method, as well. The overarching concern is the lack of rigor involved in counting the number of classes in the data. The original method of determining the number of latent classes involved examining the contribution of each Δ_i and selecting the first k Δ 's that summed to the contribution desired. For example, if a researcher was interested in accounting for 90% of the gaps, then the researcher would select the first k Δ 's that summed to 90%. This approach to counting the number of classes has shown to be viable if a researcher was using the method on a small number of data sets where each Δ vector could be examined and then a determination of the number of classes made. However, if the researcher was interested in performing the method on a large number of data sets, the original method might not be feasible.

A pilot study was conducted to examine using cut-off values (i.e., 80% – 95%). The performance of the method was sub-optimal when this approach was taken. Some of the design variables appeared to have a significant influence on the structure of the S vector and the resulting Δ vector. For example, under some design conditions for a two-class model the first k Δ 's might sum to 90%. Yet under other design conditions for a two-class model the first k Δ 's might only sum to 70%. Therefore, using a global cut-off value did not produce optimal results.

This dissertation attempted to find a more automated approach that was still in the same spirit as originally developed and intended by Marcoulides & Drezner (2000) and that could accommodate the different design conditions. Although the method developed to count the number of classes has potential, one of the concerns is the lack of a stopping rule for a one-class model. In particular, the automated approach does not check to determine if a one-class model is suitable for the data. Instead, the approach detected if a one-class approach was suitable and if not, then the solution defaulted to a one-class model. Potentially, this strategy had an impact on how well the method counted the number of classes. Using a different formulation of the automated approach might show a different pattern of results.

MEA showed great potential. Unfortunately, MEA suffers from a similar drawback as MDM. The strategy in counting the number of classes need more rigor. To count the classes, the method determines where the eigenvector begins to level out or where the largest eigengap exists. In some test cases, it was observed that the largest gap was only slightly larger than other gaps. Effectively, there could be ties where two or more eigengaps are almost equal in size. Ideally, if such a scenario appears the method should evaluate the eigengaps based on a second criteria. For example, if the difference between any two gaps is $\leq |.01|$, then a second criteria is used to determine the best eigengap to use. The second criteria might be based on the position of the eigengap relative to the others. First example, perhaps the first large gap might prove to be most effective

in dividing the data into classes. This additional deciding criteria might provide more rigor to the process of counting the number of latent classes and perhaps provide better insight to the ability of MEA.

A researcher can be guided by the results of this dissertation in several ways. The complexity of the model was most important in QFA and FMM. As the model complexity increased the performance ability of the method decreased. MEA and MDM were not influenced by the complexity of the model. Generally, with all the methods, the sample size and performance had a positive relationship. FMM and MDM were most impacted by the sample size. The number of classes had a large influence on all the methods. As the number of classes increased the performance decreased. Interesting, the mixing proportion did not play as large of a role as expected. Although the balanced design level tended to perform best the performance difference was not significant. The distributional characteristics was only a concern with the FMM method. In fact, whether the data were normal or non-normal was the largest effect of FMM. Finally, the class separation had the largest influence on FMM. Of course, as the class separation increased the performance increased. Overall, the spectral clustering methods did not appear to be as influenced by characteristics of the data, as compared to FMM and QFA. This is clearly an advantage of the spectral clustering methods over FMM.

Limitations of the Study

Several limitations to this dissertation should be noted. First, a major limitation of all simulation studies is the failure of any study to cover all possibilities that exist in real world data analysis. In the present study, the simulations only considered a select number of variables in the design. The dissertation was not meant to cover all possible scenarios that a researcher might encounter; instead it provided information on the most common situations.

Furthermore, some of the levels of the manipulated variables were quite limited. Regarding normality, the data simulated were either almost perfectly normal or highly non-normal. It is quite evident from the results that the level of normality played a significant role in the ability of FMM to detect the correct number of classes. Therefore, other levels of normality should be tested so that a determination can be made at what point non-normality is detrimental to the performance of the method. The same applies to the mixing proportion; only two levels was tested. Perhaps testing more extreme levels will provide additional information to the ability of each method when the class sizes are more imbalanced.

