ABSTRACT

Title of dissertation:	ASSESSING FIT OF LATENT CLASS MODELS TO COMPLEX SURVEY DATA: IMPLICATIONS FOR DRUG USE RESEARCH
	Carrie Elizabeth Markovitz, Doctor of Philosophy, 2003
Dissertation directed by:	Professor C. Mitchell Dayton Department of Measurement, Statistics and Evaluation

Simple random sampling is an assumption when using fit statistics to fit latent class (LC) models to data. However, LC models are often fit to datasets collected through complex survey sampling methods that may result in inaccurate estimates of standard errors, parameter estimates and fit statistics. This study examined how various comparison tests functioned for latent class models when using complex survey data. The motivation for this research is the issue of reported drug use patterns and whether changes in drug use have occurred over time. This issue was investigated using reported drug use data from the National Household Survey on Drug Abuse (NHSDA) for 1979 and 1988. Monte Carlo simulations were used to determine how well the various model comparison statistics (chi-square, AIC, BIC, RIC and Wald statistic) functioned for a variety of complex sample designs. In addition, a simulation based on the NHSDA data was used to answer the research question: Do patterns of reported drug use show change over time? The model comparison statistics were most accurate when sample sizes were large and item-specific error rates were low. Intraclass correlation, an indicator of how similar

individuals are within the same cluster, appeared to have little effect on the accuracy of the model comparison statistics. Statistics were not as accurate when sampling from unequally weighted groups. The chi-square statistics and AIC were recommended for use with complex survey data based on their high rates of accuracy. More caution was recommended when using BIC and RIC. Results indicated that reported drug use patterns changed between 1979 and 1988. Most patterns of reported drug use increased slightly, with the exception of respondents characterized by alcohol and tobacco use alone that decreased substantially.

ASSESSING FIT OF LATENT CLASS MODELS TO COMPLEX SURVEY DATA: IMPLICATIONS FOR DRUG USE RESEARCH

By

Carrie Elizabeth Markovitz

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2003

Advisory Committee:

Professor C. Mitchell Dayton, Chair
Professor David Cantor
Professor Greg Hancock
Professor George Macready
Professor Robert Mislevy

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

PURPOSE AND RATIONALE

Latent class analysis (LCA) is a method of empirically identifying categorical latent underlying variables by observing discrete manifest variables. Latent class (LC) models allow the identification of a set of exclusive and exhaustive latent classes that account for the distribution of respondents in a population. Therefore, LCA can be used for testing whether a theoretical structure adequately fits a dataset. LC models have been fit to datasets collected through complex survey sampling methods. An important property of complex surveys is that the probability of selection may not be equal for all members of the population, so that sampling weights must be used to correct for this inequality. Also, it is frequently the case that clusters of respondents are selected within groups. This impacts the estimation of standard errors since respondents within clusters tend to be positively correlated.

Due to the use of complex survey sampling, comparing latent structures across two or more periods of time or across independent groups, as in simultaneous latent structure analysis, presents problems. For example, it may be difficult to determine whether a homogenous, partial-homogeneous, or heterogeneous model best fits the data since the usual LC assumptions of simple random sampling (SRS) are not met. To date, no study has examined how model comparison statistics function for LC models when using complex survey data.

The motivation for this research was provided by the issue of reported drug use patterns among members of households located in the United States. It has been theorized that drug use occurs in stages that are specifically ordered. If this pattern of drug use exists, then respondents can be organized into a set number of classes representing different drug use states at various points in time. Furthermore, respondents should form mutually exclusive and exhaustive classes with class membership dependent on the history of drug use.

Individuals can be organized into groups or classes depending on their sequence of reported drug use because the types of drugs used, in theory, form a linear relation. Latent class scaling is a form of latent class analysis that uses linear relations to categorize respondents with respect to some underlying, latent variable. If it is theorized that a lower order drug must be used before a higher order drug, then reported drug use should form a Guttman-like scale, organized around each drug's level. In the context of a Guttman scale, a person's experience with a drug is dependent on that individual's experience with lower order drugs. In this case, latent class scaling can be used to study the research question: *Do patterns of drug use show change over time?*

The question of drug use over time was examined using responses to questions from the National Household Survey on Drug Abuse (NHSDA) for the years 1979 and 1988. The data for these two years were compared using a method developed by Clogg and Goodman (1984, 1985) referred to as simultaneous latent structure analysis. The simultaneous analysis of latent structures involves the comparison of latent structures across multiple groups or years.

Drug use data, such as the NHSDA, are often collected using complex survey sampling methods involving multistage sampling and sampling weights. Lee, Forthofer and Lorimor (1989) note that statistics routinely used for making inferences about population parameters will not produce accurate estimates when applied to data from complex survey designs. Typical statistics assume SRS and, when this assumption is not met, a potential source of error is introduced into the analysis. Lehtonen and Pahkinen (1994) argue that accounting for the sampling complexities involved in data collection is essential for producing accurate estimates. They note that the use of clustering affects modeling procedures, as well as variance and parameter estimation. However, at present these issues are often ignored in research where model comparison statistics are used for fitting LC models to complex survey data.

Purpose of the Study

The accuracy of parameter and variance estimates in fitting LC models to a single group when using survey data has been studied by Patterson, Dayton and Graubard (2002). However, the accuracy of LC comparison statistics across multiple groups when using complex sampling data has not been explicitly addressed. To address this issue, the present study investigated the behavior of various model comparison statistics when performing simultaneous latent structure analyses on complex sample survey data by examining the effects of unequal sampling and clustering in the dataset. The focus of this study was to fit LC models to data from the 1979 and 1988 NHSDA and compare the latent structures across the two years. Building on the work of Patterson, Dayton and Graubard (2002), this study focused on determining which, if any, model comparison

statistics for simultaneous latent structure analysis are appropriate for use with complex sample surveys.

The problem of fitting LC models to data across groups was addressed in two parts:

1. An examination of the various model comparison statistics for simultaneous latent structure analysis to determine how the model comparison statistics behaved under various sampling conditions; and

2. A simulation of the 1979 and 1988 NHSDA to determine if reported drug use patterns have changed over time.

A Monte Carlo simulation was used to determine how well the various model comparison statistics function with respect to various complex samples. Each simulation involved drawing samples from two groups. A variety of simulations were performed to investigate how model comparison statistics function under various conditions, including: 1) low, medium and high variability in latent class structures among groups; 2) low versus high item-specific error rates; 3) equal weighting versus oversampling of one group (unequal sampling weights); 4) three cluster sizes resulting in different intraclass correlations; and 5) small and large sample sizes.

Overview

The remainder of this document is divided into five chapters: 2) Review of Literature; 3) Analysis of NHSDA Data; 4) Methods; 5) Simulation Results; and 6) Discussion and Conclusions. The Review of Literature includes findings from drug use research, a discussion of the sampling methods used in the NHSDA and a description of

latent class scaling and simultaneous latent structure analysis. Descriptions of the model

comparison statistics used in this study are also offered.

In Chapter 3, two years of the NHSDA were analyzed individually using latent

class software. The analysis was used to identify the latent class structures that exist in

each of the years of the NHSDA. These latent class structures were used to build the

simulated cases. The parameters used in the simulations are described in Chapter 4,

Methods. An explanation of and justification for their selection is also offered. Results

and conclusions are then presented in the final two chapters.

CHAPTER 2

REVIEW OF LITERATURE

Stages of Drug Use

Recent discussion concerning drug use has, in part, focused on the theory of "gateway" drugs, the use of which may influence or contribute to the use of more addictive, illicit drugs (Kandel & Faust, 1975; Yamaguchi & Kandel, 1984; Allebeck & Romelsjo, 1993). Several less addictive drugs, such as nicotine, alcohol and marijuana, are viewed as leading to more serious forms of illicit drug use, including cocaine, crack and heroin (Blaze-Temple & Lo, 1992; Johnson, Boles, & Kleber, 2000; Lai et al, 2000). As a result, the theory summarized here holds that the development of drug use follows a stage-like progression in which individuals make transitions from one level of drug use to another.

There is interest in defining a common developmental pathway from licit to illicit drug use since the reported use of so called "street drugs," such as cocaine, crack and heroin, has been a public policy concern since the 1960's. If, in fact, more serious drug use is induced by the use of licit drugs, such as alcohol or cigarettes and then by the use of marijuana, it may be possible to reduce subsequent hard drug abuse by reducing or postponing the age of initiation of alcohol, cigarettes and marijuana (Golub & Johnson, 2001). Therefore, a typical progression of reported drug use and the identification of "gateway" drugs have been a focus of much research.

The type of relation described among different types of reported drug use is indicative of a Guttman model. A Guttman-type stages of drug use model would include

a specific sequence of use where only certain patterns would be allowable in the model. Guttman scaling assumes that a set of dichotomously scored, observable items can be ordered according to difficulty or degree. Therefore, individual responses to items higher on the scale are dependent on the responses to preceding items, forming a linear hierarchic structure. Thus, any response pattern that does not follow a Guttman sequence can be assumed to result from response errors (Guttman, 1947).

In terms of measuring drug use or addiction, the same type of hierarchical relation is hypothesized with the use of different substances correlated such that a respondent's reported use of one type of drug is dependent upon the reported use of lower order substances. Again, there is a permissible set of responses because the reported use of a higher order drug is dependent on the reported use of a lower order substance. Importantly, a key requirement of the stage hypothesis of drug use that arranges substance use on a Guttman-like scale is that those reporting use of higher order drugs will report having used lower order drugs. However, users of lower order drugs are not necessarily required to move to higher stages of drug use. A Guttman stages of drug use model has been suggested by several researchers (Yamaguchi & Kandel, 1984; Blaze-Temple & Lo, 1992; Kandel, Yamaguchi, & Chen, 1992). The common, sequential model established in these numerous studies is the basis for the model that was used in this study.

The model of drug use is typically characterized by four stages: 1) nonuse; 2) alcohol and/or tobacco (licit use); 3) marijuana; and 4) hard or "street drugs," such as cocaine, crack and heroin. Consistent with prior research on gateway drugs, the first stage of reported drug use (after no use) is the use of alcohol or cigarettes. Progression to the

next stage, marijuana use, has been shown to be very unlikely without reporting prior experimentation with alcohol or cigarettes. This is the first step from licit drug use for adults to illicit drug use. After marijuana, the next step in the drug use sequence is believed to be cocaine and heroin use. It has been shown in studies, such as those listed above, that few individuals report trying more serious illicit drugs without reporting prior experience using marijuana. More recently, Yamaguchi and Kandel (1984) and Kandel, Yamaguchi and Chen (1992) have suggested the addition of a fifth stage, prescribed psychoactive drugs. However, defining this category has been difficult since it can include several types of prescription drugs and data on the use of these drugs are limited.