Another limitation involves assigning individuals to a latent class. Analysis of heterogeneous data typically involves finding the number of populations present and then assigning individuals to the appropriate latent class. This dissertation only considered the identification of the number of latent classes. For example, suppose two latent classes exists, the first class contains individuals 1 – 5 and the second class contains individuals 6 – 10. The present study only assessed whether two latent classes were identified. The error rate of misclassified individuals is not taken into account. This is valuable information for a researcher, especially if a real world analysis is being conducted for the purpose of categorizing individuals.

Finally, overextraction and underextraction was not considered for this dissertation. If a method did not select the correct number of classes, information as to whether the method tended to overextract or underextract the number of classes was not systematically studied. Whether a method overextracts or underextracts latent classes is valuable information to the researcher that should be considered more thoroughly.

Suggestions for Future Research

There are several important directions for future research that might be of interest to quantitative mixture model researchers. As previously mentioned MDM and MEA have many advantages over FMM. In particular, the methods are not highly parameterized and the results show the performance of these methods do not vary widely across design conditions. Further development of these methods would include developing a strategy to classify individuals. The classification of individuals is typically the ultimate goal of a researcher performing a classification analysis. Improving the current strategies in counting the number of classes is an important avenue for future research, as well.

Finally, as previously mentioned the MEA method is capable of easily incorporating covariates and missing data, although covariates and missing data were not included as design conditions in this dissertation. Comparing FMM and MEA on these other potential design factors will expand the spectral clustering literature, in particular. Currently there are very few research studies that examine the impact of missing data on spectral clustering methods and there are no research studies that incorporate covariates.

References

- Akaike, H. (1981). Likelihood of a model and information criteria. *Journal of Econometrics*, *16*(1), 3–14.
- Akaike, H. (1987). Factor analysis and AIC. *Psychometrika*, *52*(3), 317–332.
- Arminger, G., Stein, P., & Wittenberg, J. (1999). Mixtures of conditional mean-and covariance-structure models. *Psychometrika*, *64*(4), 475–494.
- Bauer, D. J., & Curran, P. J. (2003). Distributional assumptions of growth mixture models: Implications for overextraction of latent trajectory classes. *Psychological Methods*, *8*(3), 338–363.
- Bauer, D. J., & Curran, P. J. (2004). The integration of continuous and discrete latent variable models: Potential problems and promising opportunities. *Psychological Methods*, *9*(1), 3–29.
- Bollen, K. A. (2002). Latent variables in psychology and the social sciences. *Annual Review of Psychology*, *53*(1), 605–634.
- Brown, T. A., O’Leary, T. A., & Barlow, D. H. (2001). Generalized anxiety disorder. In D. Barlow (Ed.), *Clinical handbook of psychological disorders: A step-by-step treatment manual* (pp. 137–189). New York, NY: Guilford Press.
- Burger, G. K. (2004). Using person factors in social science research. *Interdisciplinaria*, *99*, 35–48.
- Cattell, R. B. (1978). *The scientific use of factor analysis in behavioral and life sciences*. New York, NY: Plenum Press.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Hillsdale, NJ: Routledge Academic.

- Conor, D., & Maas, H. (1998). Fitting multivariate normal finite mixtures subject to structural equation modeling. *Psychometrika*, *63*(3), 227–253.
- Drezner, Z. (1987). A heuristic procedure for the layout of a large number of facilities. *Management Science*, *33*(7), 907–915.
- Fabrigar, L. R., Wegener, D. T., MacCallum, R. C., & Strahan, E. J. (1999). Evaluating the use of exploratory factor analysis in psychological research. *Psychological Methods*, *4*(3), 272–299.
- Fan, X., Sivo, S., & Keenan, S. (2002). *SAS for Monte Carlo studies: A guide for quantitative researchers*. Cary, NC: SAS Institute.
- Fleishman, A. I. (1978). A method for simulating non-normal distributions. *Psychometrika*, *43*(4), 521–532.
- Gagné, P. (2004). *Generalized confirmatory factor mixture modeling: A tool for assessing factorial invariance across unspecified populations*. (Doctoral dissertation). Retrieved from ProQuest Digital Dissertations.
- Gagné, P. (2006). Mean and covariance structure mixture models. In G. R. Hancock, & R. O. Mueller (Eds.), *Structural equation modeling: A second course* (pp. 197–224). Greenwich, CT: Information Age.
- Gilles, C., & Gilda, S. (1996). An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification*, *13*(2), 195–212.
- Gorsuch, R. L. (1983). *Factor analysis*. Hillsdale, NJ: Lawrence Erlbaum.
- Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, *27*(4), 857–871.