Guttman scaling involves two basic assumptions: 1) the deterministic assumption of error free responses by respondents and 2) the assumption that all persons in the population of interest are scalable on the same response pattern, meaning that all deviations from allowable responses are due to random error (McCutcheon, 1987). A common criticism of Guttman scaling is its deterministic form that implies error free measurement. Even if a scaling model correctly characterizes a population of respondents, response vectors that deviate from permissible response patterns are likely to occur. These deviations can be due to respondent errors such as mistakes in response selection or guessing. Several alternative models have been introduced that more realistically assume that each of the individual items has an associated measurement error. Therefore, items are not considered perfect measures of a true ability or attribute. Probabilistic models have been used in this context, including the Proctor model (Proctor, 1970), the intrusion-omission error model (Dayton & Macready, 1976), the item-specific

error model (Clogg & Sawyer, 1981) and the latent class-specific model (Clogg & Sawyer, 1981).

The National Household Survey on Drug Abuse

The NHSDA is designed to measure the prevalence of illicit drug, alcohol and tobacco use among members of the household population aged 12 and older, and includes information on both licit and illicit reported drug use dating from 1972. The National Commission on Marijuana and Drug Abuse sponsored the first surveys in response to the growing concern over drug abuse. Currently, the survey is sponsored by the Substance Abuse and Mental Health Services Administration (SAMHSA).

The 1979 and 1988 NHSDA used similar methodologies. The sample has always been a stratified, multistage, area probability sample of the target population with the target population being members of households located in the continental United States age 12 and older. Each year a national probability sample of households has been selected from approximately 100 primary sampling units (PSUs). The household population includes more than 98 percent of the United States population, excluding persons living in institutional group quarters, such as military installations, college dormitories, hotels, hospitals and jails and the homeless. PSUs are selected from various strata that vary from year to year. These strata are defined by groups the study is interested in focusing on for a particular year of data collection. Between 1971 and 1990 the NHSDA was conducted in households located in the continental United States. Starting in 1991, however, the sample was redesigned and more areas and populations were added to the sample frame. For instance, residents of Alaska and Hawaii were added

9

to the NHSDA sampling frame in 1991, as well as college students living in dormitories and military personnel living in the United States. For these reasons, data from years prior to 1990 were compared.

The sample design involved several selection stages. First, primary areas, specifically counties, groups of counties, or metropolitan areas, were selected, then sub-areas (area segments) within these primary areas and households within the sub-areas. Next, households were screened for basic demographics and between zero and two eligible residents were selected from each sampled household.

The basic methodology of the survey has remained the same over the three decades that data have been collected. One household member of each sample dwelling unit is asked screening questions on the household composition, including the age, race, gender and ethnicity of all household members. Based on this information, between zero and two household members are selected to participate in the interview. Face-to-face interviews are then conducted with the sampled participates, with more sensitive drug-use questions being self-administered. Response rates have generally been approximately 90 percent for the screening stage and 80 percent for the interview.

At the conclusion of data collection, sample weights are calculated that reflect the complex structure of the sampling and adjust for respondents' probability of selection. When conducting statistical analyses on complex survey samples, sampling weights, w_{Fi}, must be assigned to each respondent to compensate for unequal selection probabilities and nonresponse or noncoverage. Therefore, the analysis technique used requires sampling weights to estimate the characteristics of the target population from the reports of the sample. Sample weights are then adjusted for non-response and refusal to

participate. Finally, these sampling weights are post-stratified to census projections of population totals for age, sex, race/ethnicity and population groups. The resulting weights were used in the analyses reported in this document.

Although sections of the survey have undergone changes and additions over the years, the core questions about drug use have remained the same allowing for comparisons across years. Each respondent's history of drug use was determined from answers to the following questions: "How old were you the first time you…[used *alcohol*, *cigarettes*, *marijuana*, *cocaine* and *heroin*]?" The respondent's answers to these questions were then recoded into a reported drug use flag that records 1 if the respondent has used the drug of interest and 0 if the respondent has not used the drug of interest. The recoded answers calculated for the flag item were used to establish a pattern of reported drug use for each respondent.

Based on the models identified in the literature and the data available in the NHSDA, a number of allowable sequences of reported drug use that resemble a modified Guttman scale were identified. Drugs are ordered according to their understood order of use: 1) alcohol; 2) tobacco; 3) marijuana; 4) cocaine and/or heroin. The model follows a typical Guttman scale with one exception; tobacco can be used without prior use of alcohol {0,1,0,0}. This modified version of a Guttman scale, called a biform scale, allows for six latent classes: {0,0,0,0}, {0,1,0,0}, {1,0,0,0}, {1,1,0,0}, {1,1,1,0} and {1,1,1,1}. A biform scale is a combination of two different linear scales. In this case, two five-class, linear structures would be combined to allow six permissible response vectors.

Latent Class Scaling

Lazarsfeld and Henry (1968) provided a comprehensive examination of models that relate latent variables to scores or categories of manifest variables. Later named "latent class analysis," LCA is a method of empirically identifying categorized latent variables from discrete observed variables. LCA allows the identification of a set of exclusive and exhaustive latent classes that account for the distribution of respondents. Therefore, LCA can be used for testing whether a theoretical structure adequately fits a dataset.

Latent class scaling is a special use of LC models that assumes that the classes of the latent variable can be ordered by degree from lowest to highest. In the context of the NHSDA data, the observed measures can be rank ordered in terms of their acceptability and ease of use. Thus, members of higher latent classes have experience with higher order drugs, while members of lower latent classes have experience with lower order drugs or no experience at all.

For four observed variables, an unconditional probability for a response vector, $\mathbf{y}_s$, is:

$$P(\mathbf{y}_s) = \sum_{t=1}^{T} \pi_{it}^{\overline{A}X} \pi_{jt}^{\overline{B}X} \pi_{kt}^{\overline{C}X} \pi_{lt}^{\overline{D}X} \pi_t^X$$

The conditional probabilities, $\pi_{it}^{\overline{A}X}, \pi_{jt}^{\overline{B}X}, \pi_{kt}^{\overline{C}X}, \pi_{lt}^{\overline{D}X}$, represent the probabilities of a respondent in class t of latent variable X being at a particular level of the observed variables. Thus, if A represents reported alcohol use and i = 0 represents no use of alcohol, then $\pi_{01}^{\overline{A}X}$ is the probability of a respondent in class 1 of latent variable X reporting no use of alcohol. In addition, π_t^X, represents latent class probabilities for the

distribution of classes of the latent variable X. In the case of NHSDA data, the number of latent classes represents the number of allowable reported drug use classes (Goodman, 1974).

Restrictions on the probabilities are:

1) Within each of the T latent classes, the item conditional probabilities sum to

$$\text{one}\left(\text{i.e.} \sum_i \pi_{it}^{\overline{A}X} = \sum_j \pi_{jt}^{\overline{B}X} = \sum_k \pi_{kt}^{\overline{C}X} = \sum_l \pi_{lt}^{\overline{D}X} = 1.00\right); \text{ and}$$

2) The sum of the latent class probabilities over the T classes of latent variable X is one $\left(\text{i.e.,} \sum_t \pi_t^X = 1.00\right)$.

Given that $P(\mathbf{y}_s)$ is the unconditional probability for a response vector, then the log-likelihood for a sample of N respondents, assuming SRS, is:

$$\ln(L) = \sum_{s=1}^{N} \ln P(\mathbf{y}_s)$$

The maximum likelihood (ML) estimates for the LC proportions and conditional probabilities for a specific LC model can be determined by maximizing L with respect to the parameters. The ML estimator is the value that generates the greatest probability for the observed response pattern (Hambleton & Swaminathan, 1985). ML parameter estimates can be calculated using iterative algorithms, such as the estimation maximization (EM) algorithm (Dempster, Laird & Rubin, 1977), Fisher's method of scoring (Rao, 1965) or Newton-Raphson iterations (Haberman, 1979). The fit of LC models to data is typically assessed using chi-square tests. The degrees of freedom (DF) for such tests is, in general, the number of possible response patterns decreased by the number of identified parameter estimates plus one.

When complex survey sampling methods are used in data collection, sample weights, w_i, are assigned to each respondent. Thus, incorporating sample weights into the log-likelihood, a pseudo-likelihood can be defined as:

$$\ln(L^*) = \sum_{s=1}^{N} w_s \ln P(\mathbf{y}_s)$$

As shown by Patterson, Dayton and Graubard (2002), the sample-weighted, pseudo-likelihood method can be used to obtain LC proportions and conditional probabilities from the survey data. Pfeffermann (1993) also supports the use of the pseudo-likelihood method by finding that its application results in accurate parameter estimates.

Scaling models represent restricted LC models. The latent classes correspond to permissible scale types such as those representing a Guttman scale (e.g., 0000, 1000, 1100, 1110, 1111). Conditional probabilities are restricted so that deviations from the permissible response patterns can be interpreted as errors. Specifically, the Proctor model posits a single overall error rate. The intrusion-omission error model posits two distinct types of errors: intrusion errors and omission errors. The item-specific error model restricts conditional probabilities such that there are distinct error rates for each variable and the latent class-specific model has distinct error rates for each of the latent classes. Dayton (1999) shows the required restrictions for these models in detail.

Simultaneous Latent Structure Analysis

Clogg and Goodman (1984, 1985) present a method for comparing latent structures among groups called simultaneous latent structure analysis. Clogg and Goodman (1986) extended this method for multiple group comparisons for use with

scaling models. They refer to three types of models for comparing latent structures: 1) the unconstrained or heterogeneous model; 2) the partial-homogeneous model; and 3) the model of complete homogeneity. The unconstrained heterogeneous model has no restriction placed on the within group parameters—error rates and latent class probabilities are unrestricted. In other words, it is the usual LC model with the addition of a blocking variable (Dayton, 1999).

Clogg and Goodman (1984, 1985) also refer to models that can be partially-homogeneous. In this case, some of the error rates and latent class probabilities are constrained to be equal. According to McCutcheon (1987), constraints on the error rates are usually the first type of restrictions imposed when testing hypotheses about group differences. Constraints on the latent class probabilities are considered last. This order of imposing constraints allows us to first learn if the groups' classes are similar in structure by comparing probabilities of responses for corresponding classes across groups. Latent class probabilities can then be constrained to study whether the proportions in each class are similar. Models with various types of partial homogeneity constraints can be developed to test between-group similarities by imposing sets of constraints on different error rates and latent class probabilities (Dayton, 1999). For the purposes of this study, a partial-homogeneous model refers to a model with constraints on the error rates.

Clogg and Goodman (1984, 1985) also present a completely homogeneous model where the latent classes of all groups are the same. For this type of model equality restrictions are placed on all error rates and conditional latent class probabilities.