- Green, S., Levy, R., Thompson, M., Lu, M., & Lo, W. (2012). A proposed solution to the problem with using completely random data to assess the number of factors with parallel analysis. *Educational and Psychological Measurement*, *72*(3), 357–374.
- Hagenaars, J., & McCutcheon, A. (2002). *Applied latent class analysis*. New York, NY: Cambridge University Press.
- Hatcher, L. (2007). *A step-by-step approach to using SAS for factor analysis and structural equation modeling*. Cary, NC: SAS Institute.
- Jedidi, K., Jagpal, H. S., & DeSarbo, W. S. (1997). Finite-mixture structural equation models for response-based segmentation and unobserved heterogeneity. *Marketing Science*, *16*, 39–59.
- Jöreskog, K. G. (1971). Simultaneous factor analysis in several populations. *Psychometrika*, *36*(4), 409–426.
- Kaiser, H. F., & Dickman, K. (1962). Sample and population score matrices and sample correlation matrices from an arbitrary population correlation matrix. *Psychometrika*, *27*(2), 179–182.
- Lazarsfeld, P. F., & Henry, N. W. (1968). *Latent structure analysis*. Boston, MA: Houghton Mifflin.
- Li, K., & Liu, Y. S. (2007). A spectral clustering algorithm based on self-adaption. In *Machine Learning and Cybernetics, 2007 International Conference on* (Vol. 7, pp. 3965–3968).
- Lubke, G., & Muthén, B. O. (2007). Performance of factor mixture models as a function of model size, covariate effects, and class-specific parameters. *Structural Equation Modeling*, *14*(1), 26–47.

- Lubke, G., & Neale, M. (2008). Distinguishing between latent classes and continuous factors with categorical outcomes: Class invariance of parameters of factor mixture models. *Multivariate Behavioral Research*, *43*(4), 592–620.
- Lubke, G., & Tueller, S. (2010). Latent class detection and class assignment: A comparison of the MAXEIG-taxometric procedure and factor mixture modeling approaches. *Structural Equation Modeling*, *17*(4), 605–628.
- Lubke, G. H., Dolan, C. V., Kelderman, H., & Mellenbergh, G. J. (2003). On the relationship between sources of within-and between-group differences and measurement invariance in the common factor model. *Intelligence*, *31*(6), 543–566.
- Lubke, G. H., & Muthén, B. O. (2005). Investigating population heterogeneity with factor mixture models. *Psychological Methods*, *10*(1), 21–39.
- MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam, & J. Neyman (Eds.), *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281–297). Berkeley, CA: University of California Press.
- Magidson, J., & Vermunt, J. K. (2002). Latent class models for clustering: A comparison with k-means. *Canadian Journal of Marketing Research*, *20*(1), 36–43.
- Magidson, J., & Vermunt, J. K. (2004). Latent class models. In D. Kaplan (Ed.), *The SAGE handbook of quantitative methodology for the social sciences* (pp. 175–198). Thousand Oaks, CA: SAGE Publications.
- Marcoulides, G., & Drezner, Z. (2000). A procedure for detecting pattern clustering in measurement designs. In M. Wilson, & G. Engelhard Jr. (Eds.), *Objective measurement: Theory into practice* (Vol. 5, pp. 261–277). Norwood, NJ: Ablex.
- McLachlan, G., & Peel, D. (2000). *Finite mixture models*. New York, NY: John Wiley & Sons.

- Micceri, T. (1989). The unicorn, the normal curve, and other improbable creatures. *Psychological Bulletin*, *105*, 156–166.
- Muthén, B. O. (1989). Latent variable modeling in heterogeneous populations. *Psychometrika*, *54*(4), 557–585.
- Muthén, B. O., & Shedden, K. (1999). Finite mixture modeling with mixture outcomes using the EM algorithm. *Biometrics*, *55*(2), 463–469.
- Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In T. G. Dietterick, S. Becker, & Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems* (Vol. 2, pp. 849–856). Cambridge, MA: MIT Press.
- Nylund, K. L., Asparouhov, T., & Muthén, B. O. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study. *Structural Equation Modeling*, *14*(4), 535–569.
- O'Connor, B. P. (2000). SPSS and SAS programs for determining the number of components using parallel analysis and Velicer's MAP test. *Behavior Research Methods, Instruments, & Computers*, *32*(3), 396–402.
- Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, *185*, 71–110.
- Pearson, K. (1895). Contributions to the mathematical theory of evolution. II. Skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London. A*, *186*, 343–414.
- Raykov, T., & Marcoulides, G. (2008). *An introduction to applied multivariate analysis*. New York, NY: Routledge Academic.

- Rencher, A. C., & Christensen, W. F. (2012). *Methods of multivariate analysis*. Hoboken, NJ: John Wiley & Sons.
- Rindskopf, D. (2003). Mixture or homogeneous? Comment on Bauer and Curran (2003). *Psychological Methods*, *8*(3), 364–368.
- Roeder, K., & Wasserman, L. (1997). Practical Bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, *92*(439), 894–902.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.
- Sclove, S. L. (1987). Application of model-selection criteria to some problems in multivariate analysis. *Psychometrika*, *52*(3), 333–343.
- Sörbom, D. (1974). A general method for studying differences in factor means and factor structure between groups. *British Journal of Mathematical and Statistical Psychology*, *27*, 229–239.
- Söromenho, G. (1994). Comparing approaches for testing the number of components in a finite mixture model. *Computational Statistics*, *9*(1), 65.
- Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, *15*, 72–101.
- Stephenson, W. (1935). Technique of factor analysis. *Nature*, *136*, 297.
- Stevens, J. P. (2012). *Applied multivariate statistics for the social sciences*. New York, NY: Routledge Academic.
- Thompson, G. H. (1935). On complete families of correlation coefficients, and their tendency to zero tetrad differences: Including a statement of the sampling theory of abilities. *British Journal of Psychology*, *26*(1), 63–92.

- Tofighi, D., & Enders, C. K. (2008). Identifying the correct number of classes in a growth mixture. In G. R. Hancock, & K. M. Samuelson (Eds.), *Advances in latent variable mixture models* (pp. 317–341). Greenwich, CT: Information Age.
- Vale, C., & Maurelli, V. (1983). Simulating multivariate nonnormal distributions. *Psychometrika*, *48*(3), 465–471.
- Verma, D., & Meila, M. (2003). A comparison of spectral clustering algorithms.
- Vermunt, J. K., & Magidson, J. (2002). Latent class cluster analysis. In J. A. Hagenaars, & A. L. McCutcheon (Eds.), *Applied latent class analysis* (pp. 283–297). New York, NY: Cambridge University Press.
- Waller, N. G., & Meehl, P. E. (1998). *Multivariate taxometric procedures: Distinguishing types from continua*. Thousand Oaks, CA: Sage.
- Weldon, W. F. R. (1892). Certain correlated variations in crangon vulgaris. *Proceedings of the Royal Society of London*, *51*, 1–21.
- Yang, C. C. (2006). Evaluating latent class analysis models in qualitative phenotype identification. *Computational Statistics & Data Analysis*, *50*(4), 1090–1104.
- Yung, Y. F. (1997). Finite mixtures in confirmatory factor-analysis models. *Psychometrika*, *62*(3), 297–330.
- Zwick, W. R., & Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. *Psychological Bulletin*, *99*(3), 432–442.