Model Comparison Statistics

A variety of model comparison statistics is available for selecting a best fitting model for a given dataset. Five model comparison indices that are frequently used in applications were considered in this study: 1) chi-square difference tests; 2) Akaike Information Criterion (AIC); 3) Bayesian Information Criterion (BIC); 4) Rissanen (RIC) (1978) statistic; and 5) Wald test.

Random sampling leads to the assumption of an approximate chi-square distribution for the familiar Pearson statistic, X^2, the likelihood-ratio statistic, G^2 and the Read-Cressie statistic, I^2 (Read & Cressie, 1988) setting their λ parameter to 2/3. When using complex survey data, it is not possible to make this assumption (Lohr, 1999). This issue is compounded when clustering is used in data collection. Clustering can often lead to a loss in accuracy when individuals within clusters are more similar to one another than to individuals in other clusters. The intraclass correlation coefficient (ICC) is an indicator of how similar individuals are within the same cluster. That is, it is a measure of the level of homogeneity among members of the same cluster. The ICC, or ρ, is a coefficient that takes on values between -1 and 1, where 1 occurs when individuals in the same cluster have identical values. The ICC is a comparison of the within cluster and between cluster variation and is defined as:

$$\rho = 1 - \frac{S_w^2}{S^2} \frac{N}{N-1} \approx \frac{S_b^2}{S^2}$$

where S_w^2 is the variance within clusters, S_b^2 is the variance between clusters and S^2 is the total sample variance. When $S_w^2 = 0$, all units within each cluster have the same value and

$\rho = 1$. When $S_b^2 = 0$, all cluster means are equal (i.e. all variance is within clusters) and $\rho = 0$.

Generally, when cluster sampling is used, there is a positive intraclass correlation that results in underestimation of sampling variances and consequent underestimation of p-values for significance tests (Kalton, 1983). In reality, a p-value can be much larger than the one reported when the sampling method is not considered. For this reason, clustering can not be ignored when using chi-square tests with complex survey data.

The Akaike (1973, 1987) Information Criterion, AIC, favors models that are expected to show the smallest decrease in likelihood when cross-validated on another sample selected from the same population. Therefore, the model with the lowest AIC is preferred. AIC for the hth of H different models is defined as:

$$AIC_h = -2\ln (L_h) + 2m_h$$

where m_h is the number of independent parameters estimated when fitting the hth model to the data. When unequal sampling weights are applied, a pseudo-likelihood is used and L_h is replaced by L_h*.

The Bayesian Information Criterion, BIC, is an alternative comparison index to AIC. BIC is similar to AIC, except that BIC corrects for the criticism that AIC is not asymptotically consistent (that is, it does not account for the sample size, n, when making comparisons). BIC is given by:

$$BIC = -2\ln (L_h) + \ln(n)m_h$$

Due to the fact that $\ln(n)$ is greater than two if n is greater than seven, BIC tends to favor less complex models than AIC. When unequal sampling weights are applied, L_h is replaced by L_h*.

Sclove (1987) describes another variation on AIC presented by Rissanen (1978). The Rissanen RIC statistic is given by:

$$RIC = -2\ln (L_h) + \ln\left(\frac{n+2}{24}\right) m_h$$

The likelihood is replaced with a pseudo-likelihood (L_h*) when unequal sampling weights are applied.

AIC, BIC and RIC vary in terms of the penalty per parameter. For instance, using the sample size from the 1979 NHSDA, $n = 7224$, the penalty for each parameter for the AIC formula is 2 and for BIC is 8.88, while the penalty per parameter used in RIC falls between these two at 5.7. The latent variable program, LEM (Vermunt, 1997), reports both AIC and BIC. Although AIC and BIC have been used in assessing fit in LC models, their effectiveness when using complex survey data is not known (Patterson, Dayton & Graubard, 2002).

The Wald test is an alternative to chi-square statistics for testing model fit. If $\theta = [\theta_1, \theta_2 \ldots \theta_p]^T$ is a p x 1 vector of parameters, the Wald statistic is given by:

$$X_W^2 = \hat{\theta}' V^{-1} \hat{\theta}$$

where V is the estimated covariance matrix of the maximum likelihood estimate, $\hat{\theta}$.

The Wald statistic approximately follows a X^2 distribution for large samples (Lohr, 1999). The Wald test has been adapted to data from complex surveys by using a jackknife estimate of the variance/covariance matrix (Patterson, Dayton & Graubard, 2002). In the context of LC models, θ is a vector of unique LC proportions and conditional probabilities. For example, with a single sample and a unconstrained model with two latent classes and four manifest, dichotomous variables,

$$\theta = \left[\pi_1^X, \pi_{11}^{\overline{AX}}, \pi_{11}^{\overline{BX}}, \pi_{11}^{\overline{CX}}, \pi_{11}^{\overline{DX}}, \pi_{12}^{\overline{AX}}, \pi_{12}^{\overline{BX}}, \pi_{12}^{\overline{CX}}, \pi_{12}^{\overline{DX}} \right]^T.$$

The jackknife method is a resampling method where one observation is removed and the statistic of interest is then recalculated. This process is repeated until the statistic of interest has been calculated n number of times. Note that an observation may be a single respondent, a cluster or some other unit. Lee, Forthofer and Lorimor (1986) and Wolter (1985) demonstrate this method for sample weighted data.

Jackknife estimates can also be calculated by deleting individual PSUs or clusters without organizing them into groups. It is important to note that because the standard error of an estimated parameter can be accurately estimated from the variation among cluster totals, it is possible to calculate jackknife estimates while ignoring the subsampling procedures used within clusters. As long as the first-stage sampling fraction is low, the contribution from later stage variances is incorporated in the sampling error estimated using the clusters (Lee, Forthofer & Lorimor, 1986).

CHAPTER 3

PRELIMINARY ANALYSIS OF

NATIONAL HOUSEHOLD SURVEY ON DRUG ABUSE DATA

The design of the simulations conducted in this study was based on preliminary analyses of the data from the 1979 and 1988 NHSDA. Although test statistics, such as chi-square and AIC, have not been evaluated for use with complex survey data, they were used as indicators of fit for the preliminary analyses. However, the main purpose was to identify LC proportions and error rates that represented realistic values for use in simulations.

Both five- and six-class models were fit to the sample data from the 1979 and 1988 NHSDA. The five-class model represents a linear hierarchy following a Guttman scale as supported by the literature. However, the literature on reported drug use also suggests a six-class model representing a biform scale. That is, prior drug-use research suggests a model that follows a Guttman scale with the possible exceptions of the use of alcohol and tobacco (Collins et al., 1994). This exception includes the use of tobacco without prior use of alcohol.

Table 1 shows results of fitting various error models to the 1979 NHSDA data using a five- and six-class structure using the latent class program LEM (Vermunt, 1997). The chi-square statistics for the models are relatively large and, as is often the case with large sample sizes, none of the models yields a non-significant fit statistic. However, the six-class, item-specific error model has the lowest chi-square values ($X^2 = 19.4$, $G^2 =$

20.1 and $I^2 = 18.9$ with 6 degrees of freedom), with an index of dissimilarity that is

relatively low (I_d = .005).[1]

Each of the other error models, comprising Proctor, intrusion-omission and latent

class-specific error models, for both five- and six-class models produced relatively higher

chi-square values and dissimilarity indices. The five-class, item-specific error model also

produced poor fit statistics and higher AIC, BIC and RIC values than the six-class, item-

specific error model. For these reasons, the six-class, item-specific error model was

selected as the best fit to the data. Although the six-class, item-specific error model does

not have a good absolute fit to the data, it does appear to be a reasonable approximating

model. In the context of the drug use data, an item error would occur, for example, if a

respondent who had never smoked accidentally responded positively to tobacco use or if

a tobacco user denied use.

Table 1. Results from fitting error models to 1979 NHSDA data

Latent Classes	Error Model	X^2	G^2	I^2	I_d	AIC	BIC	RIC	DF
Five	Proctor	366.2	437.1	374.1	.029	22486	22520	22505	10
	Intrusion-Omission	149.8	203.5	160.9	.022	22254	22296	22277	9
	Item-Specific	92.9	116.2	97.2	.019	22171*	22226*	22201*	7
	LC–Specific	339.5	409.3	348.1	.024	22466	22528	22500	6
Six	Proctor	312.3	278.3	287.6	.021	22329	22370	22352	9
	Intrusion-Omission	155.0	172.3	155.2	.014	22225	22273	22251	8
	Item-Specific	19.4	20.1	18.9	.005	22077*	22139*	22111*	6
	LC–Specific	112.5	125.9	113.7	.009	22187	22262	22228	4

*Minimum value

---

[1] According to Dayton (1999), it is desirable to have an index of dissimilarity less than .05.

Because the item-specific error model appeared to demonstrate the best fit of the four models examined, the fit of several item-specific error models with more than six classes were tested to determine whether a better fitting model could be found. Table 2 shows the number of latent classes in each model tested and also lists the additional classes added. These classes were selected because they fit potential patterns of reported drug use that may not have been included in the six-class model.

Table 2. Results of fitting item-specific error models to 1979 NHSDA data

Latent Classes	Added Classes	X^2	G^2	I^2	I_d	AIC	BIC	RIC	DF
5	None	92.9	116.2	97.2	.019	22171	22226	22201	7
6	{0100}	19.4	20.1	18.9	.005	22077	22139*	22111	6
7	{0100} {1010}	13.1	13.7	12.6	.002	22072*	22141	22110*	5
8	{0100} {1010} {1011}	13.1	13.7	12.6	.002	22074	22150	22116	4

*Minimum value

As shown in Table 2, adding additional classes {1010} and {1011} to the six-class model improves model fit. The chi-square statistics shrink somewhat and the index of dissimilarity is lowered. Although these fit statistics suggest that a seven- or eight-class model may fit the data better than the six-class model, the latent classes that are added to the original model constitute a very small proportion of the population. The additional latent class in the seven-class model only represents .0064 of the population and the additional latent class in the eight-class model was nearly zero. Also, the addition of a class or two to the original six-class model only affected fit in one cell in the design, suggesting that the improvement in fit is negligible in practical terms.

Because the literature on reported drug use favors a biform scale, the LC proportions are reasonable and practical, with adequate model fit, it was concluded that

the six-class, item-specific error model is the most reasonable approximating model for

this study (Table 3). In addition, BIC is lowest for this model, which is an important

indicator given the large size of the sample. The index of dissimilarity is also relatively

low for the six-class model (I_d = .005). Because of the reasonable approximating fit of the

model to the 1979 NHSDA data, the six-class, item-specific error model appears to be the

most appropriate model for this research.