Appendix A

The following SAS code was used to generate the data. The first macro created the factor scores. The second macro created the observed data.

```
*****
* THIS MACRO IS USED TO GENERATE DATA FOR MODEL 2 AND 2 CLASSES USING
* THE FOLLOWING VARIABLE SPECIFICATIONS:
*
* B1, C1, C1 = FLEISHMANS COEFFICIENTS;
* SAMPSIZE = THREE LEVELS OF THE SAMPLE SIZE;
* MD = FOUR LEVELS OF MAHALANOBIS DISTANCE;
* MEAN = VALUE OF MEAN VECTOR NEEDED FOR SPECIFIED MD;
* K = FOUR LEVELS OF NUMBER OF CLASSES;
* MIX = TWO LEVELS OF MIXING PROPORTION;
* I_CORR = INTERMEDIATE CORRELATION MATRIX;
*****

%MACRO DATA2(VARLIST=ETA1 ETA2 GROUP,  $\mathbf{V}$ = X1 X2 X3 X4 X5 X6 X7 X8 X9
X10 GROUP, CLASS=K, SPACE=R,
B1=0.832216343, C1=0.128396714, D1=0.048032053);

%DO SAMPSIZE=1 %TO 3;
%IF &SAMPSIZE=1 %THEN %DO; %LET LET=A; %LET N=180; %END;
%IF &SAMPSIZE=2 %THEN %DO; %LET LET=B; %LET N=540; %END;
%IF &SAMPSIZE=3 %THEN %DO; %LET LET=C; %LET N=1080; %END;

%DO MD=1 %TO 4;
%IF &MD=1 %THEN %DO; %LET MEAN=1.096; %LET DIST=15; %END;
%IF &MD=2 %THEN %DO; %LET MEAN=1.265; %LET DIST=20; %END;
%IF &MD=3 %THEN %DO; %LET MEAN=1.414; %LET DIST=25; %END;
%IF &MD=4 %THEN %DO; %LET MEAN=1.550; %LET DIST=30; %END;

%DO K=1 %TO 4;

%DO MIX=1 %TO 2;
%IF &MIX=1 %THEN %DO; %LET SS=&N; %LET SS1=&N/&K; %LET SS2=&N/&K;
%LET SS3=&N/&K; %LET SS4=&N/&K; %LET MP=E; %END;
%IF &MIX=2 %THEN %DO; %LET SS=&N; %LET SS1=&N*.7; %LET SS2=&N*(.3/(&K-1));
%LET SS3=&N*(.3/(&K-1)); %LET SS4=&N*(.3/(&K-1)); %LET MP=L; %END;

%DO GAUSS=1 %TO 2;

%LET I=M2&LET&DIST&MP&K&CLASS&GAUSS;

DATA _NULL_;
X MKDIR "E:\DATA\M2DATA\&I";
RUN;

%IF &K=2 %THEN %DO;
%DO J=1 %TO 500;

PROC IML;
USE PATTERN;
READ ALL VAR _NUM_ INTO F;
F=F';

DATA=RANNOR(J(&SS1,2,0));
DATA=DATA';
Z1 = F*DATA;
Z1 = Z1';

ETA1 = &C1*(-1)+ &B1*Z1[,1] + &C1*Z1[,1]##2 + &D1*Z1[,1]##3;
ETA2 = &C1*(-1)+ &B1*Z1[,2] + &C1*Z1[,2]##2 + &D1*Z1[,2]##3;
Z1=ETA1||ETA2;

GROUP=J(&SS1,1,1);
Z1=Z1||GROUP;

CREATE SIMULATE FROM Z1 [COLNAME={&VARLIST}];
APPEND FROM Z1;

%INCLUDE "C:\USERS\SR MCDONALD\DROPBOX\DISSERTATION DONE\LATENT DATA GENERATION\MANIFESTGENERATOR2.SAS";

DATA G1&I&SPACE&J;
SET SIMDAT;
PUT  $\mathbf{V}$ ;
RUN;
```

```

%END;

%DO J=1 %TO 500;

PROC IML;
USE PATTERN;
READ ALL VAR _NUM_ INTO F;
F=F';

DATA=RANNOR(J(&SS2,2,0));
DATA=DATA';
Z2 = F*DATA;
Z2 = Z2';

ETA1 = &C1*(-1)+ &B1*Z2[,1] + &C1*Z2[,1]##2 + &D1*Z2[,1]##3;
ETA2 = &C1*(-1)+ &B1*Z2[,2] + &C1*Z2[,2]##2 + &D1*Z2[,2]##3;
Z2=ETA1||ETA2;

ADD=J(&SS2,1,&MEAN);
Z2=Z2+ADD*1;

GROUP=J(&SS2,1,2);
Z2=Z2|GROUP;

CREATE SIMULATE FROM Z2 [COLNAME={&VARLIST}];
APPEND FROM Z2;

%INCLUDE "C:\USERS\SR MCDONALD\DROPBOX\DISSERTATION DONE\LATENT DATA GENERATION\MANIFESTGENERATOR2.SAS";

%END;
%END;
%END;
%END;
%END;

%MEND DATA2;

DATA I_CORR (TYPE=CORR);
_TYPE_='CORR';
INPUT K1 K2;
DATALINES;
1 .
0.6130263344 1;
RUN;

PROC FACTOR N=2 OUTSTAT=FACOUT;
RUN;

DATA PATTERN;
SET FACOUT;
IF _TYPE_='PATTERN';
DROP _TYPE_ _NAME_;
RUN;

%DATA2

*****
* THIS FILE IS CALLED FROM THE MAIN MACRO TO CREATE THE OBSERVED DATA;
* LAM = FACTOR LOADINGS;
* NU = INTERCEPTS;
* EPSILON = ERRORS;
* ETA = FACTOR SCORES;
*****

DATA SIMDAT;
SET SIMULATE;
N1=RANNOR(0); N2=RANNOR(0); N3=RANNOR(0); N4=RANNOR(0); N5=RANNOR(0);
N6=RANNOR(0); N7=RANNOR(0); N8=RANNOR(0); N9=RANNOR(0); N10=RANNOR(0);

LAM1=.8;

NU1=0; NU2=0; NU3=0; NU4=0; NU5=0;
NU6=0; NU7=0; NU8=0; NU9=0; NU10=0;

EPSILON1 = SQRT(.36)*N1;
EPSILON2 = SQRT(.36)*N2;
EPSILON3 = SQRT(.36)*N3;
EPSILON4 = SQRT(.36)*N4;
EPSILON5 = SQRT(.36)*N5;
EPSILON6 = SQRT(.36)*N6;
EPSILON7 = SQRT(.36)*N7;
EPSILON8 = SQRT(.36)*N8;
EPSILON9 = SQRT(.36)*N9;

```

```
EPSILON10 = SQRT(.36)*N10;  
  
X1 = NU1 + LAM1*ETA1 + EPSILON1;  
X2 = NU2 + LAM1*ETA1 + EPSILON2;  
X3 = NU3 + LAM1*ETA1 + EPSILON3;  
X4 = NU4 + LAM1*ETA1 + EPSILON4;  
X5 = NU5 + LAM1*ETA1 + EPSILON5;  
X6 = NU6 + LAM1*ETA2 + EPSILON6;  
X7 = NU7 + LAM1*ETA2 + EPSILON7;  
X8 = NU8 + LAM1*ETA2 + EPSILON8;  
X9 = NU9 + LAM1*ETA2 + EPSILON9;  
X10 = NU10 + LAM1*ETA2 + EPSILON10;  
  
KEEP X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 GROUP;  
RUN;
```

Appendix B

The following *Mplus* code was used to obtain the FMM results.

```
Title:
M2A15E3K2T3.inp

Data:
File="e:/FMM Data/M2DataF/M2A15E3K2/ListFile.txt";
Type=Montecarlo;

Variable:
Names are X1-X10 Group;
Usevariables X1-X10;
Classes=c(3);

Analysis:
Type=mixture;
Starts 300 30;

Model:
%Overall%
F1 by X1* X2 X3 X4 X5;
F2 by X6* X7 X8 X9 X10;
F1@1;
F2@1;
F1 with F2;
[X1-X10@0];

Savedata:
Results is "e:/FMM Results/FMM2/Output/M2A15E3K2T3.txt";

Output:
Tech9;
```

Appendix C

The following SAS code was used to obtain the QFA results.

```
FILENAME DR "E:\FMM DATA\&DATA\&MODEL\&LSTNAME";
DATA QFA;
INFILE DR;
INPUT &VARLIST;
DROP GROUP;
RUN;

PROC STDIZE DATA=QFA OUT=XZ;
RUN;

PROC TRANSPOSE DATA=XZ PREFIX=X OUT=XZT (DROP=_NAME_);
RUN;

PROC STDIZE DATA=XZT OUT=XZTT;
RUN;

PROC FACTOR DATA=XZTT PRIORS=MAX METHOD=PRINCIPAL ROTATE=PROMAX
OUTSTAT=EIGENS NOPRINT NOCORR;
RUN;

DATA EIGENS;
SET EIGENS;
WHERE _TYPE_='EIGENVAL';
DROP _TYPE_ _NAME_;
RUN;

PROC TRANSPOSE DATA=EIGENS OUT=EIGENS (DROP=_NAME_);
RUN;
```