Table 3. Estimated latent class proportions and item-specific error rates for six-
class, item-specific error model fit to 1979 NHSDA data

Latent Classes and Item-Specific Error Rates	Proportion
$\hat{\pi}_1^X$	.08
$\hat{\pi}_2^X$	.03
$\hat{\pi}_3^X$	.09
$\hat{\pi}_4^X$	.51
$\hat{\pi}_5^X$	.21
$\hat{\pi}_6^X$	.09
$\hat{\pi}_{e1}$	.004
$\hat{\pi}_{e2}$	.060
$\hat{\pi}_{e3}$	.005
$\hat{\pi}_{e4}$	.009

In addition to the 1979 NHSDA data, a six-class model also was fit to the sample

data from the 1988 NHSDA. Table 4 shows the results of fitting the various error models

to these data. As with the 1979 NHSDA data, the absolute fit of the models to the data is

not satisfactory. However, model selection was based on the best reasonable fit.

The chi-square statistics for each of the models are relatively large. Similar to the

analysis of the 1979 NHSDA, the item-specific error model does not have a good

absolute fit to the data, but it does appear to be a reasonable approximating model. The

chi-square statistics are high, indicating a poor fit. However, the index of dissimilarity is

quite good. Also, when examining the AIC, BIC and RIC measures, the item-specific

error model produced the lowest statistics.

Table 4. Results from fitting six-class error models to 1988 NHSDA data

Error Model	X^2	G^2	I^2	I_d	AIC	BIC	RIC	DF
Proctor	333.1	308.0	314.2	.023	29834	29877	29858	10
Intrusion-Omission	180.9	193.2	181.1	.016	29721	29771	29749	8
Item-Specific	20.7	18.6	19.5	.004	29551*	29615*	29586*	6
Latent Class – Specific	133.1	137.7	132.5	.010	29674	29752	29717	4

*Minimum value

Although the item-specific error model was the best fitting of the four models

examined, the fit of several item-specific error models with more and less than six classes

were tested to determine whether a better fitting model could be found (Table 5).

Table 5. Results of fitting several item-specific error models to 1988 NHSDA data

Latent Classes	Added Classes	X^2	G^2	I^2	I_d	AIC	BIC	RIC	DF
5	None	191.6	213.0	195.3	.032	29743	29800	29775	7
6	{0100}	20.7	18.6	19.5	.004	29551	29615*	29586*	6
7	{0100} {1010}	18.0	15.8	16.8	.002	29550*	29621	29590	5
8	{0100} {1010} {1011}	18.0	15.7	16.7	.002	29552	29630	29595	4

*Minimum value

As with the 1979 NHSDA data, adding classes improves the model fit indices but

the additional classes represent only a small portion of the population. The fourth class of

the seven-class model only represents .0043 of the population and in the eight-class

model, once again, the eighth class went nearly to zero. For these reasons, and due to the

BIC and RIC values that favor the six-class model, the less complex, six-class model was

selected as the best approximating model. Table 6 shows the LC proportions and item-

specific error rates resulting from the analysis of the 1988 NHSDA.

Table 6. Estimated latent class proportions and item-specific error rates for six-class,
item-specific error model fit to 1988 NHSDA data

Latent Classes and Item-Specific Error Rates	Proportion
$\hat{\pi}_1^X$	.11
$\hat{\pi}_2^X$	.03
$\hat{\pi}_3^X$	.10
$\hat{\pi}_4^X$	.42
$\hat{\pi}_5^X$	.23
$\hat{\pi}_6^X$	.11
$\hat{\pi}_{e1}$	.007
$\hat{\pi}_{e2}$	.062
$\hat{\pi}_{e3}$	.004
$\hat{\pi}_{e4}$	.003

Six-class homogeneous, partial-homogeneous and heterogeneous models were fit

to the data from the 1979 and 1988 NHSDA and model comparison statistics were

calculated (Table 7).

Table 7. Results of fitting homogeneous, partial-homogeneous and heterogeneous models
to 1979 and 1988 NHSDA data

Comparison Model	X^2	G^2	I^2	I_d	AIC	BIC	RIC	DF
Homogeneous	184.1	182.3	181.5	.043	73831	73908	73876	21
Partial	46.1	43.2	43.3	.005	73702*	73817*	73770*	16
Heterogeneous	40.2	38.7	38.4	.005	73706	73852	73792	12

*Minimum values

Based on the model comparison statistics, the partial-homogeneous model is the best approximating model for the NHSDA data. AIC, BIC and RIC favor the partial-homogeneous model and the chi-square and index of dissimilarity measures are no worse than the other two models. In addition, the LC proportions and item-specific error rates in the partial-homogeneous model are similar to the LC proportions and item-specific error rates identified in the separate analyses of each year. Table 8 presents the latent classes and item-specific error rates conditional on group membership from the partial-homogeneous model.

Table 8. Estimated latent class proportions and item-specific error rates by group membership for partial-homogeneous model fit to 1979 and 1988 NHSDA data

Latent Classes and Item-Specific Error Rates	1979	1988
$\hat{\pi}_1^X$	.08	.11
$\hat{\pi}_2^X$	.02	.04
$\hat{\pi}_3^X$	.09	.10
$\hat{\pi}_4^X$	.51	.42
$\hat{\pi}_5^X$	.21	.22
$\hat{\pi}_6^X$	.09	.11
$\hat{\pi}_{e1}$	.006	.006
$\hat{\pi}_{e2}$	.061	.061
$\hat{\pi}_{e3}$	.005	.005
$\hat{\pi}_{e4}$	.002	.002

As shown in Table 8, the latent class parameter estimates for the two years of data resulted in very similar LC proportions with the exception of the fourth latent class. Each of the latent classes increased slightly, apart from the fourth latent class, which decreased. The fourth class is characterized by individuals who have used alcohol and tobacco, but

have refrained from marijuana and harder "street drugs." The LC proportions and item-specific error rates of the six-class, item-specific error model for the 1979 and 1988 NHSDA (Tables 3 and 6) were similar to those resulting from fitting the partial-homogeneous model, where item-specific error rates are constrained to be equal across groups (Table 8).

CHAPTER 4

METHODS

The purpose of this study was to assess how well model comparison statistics function when using complex survey data with latent class comparison methods. To examine this issue, simulations were conducted that use data with weighting and clustering. A partial-homogeneous model was assumed to be the true model as identified in the preliminary analysis of the NHSDA for 1979 and 1988. Then, the nature of the relation among groups (i.e. heterogeneous, partial-homogeneous, or homogeneous) was analytically derived from the data and compared to the "true" relation among groups.

The fit of homogeneous, partial-homogenous and heterogeneous models was assessed using seven model comparison statistics: 1) X^2; 2) G^2; 3) I^2; 4) AIC; 5) BIC; 6) RIC; 7) Wald test. The proportion of times each model comparison statistic detected the true relation among groups was used as a measure of the overall accuracy of the model comparison statistic with complex sample survey data.

Simulation Sample Design

The basic sample design consisted of two groups from which certain numbers and sizes of clusters were sampled. Each group corresponded to a different pattern of LC proportions that were compared across groups using the model comparison statistics above. Item-specific error rates were held constant across groups, clusters and classes. Sample weights were applied either equally across groups using a 1:1 ratio or unequally

using a 1:5 ratio. A 1:5 ratio was used as in Patterson, Dayton, and Graubard (2002) to ensure a clear difference between the two groups.

We assumed that samples were drawn from a very large finite population of size N with K clusters representing the primary sampling unit (PSU) so it was reasonable to ignore the finite sampling correction. A design similar to that used in the 1988 NHSDA, consisting of two groups and several clusters, was used. The number of clusters sampled was between 20 and 600 depending on the sample size. This design is consistent with the NHSDA data that tends to sample large numbers of clusters across two years. A number of factors were fixed throughout all simulations and other factors were manipulated so the differences in behavior between the model comparison statistics under differing circumstances could be studied. Table 9 shows the parameters of the simulated cases.

Table 9. Design specifications for simulated cases

Factor	Specifications
Sample Sizes	500
	3000
Item Error Rates	.05
	.25
LC Patterns	$\boldsymbol{\pi}_{1t}^{G\overline{X}} = \{.08, .03, .08, .50, .23, .08\}$
	LCP1:
	$\boldsymbol{\pi}_{2t}^{G\overline{X}} = \{.10, .05, .10, .40, .25, .10\}$
	$\boldsymbol{\pi}_{1t}^{G\overline{X}} = \{.14, .14, .14, .30, .14, .14\}$
	LCP2:
	$\boldsymbol{\pi}_{2t}^{G\overline{X}} = \{.17, .17, .17, .17, .17, .17\}$
	$\boldsymbol{\pi}_{1t}^{G\overline{X}} = \{.11, .11, .11, .22, .22, .22\}$
	LCP3:
	$\boldsymbol{\pi}_{2t}^{G\overline{X}} = \{.22, .22, .22, .11, .11, .11\}$
Sample Weights	1:1
	1:5
Cluster Sizes	5 (ICC=.05)
	10 (ICC=.10)
	25 (ICC=.20)

<u>Fixed Factors</u>

The following factors were fixed throughout the simulations: the number of items and the number of latent classes.

1) The number of items was set at four. This is the same number of items often found in drug use research and used in the preliminary analysis of the NHSDA data.

2) Based on the results of the preliminary analysis, a biform scale was fit to the data using six latent classes.

<u>Manipulated Factors</u>

To examine the ways in which the competing model comparison statistics function, various factors were manipulated to produce simulations with varying characteristics. The manipulated factors were: sample size, item-specific error rates, LC proportions, sample weights and cluster size.

1) The sample sizes, 500 and 3000, were chosen to correspond to relatively small and relatively large drug use studies.

2) The item-specific error rates were chosen to be equal across groups to correspond to a partial-homogeneous model. The item-specific error rates were set equal across items and latent classes based on the results of fitting the partial-homogeneous model to the NHSDA data. In the preliminary analysis, the item-specific error rates from the partial model were very low for each item. Therefore the item-specific error rates were set low at $\pi_{e(1)} = \{.05,.05,.05,.05\}$ for one set of simulations. Since this group of item-specific error rates represents an extreme case, item-specific error rates were set

to .25 for a second set of simulations to explore how the model comparison statistics function when the item-specific error rates are higher. (Table 9).

3) LC proportions were selected that were similar to those identified in the analysis of the NHSDA data. As discussed previously, the LC proportions for the two years of data were similar with the exception of the fourth latent class. Each of the latent classes increased slightly between years, apart from the fourth latent class that decreased. Therefore, two sets of latent class values that were similar to those found in the preliminary analysis were replicated in the simulation. These latent class values were: $\pi_{1t}^{G\overline{X}} = \{.08,.03,.08,.50,.23,.08\}$ and $\pi_{2t}^{G\overline{X}} = \{.10,.05,.10,.40,.25,.10\}$.