Appendix D

The following SAS code was used to obtain the MDM results.

```
FILENAME DR "E:\FMM DATA\&DATA\&MODEL\&LSTNAME";
DATA MDM;
INFILE DR;
INPUT &VARLIST;
DROP GROUP;
RUN; QUIT;

PROC STDIZE DATA=MDM METHOD=STD OUT=STAND;
RUN;QUIT;

PROC DISTANCE DATA=STAND PREFIX=X METHOD=EUCLID SHAPE=SQUARE OUT=DISTANCE;
VAR INTERVAL (&VARLIST2);
RUN;QUIT;

DATA RDISTANCE;
SET DISTANCE;
ARRAY RDISTANCE {*} _NUMERIC_;
DO I=1 TO DIM(RDISTANCE);
IF RDISTANCE{I} > 0 THEN RDISTANCE{I}=1/RDISTANCE{I};
END;
DROP I;
RUN;QUIT;

PROC IML;
USE RDISTANCE;
READ ALL INTO OFFDIAG;
COLSUM=OFFDIAG[+,];
DIAGONAL=DIAG(COLSUM);
DI=INV(DIAGONAL);
SMATRIX=DIAGONAL-OFFDIAG;
 $\mathbf{V} = \text{EIGVEC}(\text{SMATRIX});$ 
EV=EIGVAL(SMATRIX);
SECVEC= $\mathbf{V}$ [,NCOL( $\mathbf{V}$ )-2];
VMIN=MIN(SECVEC);
VMAX=MAX(SECVEC);
S=(SECVEC-VMIN)/(VMAX-VMIN);
CALL SORT(S,1);
DELTA=J(NROW(S),1,0);
N=NROW(S)-1;
DO I=1 TO N;
DELTA[I,1]=S[I+1,1]-S[I,1];
END;
CALL SORT(DELTA,1,1);
CUMDELTA=J(NROW(S),1,-333);
CUM=0;

DO I=1 TO N;
CUMDELTA[I,1]=CUM+DELTA[I,1];
CUM=CUM+DELTA[I,1];
END;
M=NROW(S)-1;
CUMDELTA_DIFF=J(NROW(S),1,0000);
DO I=1 TO M;
CUMDELTA_DIFF[I,1]=CUMDELTA[I+1,1]-CUMDELTA[I,1];
END;

CREATE CUMDELTA_DIFF FROM CUMDELTA_DIFF;
APPEND FROM CUMDELTA_DIFF;

DATA SUCCESS;
ID+1;
SET CUMDELTA_DIFF;
RENAME COL1=DATA;
RUN; QUIT;

DATA SUCCESS;
SET SUCCESS;
DIFF=DATA-0.05;
IF DIFF>0 THEN COUNT=1;
ELSE COUNT=0;
RUN; QUIT;
```

Appendix E

The following SAS code was used to obtain the MEA results.

```
FILENAME DR "E:\FMM DATA\&DATA\&MODEL\&LSTNAME";
DATA MEA;
INFILE DR;
INPUT &VARLIST;
DROP GROUP;
RUN; QUIT;

PROC DISTANCE DATA=MEA PREFIX=X METHOD=DGOWER SHAPE=SQUARE OUT=DISTANCE;
VAR INTERVAL (&VARLIST2);
RUN; QUIT;

DATA RDISTANCE;
SET DISTANCE;
ARRAY RDISTANCE {*} _NUMERIC_;
DO I=1 TO DIM(RDISTANCE);
IF RDISTANCE{I} > 0 THEN RDISTANCE{I}=RDISTANCE{I};
END;
DROP I;
RUN; QUIT;

PROC IML;
USE RDISTANCE;
READ ALL INTO OFFDIAG;
COLSUM=OFFDIAG[+,];
DIAGONAL=DIAG(COLSUM);
DI=INV(DIAGONAL);
SMATRIX=DIAGONAL-OFFDIAG;
EV=EIGVAL(SMATRIX);
CALL SORT(EV,1,1);
DELTA=J(NROW(EV),1,0);
N=NROW(EV)-1;
DO I=1 TO N;
DELTA[I,1]=EV[I,1]-EV[I+1,1];
END;

CREATE DELTA FROM DELTA;
APPEND FROM DELTA;
```