Since the differences among these sets of LC proportions are relatively small, meaning it may be difficult for the model comparison statistics to detect, more extreme cases were explored. A second pattern of latent classes was compared in which the difference among groups was larger in the fourth latent class: $\pi_{1t}^{G\overline{X}} =$ $\{.14,.14,.14,.30,.14,.14\}$ and $\pi_{2t}^{G\overline{X}} = \{.17,.17,.17,.17,.17,.17\}$. Thus, the only major difference can be detected in the fourth group. However, this difference involves reducing the fourth class by almost half. In a third pattern, the latent classes were varied in a more extreme manner, making the differences in the latent structures more obvious: $\pi_{1t}^{G\overline{X}} = \{.11,.11,.11,.22,.22,.22\}$ and $\pi_{2t}^{G\overline{X}} = \{.22,.22,.22,.11,.11,.11\}$. Thus, the first LC pattern has low variability, the second LC pattern has medium variability and the third LC pattern has high variability.

To obtain random variation across clusters, LC proportions were generated using a Dirichlet distribution, the multivariate generalization of the beta distribution. The Dirichlet distribution is the natural conjugate prior for the multinomial distribution

and is often used when Bayesian LCA is performed (Vermont & Magidson, 2000).

For constants $c_i > 0$, $i = 1,\ldots,k$ and c_0 with $c = \sum\limits_{i=1}^{k} c_i$ the Dirichlet distribution is

defined as:

$$\frac{n!\,\Gamma\left(\sum\limits_{i=1}^{k} c_i\right)}{\Gamma\left(n + \sum\limits_{i=1}^{k} c_i\right)} \prod\limits_{i=1}^{k} \frac{X_i + c_i}{c_i}, \quad \sum\limits_{i=1}^{k} X_i = n, \quad X_i \geq 0$$

with a mean nc_i/c and variances and covariances that correspond to those of a

multinomial distribution with $p_i = c_i/c$. When $k = 1$, the Dirichlet distribution is

equivalent to a beta distribution (Evans, Hastings & Peacock, 1993).

Values were generated from a Dirichlet density by transforming random variates

generated from gamma distributions. Let $Y_1,\ldots,Y_{k+1}$ be independent gamma random

variables with parameters $(1, c_1)$ with $c_0 = 1$. If we define $Y = \Sigma Y_i$ then $X_i = Y_1/Y$ for

$i = 1,2,\ldots,k$. The result is $(X_1,\ldots,X_k)$ with a Dirichlet distribution with parameters

$(c_{11},\ldots, c_{k+1})$. For a six-class simulation, the result is $(X_1,\ldots,X_5)$ with parameters, say,

$\alpha_1,\ldots,\alpha_6$. The ratio of Dirichlet values is the same as the ratio of these α-values so a

set of latent class sizes could be generated to conform in expectation to a specified

pattern of LC proportions.

4) To demonstrate the effects of equal sample weights, all sample weights were set to

1.0 for some of the simulated cases. The effect of unequal sampling weights was

explored by using sampling fractions representing the selection of respondents

following a 1:5 ratio, in favor of group 1. Thus, respondents in group 1 have a higher

probability of selection than respondents in group 2. Following Patterson, Dayton and

Graubard (2002), a large ratio was selected so that the difference was sufficient to

demonstrate the effect of unequal sampling weights. When the sample weights used

in the 1988 NHSDA were plotted, they followed a lognormal distribution with a

median of .510 and a variance of 1.56. Therefore, sample weights were generated for

the observations in the simulations using a lognormal distribution with comparable

mean and variance.

5) The intraclass correlation coefficient (ICC) is an indicator of how similar individuals

are within clusters. Lohr (1999) and Kalton (1983) note that positive intraclass

correlation between observations is a likely result when performing complex survey

sampling. For this reason, the cluster size and number were varied to introduce

different levels of the ICC into the samples. The cluster size varied between 5, 10 and

25 that in turn determined the number of possible clusters depending on the sample

size. This yielded ICCs of .05, .10 and .20.

Clusters of responses were developed around the LC proportions derived from the

Dirichlet distributions. Let θ_{lh} be the LC proportion in the first latent class in group h

and k_h clusters of equal size b exist within group h. To induce intraclass correlation,

sets of values of θ_{lh} were generated from Dirichlet values so that $\theta_{1hs},\ldots,\theta_{6hs}$ where s

refers to the set of values and $s = 1,\ldots, k_h$. These values $\theta_{1hs},\ldots,\theta_{6hs}$ were then used to

generate b respondents and assign them to latent classes. This process was repeated k_h

times until the desired sample size was achieved. By following this procedure, k_h

clusters containing b individuals were formed around k_h or s sets of LC proportions.

Three cluster sizes were selected because they were reasonable values (Kalton, 1983)

and their ICCs varied in size (Table 10).

Table 10. ICCs of simulated cases

Cluster Size	Number of Clusters	Group Sample Size	ICC
5	100	500	.20
5	600	3000	.20
10	50	500	.10
10	300	3000	.10
25	20	500	.04
25	120	3000	.05

Final Analysis of NHSDA

Because NHSDA data are collected using complex survey sampling methods, it is not possible to rely on traditional model comparison statistics to understand how patterns of drug use have changed over time in the U.S. population. In the preliminary analysis, each of the latent classes in the six-class model increased slightly, apart from the fourth latent class, which decreased more substantially. Therefore, using traditional model comparison statistics, it is possible to conclude that the proportion of individuals in the fourth class, characterized by individuals who have reported alcohol and tobacco use, but have refrained from marijuana and harder drugs, have decreased, while all other proportions of individuals in various stages of drug use have increased slightly.

Nevertheless, the accuracy of the model comparison statistics is unknown. Therefore, a portion of this simulation exercise was dedicated to reproducing two samples that have similar parameters to those observed in the 1979 and 1988 NHSDA data. The simulations were based on the assumption of a correct model. In this case, however, two samples representing respondents from the 1979 and 1988 NHSDA were simulated following the latent class and item-specific error rates defined similar to those from the preliminary analysis of the NHSDA data: $\pi_{1t}^{G\overline{X}} = \{.08,.03,.08,.50,.23,.08\}$; $\pi_{2t}^{G\overline{X}} =$

$\{.10,.05,.10,.40,.25,.10\}$; $\pi_{e(1)} = \{.05,.05,.05,.05\}$; and $\pi_{e(2)} = \{.25,.25,.25,.25\}$,

corresponding to groups. Model comparison statistics were calculated using

nontraditional methods for both real and simulated data. The simulated cases were used

to determine the accuracy of our conclusions based on the model comparison statistics

calculated from the real data.

The difference in fit of models for two groups was assessed, specifically

comparing the latent structures between the NHSDA for 1979 and 1988. A partial-

homogeneous model was assumed as identified in the preliminary analysis. A simple

random sample design with replacement, consisting of 100 clusters of 80 observations,

was used based on the parameters of the 1988 NHSDA. Clusters were selected from two

groups following a 1:1 ratio because the 1979 and 1988 NHSDA samples were similar in

size.

Five model comparison indices were examined that are often used in

simultaneous latent structure analysis to determine the difference in fit between

homogeneous and heterogeneous models: 1) chi-square; 2) AIC; 3) BIC; 4) RIC; 5) Wald

statistic. The variance of the Wald statistic was jackknifed. However, the matrix of

differences was calculated by taking the difference between expected values for the

homogeneous and partial-homogeneous models and from the partial-homogeneous and

heterogeneous models (Korn & Graubard, 1999).

Simulation Details

There is no general purpose LC software for analyzing complex survey data.

LEM (Vermunt, 1997) can be used to obtain weighted estimates of LC parameters,

although, it does not provide estimates of the standard errors that take into account clustering. Therefore, SAS IML code (SAS Institute, 1995) was written to implement an EM algorithm and jackknife variances for the Wald statistic. SAS IML was also used for simulating data for analysis. The programming criteria used in all simulations was:

- Each simulation consisted of 1000 replications.

- The convergence criterion was 10^{-5} in most cases. When item-specific error rates were .25, the convergence criterion was set to 10^{-4} to maintain reasonable computational time.

- An EM algorithm was programmed in SAS IML to calculate actual estimates of LC proportions and item-specific error rates.

- The maximum number of iterations allowed to achieve convergence in the LCA algorithm was set to 500. If the model did not converge within the set number of iterations, the results were ignored and an additional run was completed. The results from the additional run were substituted for the original.

CHAPTER 5

SIMULATION RESULTS

The proportion of correct decisions for each model comparison statistic for each set of specifications is reported as a measure of its accuracy in Table 11. Findings for X^2, G^2 and I^2 are discussed first, assessing their performance depending on sample size, item-specific error rates, cluster size and LC proportions. The performance of these model comparison statistics when using unequal sampling weights was also evaluated. Next, the results from AIC, BIC and RIC are reviewed. Again, their performances under various conditions were assessed, including sample size, item-specific error rates, cluster size, LC proportions and the use of unequal sampling weights. Finally, results for the Wald statistic were discussed while varying some of the same conditions. Because the calculation of the Wald statistic was computationally intensive, all of the same conditions were not evaluated. Finally, how each of the seven model comparison statistics performed based on the simulation of the 1979 and 1988 NHSDA data was discussed.

Determining Best Fitting Model

The accuracy of the model comparison statistics for selecting the correct model (i.e., partial-homogeneous model) for each of the simulated cases was tested by calculating chi-square statistics for each possible model—homogeneous, partial-homogeneous and heterogeneous. Since the homogeneous model is nested within the partial-homogeneous model and the partial-homogeneous model is nested within the heterogeneous model, the best fitting model can be determined by conducting chi-square

38

difference tests. If the null is rejected for the test comparing the homogeneous and partial-homogeneous models and accepted for the test comparing the partial-homogeneous and heterogeneous models, then the chi-square difference test will have selected the correct model. A similar method was used for the Wald statistic. The matrix of variances and covariances was replaced by matrices of differences in estimated values from the homogeneous and partial-homogeneous models and from the partial-homogeneous and heterogeneous models. Again, if the null was rejected for the difference test between the homogeneous and partial-homogeneous models and accepted for the difference test between the partial-homogeneous and heterogeneous models, then the Wald test selected the correct model. AIC, BIC and RIC indicate the best fitting model by their minimum value. For each statistic, the model that produced the lowest value was preferred. For the seven model comparison statistics, the proportion of correct decisions out of 1000 replications was recorded for each of the simulations. Sometimes the computing algorithm failed to converge to the ML estimator within 500 iterations. This occurred with a maximum rate of 1.2 percent and an average rate of .154 percent.

Accuracy for Equally Weighted Groups for X^2, G^2 and I^2

First, the results from equally weighted groups using X^2, G^2 and I^2 were examined. As expected, these model comparison statistics tended to perform well under similar circumstances. I^2 was not necessarily a more accurate model comparison statistic than the other two when sample sizes were small, although this is often noted as an advantage of I^2 (Read & Cressie, 1988). Tables 11 and 12 show results for X^2, G^2 and I^2.

Because results were very similar among the three chi-square statistics, bar graphs are presented only for X^2.

The accuracy of the three model comparison statistics was assessed by varying the sample size of comparison groups between 500 and 3000 respondents. Table 11 shows the proportion of times out of 1000 when X^2, G^2 and I^2 selected the correct model at the .05 significance level. Appendix A shows the proportion correct at both .05 and .01 significance levels. Highlighted cells indicate a proportion greater than .5. Light shading indicates accuracy levels between .501 and .750, medium shading indicates accuracy levels between .751 and .900 and the darkest shading indicates over 90 percent accuracy. When an incorrect model was selected by X^2, in 82 percent of the cases the homogeneous model was selected. Seventy-eight percent of the inaccurate results for G^2 and 81 percent of the inaccurate results for I^2 resulted in the selection of the homogeneous model.

Table 11 shows that all of the cases where sample sizes were large resulted in better than expected levels of accuracy (>50%). When sample sizes were large and item-specific error rates were low, the rate of correct model detection was greater than 90 percent. Similar levels of accuracy were observed in only two-thirds of the cases where the sample sizes were small. Simulations where the item-specific error rates were low and sample sizes were small resulted in less accurate results.

Table 11. Accuracy for equally weighted groups by sample size

with .05 significance level

Sample Size	Cluster Size	LC Pattern	Item-Specific Error Rate	X^2	G^2	I^2
500	5	1	.05	.532	.530	.541
			.25	.131	.162	.146
		2	.05	.864	.849	.862
			.25	.231	.261	.246
		3	.05	.953	.945	.952
			.25	.965	.913	.949
	10	1	.05	.529	.530	.533
			.25	.142	.164	.145
		2	.05	.877	.872	.881
			.25	.243	.270	.257
		3	.05	.953	.944	.958
			.25	.965	.922	.943
	25	1	.05	.546	.545	.553
			.25	.134	.170	.149
		2	.05	.869	.863	.871
			.25	.249	.299	.278
		3	.05	.943	.928	.945
			.25	.962	.924	.948
3000	5	1	.05	.944	.938	.943
			.25	.636	.840	.791
		2	.05	.968	.968	.970
			.25	.930	.939	.942
		3	.05	.947	.948	.945
			.25	.962	.841	.898
	10	1	.05	.958	.957	.958
			.25	.604	.858	.811
		2	.05	.948	.947	.950
			.25	.917	.921	.927
		3	.05	.953	.954	.953
			.25	.964	.825	.907
	25	1	.05	.942	.945	.945
			.25	.627	.867	.812
		2	.05	.951	.955	.955
			.25	.899	.890	.894
		3	.05	.961	.958	.960
			.25	.970	.824	.893

KEY

☐	< .500
☐	.501 to .750
☐	.751 to .900
☐	> .901

Figures 1 and 2 show the results by low and high item-specific error rates. As

expected, the model comparison statistics were more accurate when simulated cases

included low item-specific error rates, especially for small sample sizes. When sample sizes were large, the model comparison statistics were more accurate at identifying the correct model regardless of the item-specific error rates used.
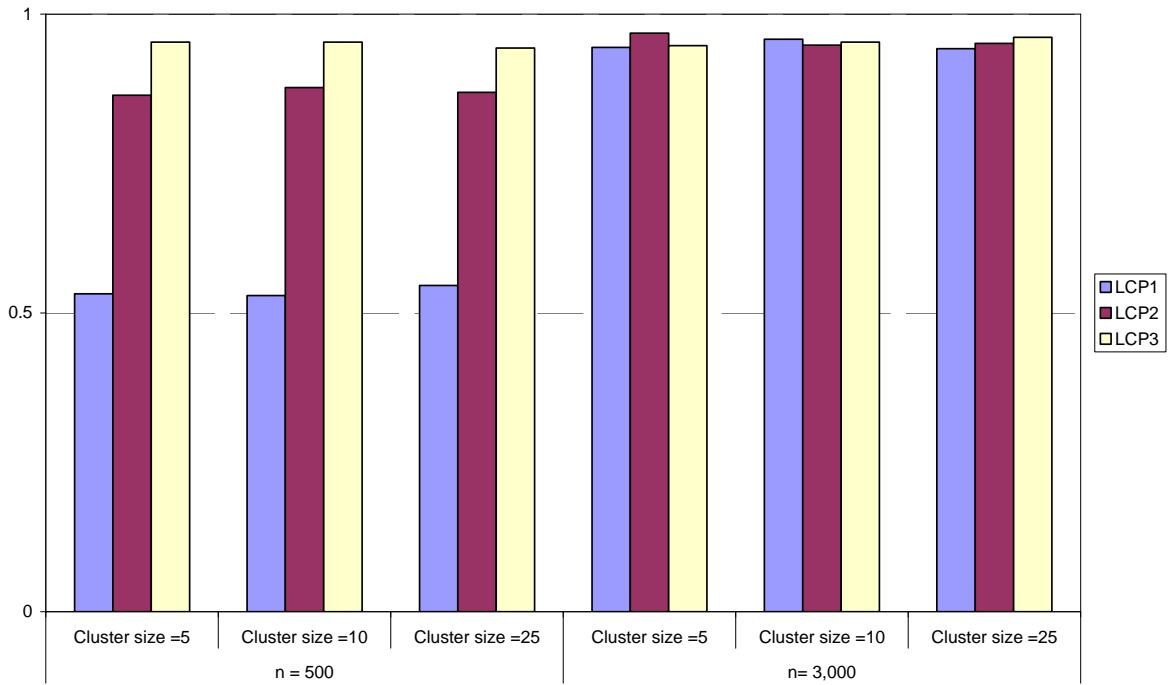
Figure 1. Accuracy for equally weighted groups with item-specific error rate of .05 for Pearson chi-square (X^2)
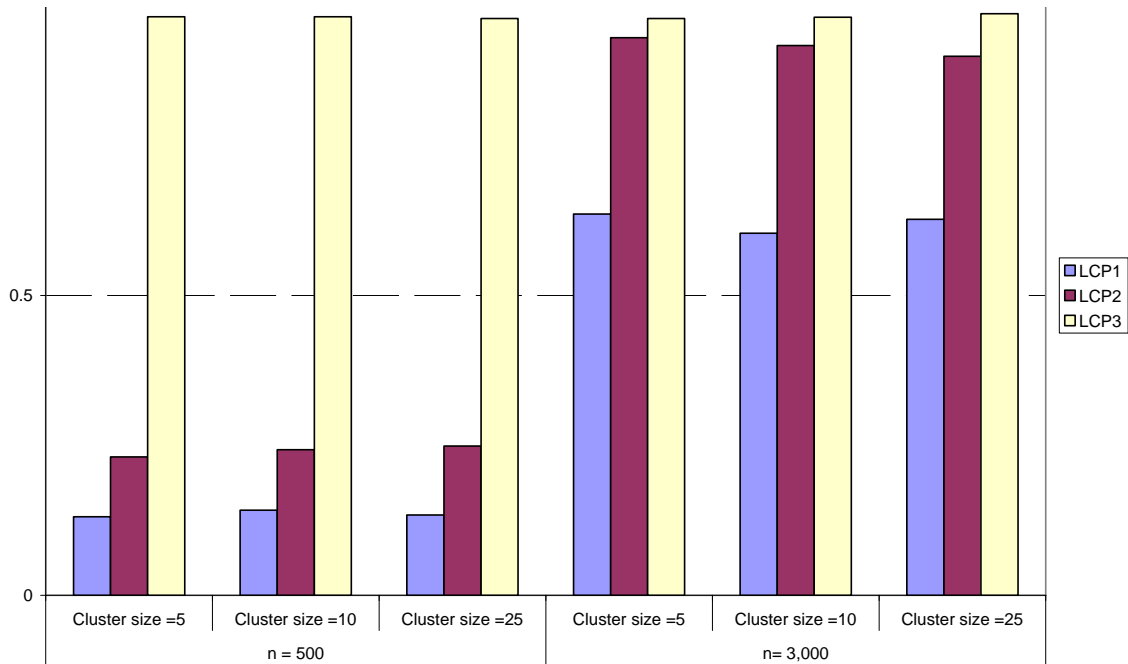
Figure 2. Accuracy for equally weighted groups with item-specific error rate of .25 for Pearson chi-square (X^2)

The results indicate that the accuracy of X^2, G^2 and I^2 were unaffected by the cluster size. Figures 3, 4 and 5 demonstrate little difference in the results based on cluster size. The figures show that the model comparison statistics performed well when item-specific error rates were low and sample sizes were large.

Figure 3. Accuracy for equally weighted groups with cluster size of 5 for Pearson chi-square (X^2)

Figure 4. Accuracy for equally weighted groups with cluster size of 10 for Pearson chi-square (X^2)
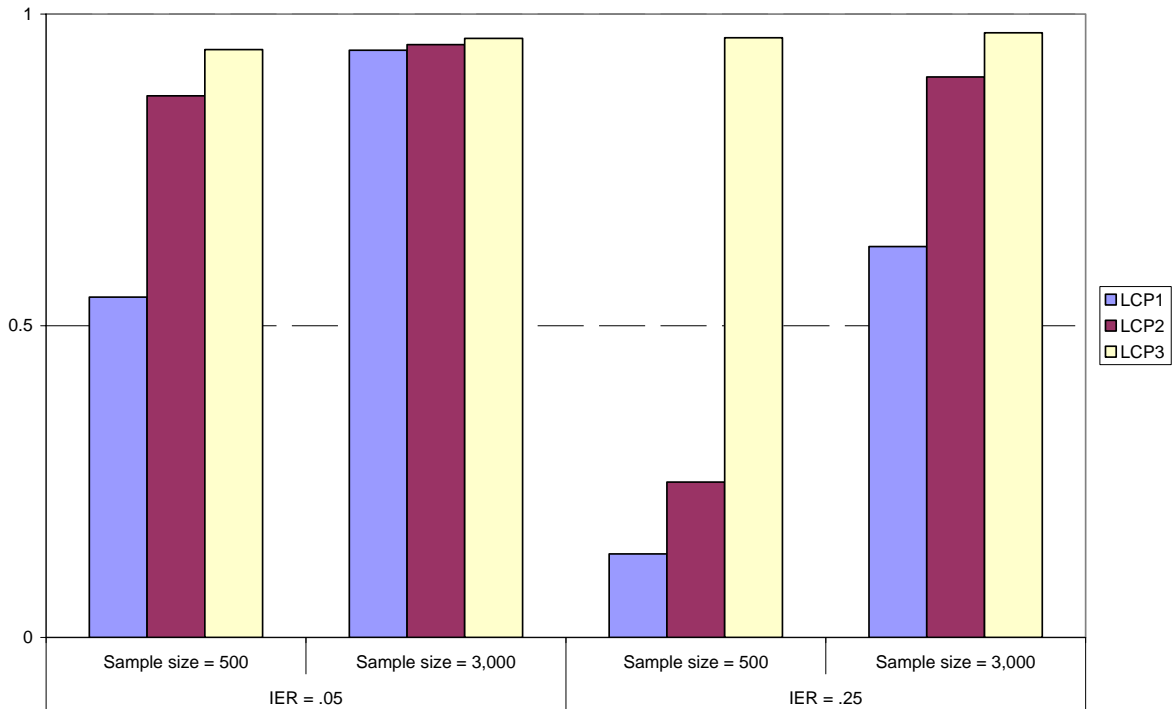
Figure 5. Accuracy for equally weighted groups with cluster size of 25 for Pearson chi-square (X^2)

Next, the results by LC pattern used were examined. As noted earlier, three patterns of LC proportions with differing levels of variability were tested. In other words, the difference between group LC proportions becomes more obvious between LC patterns one through three.

LC variability was examined in terms of sample size. Figures 6, 7 and 8 show the results for each of the LC patterns when group sample sizes were small. In instances where the first and second pattern of LC proportions were present, the model comparison statistics were more accurate when item-specific error rates were low. However, the model comparison statistics did not produce satisfactory results when the item-specific error rates were high. Simulations involving the third pattern of LC proportions resulted in accurate model identification despite the item-specific error rates introduced.
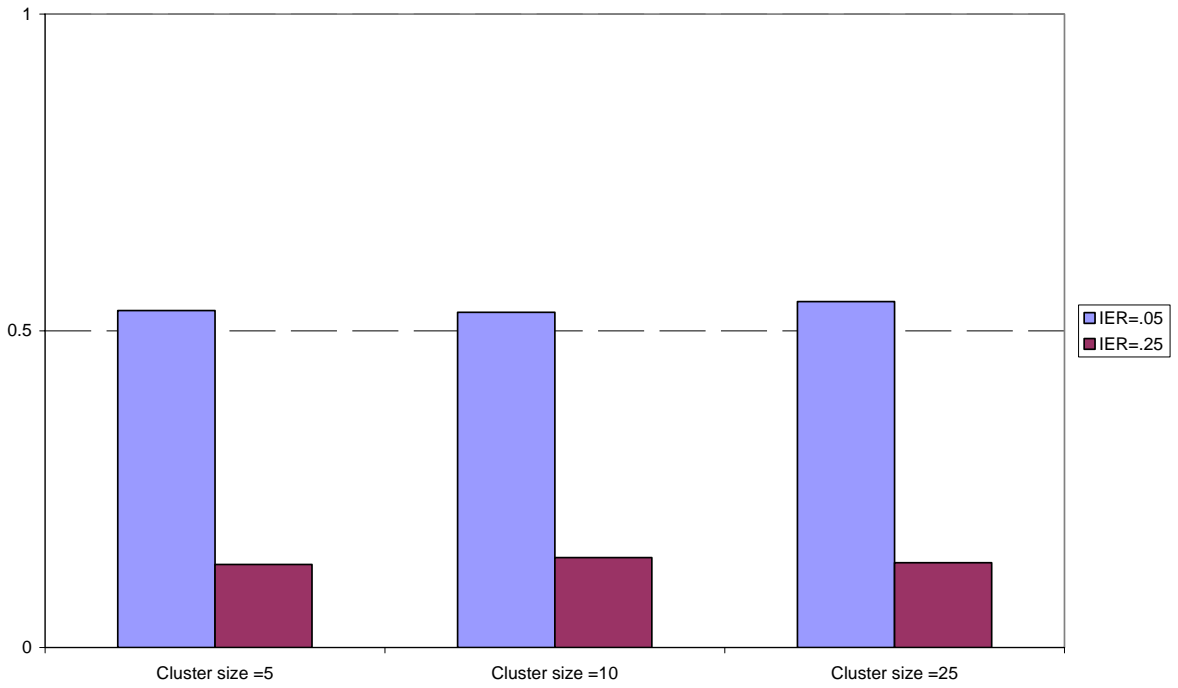
Figure 6. Accuracy for equally weighted groups with first LC pattern and sample size of 500 for Pearson chi-square (X^2)
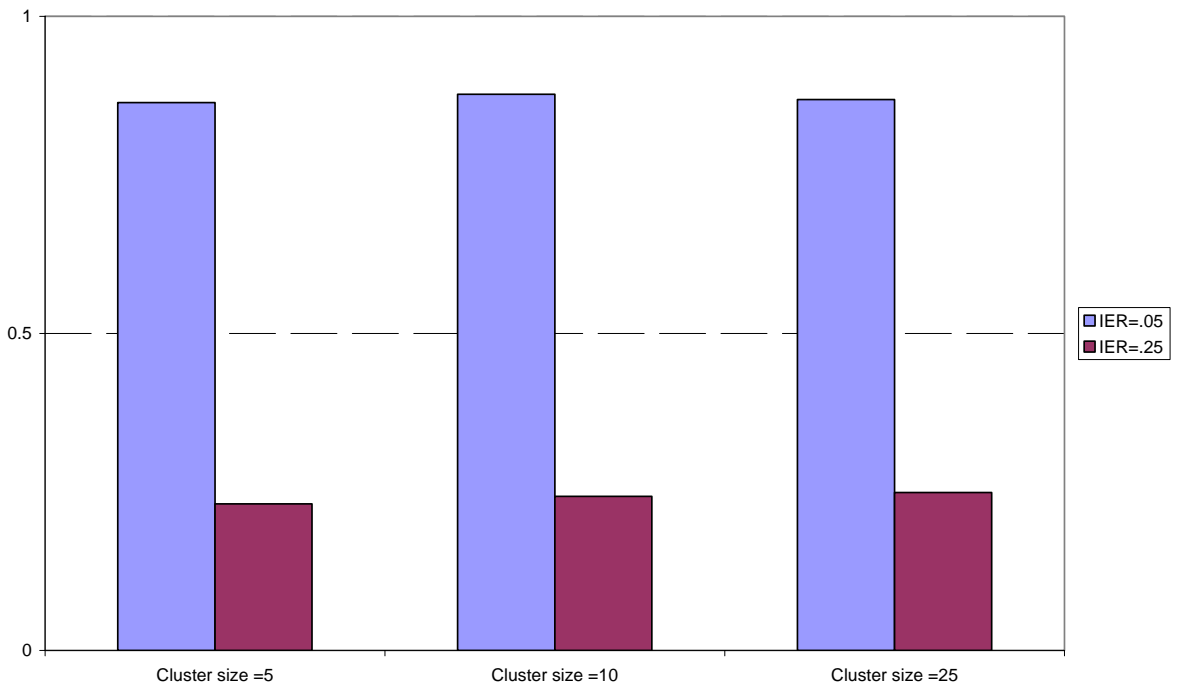
Figure 7. Accuracy for equally weighted groups with second LC pattern and sample size of 500 for Pearson chi-square (X^2)
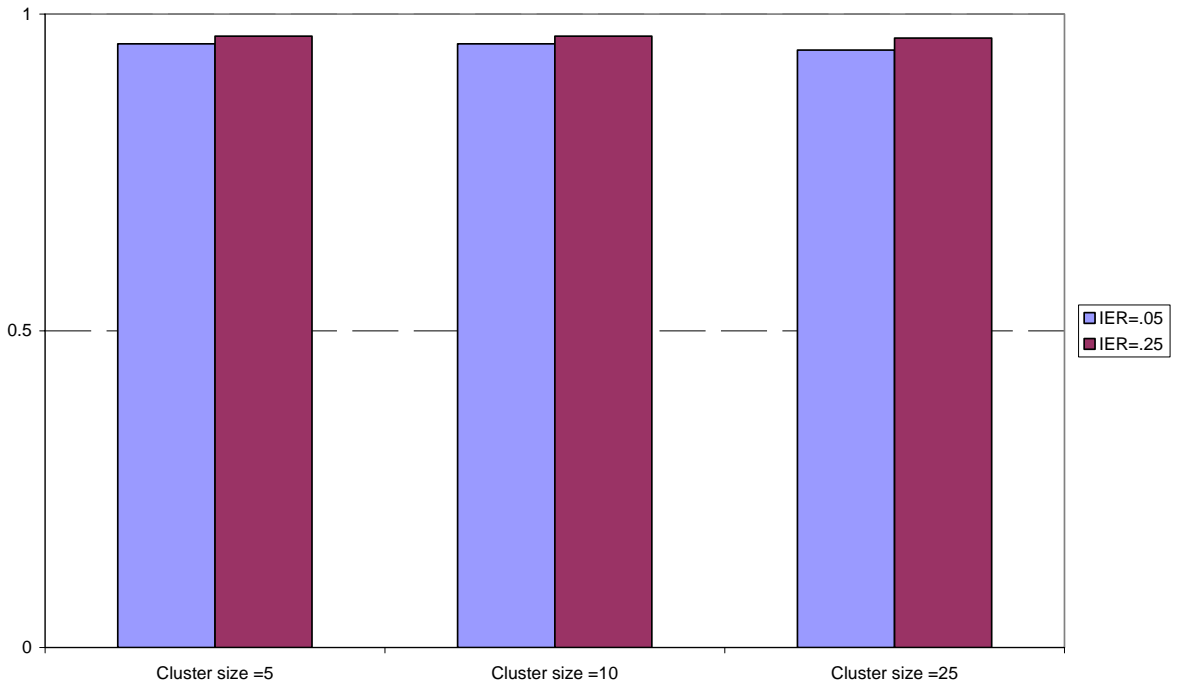
Figure 8. Accuracy for equally weighted groups with third LC pattern and sample size of 500 for Pearson chi-square (X^2)

Figures 9, 10 and 11 show the results for each of the LC patterns when group sample sizes were large. In each case, the LC patterns used did not greatly affect the model comparison statistics' accuracy.
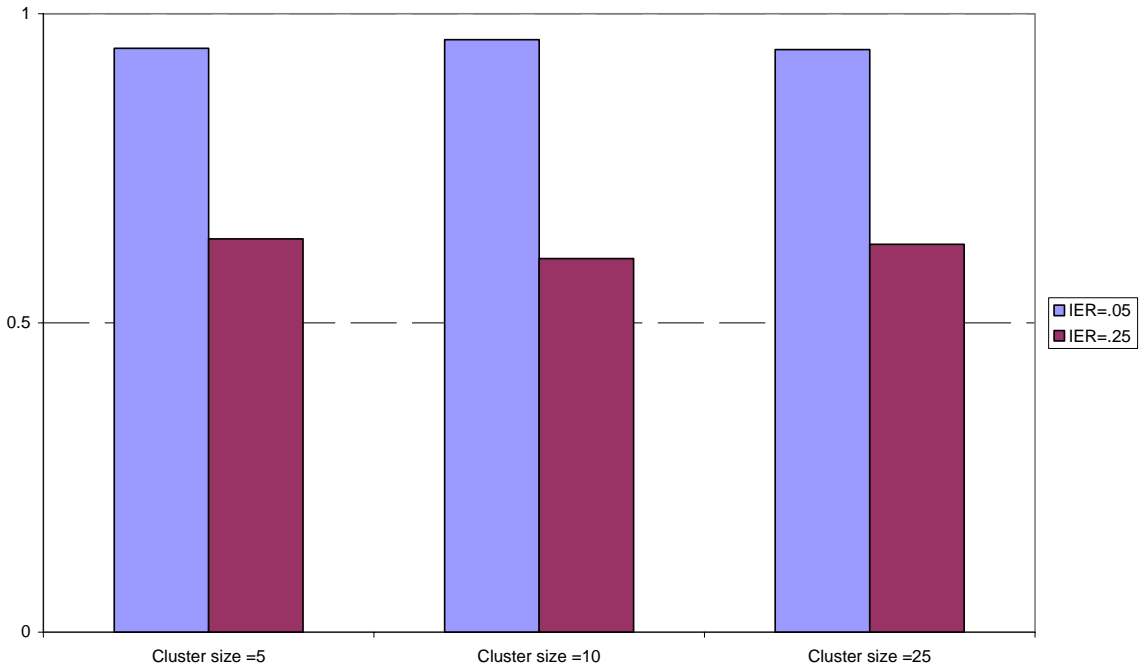
Figure 9. Accuracy for equally weighted groups with first LC pattern and sample size of 3,000 for Pearson chi-square (X^2)
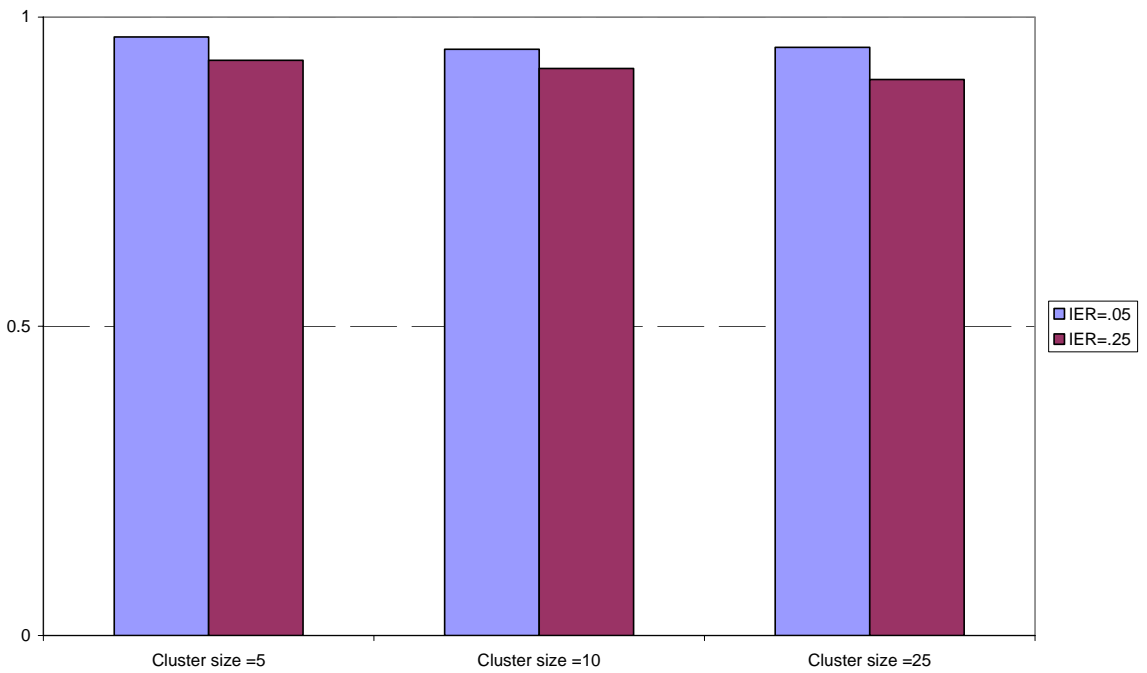
Figure 10. Accuracy for equally weighted groups with second LC pattern and sample size of 3,000 for Pearson chi-square (X^2)
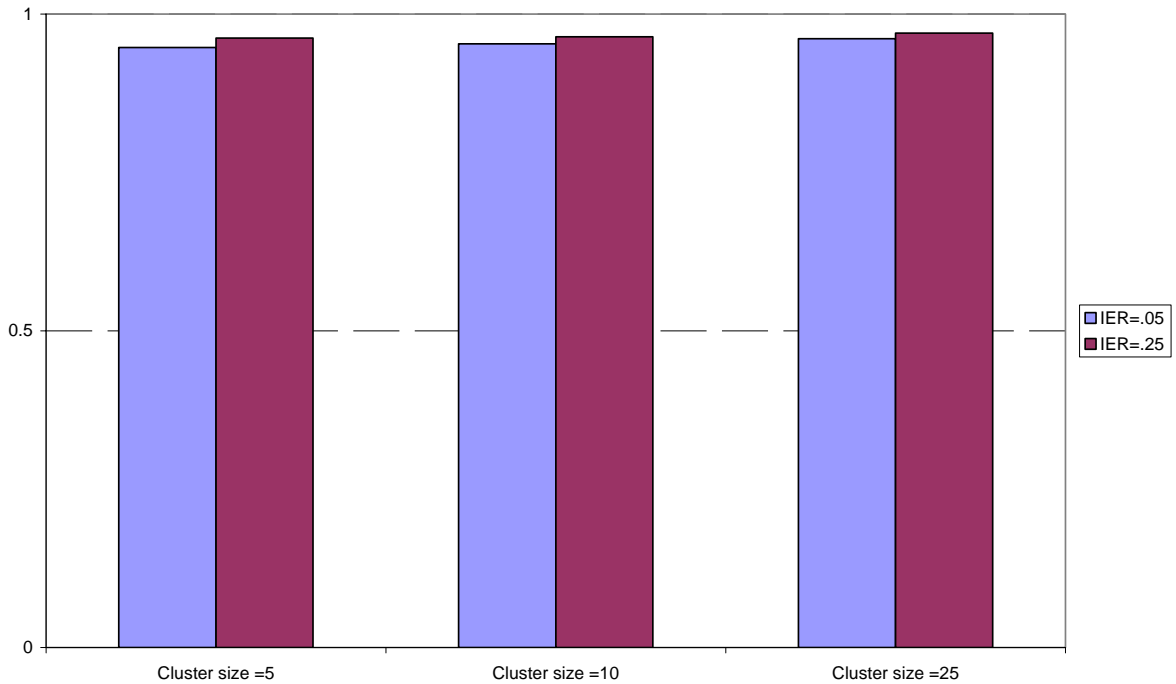
Figure 11. Accuracy for equally weighted groups with third LC pattern and sample size of 3,000 for Pearson chi-square (X^2)

In summary:

- As expected, sample size was an important factor in the accuracy of X^2, G^2 and I^2. Typically, model comparison statistics were more accurate for large sample sizes.

- Also, as expected, model comparison statistics were more accurate for the third pattern of LC proportions.

- Item-specific error rates did not have a substantial effect on the accuracy of X^2, G^2 and I^2 for large sample sizes. When sample sizes were small, the model comparison statistics were not accurate under all error rate conditions.

- Cluster size (i.e., ICC) had virtually no effect on the accuracy of X^2, G^2 and I^2.

<u>Accuracy for Unequally Weighted Groups for X^{2*}, G^{2*} and I^{2*}</u>

In addition to the other factors considered in the simulated cases, the effect of unequal sampling was examined. Sampling fractions were used that selected respondents following a 5:1 ratio, favoring group 1. Unequally weighted observed and expected values were used to calculate X^{2*}, G^{2*} and I^{2*} that can be considered pseudo-chi-square values.

When using equally weighted groups, high levels of accuracy were found in all of the cases where the sample sizes were large. When using unequally weighted groups, as shown in Table 12, X^{2*}, G^{2*} and I^{2*} still were more accurate with large sample sizes. However, some of the other factors being evaluated in the simulated cases appeared to interact with the sample size and affect the accuracy of the model comparison statistics.

For instance, when item-specific error rates were examined by sample size, some clear patterns were found. Figure 12 demonstrates that X^{2*}, G^{2*} and I^{2*} was more accurate at detecting the correct model when item-specific error rates were low and sample sizes were large. In instances where the opposite occurred (i.e., sample sizes were small and item-specific error rates were high) the model comparison statistics demonstrated high levels of accuracy in only three of nine cases.

Table 12. Accuracy for unequally weighted groups by sample size
with .05 significance level

Sample Size	Cluster Size	LC Pattern	Item Error Rate	X^{2*}	G^{2*}	I^{2*}
500	5	1	.05	.161	.194	.173
			.25	.031	.041	.034
		2	.05	.638	.641	.644
			.25	.071	.072	.069
		3	.05	.953	.974	.966
			.25	.937	.928	.938
	10	1	.05	.148	.187	.163
			.25	.025	.042	.027
		2	.05	.642	.624	.636
			.25	.087	.085	.086
		3	.05	.953	.963	.963
			.25	.931	.911	.927
	25	1	.05	.177	.220	.187
			.25	.027	.038	.032
		2	.05	.643	.641	.645
			.25	.066	.063	.063
		3	.05	.943	.963	.962
			.25	.919	.911	.917
3000	5	1	.05	.977	.984	.980
			.25	.298	.360	.336
		2	.05	.992	.996	.993
			.25	.697	.701	.708
		3	.05	.972	.975	.973
			.25	.961	.933	.959
	10	1	.05	.975	.981	.979
			.25	.306	.356	.331
		2	.05	.988	.991	.990
			.25	.670	.684	.677
		3	.05	.969	.974	.970
			.25	.972	.934	.963
	25	1	.05	.972	.976	.973
			.25	.310	.368	.314
		2	.05	.990	.994	.992
			.25	.729	.732	.732
		3	.05	.974	.976	.974
			.25	.964	.936	.958

KEY

	< .500
	.501 to .750
	.751 to .900
	> .901
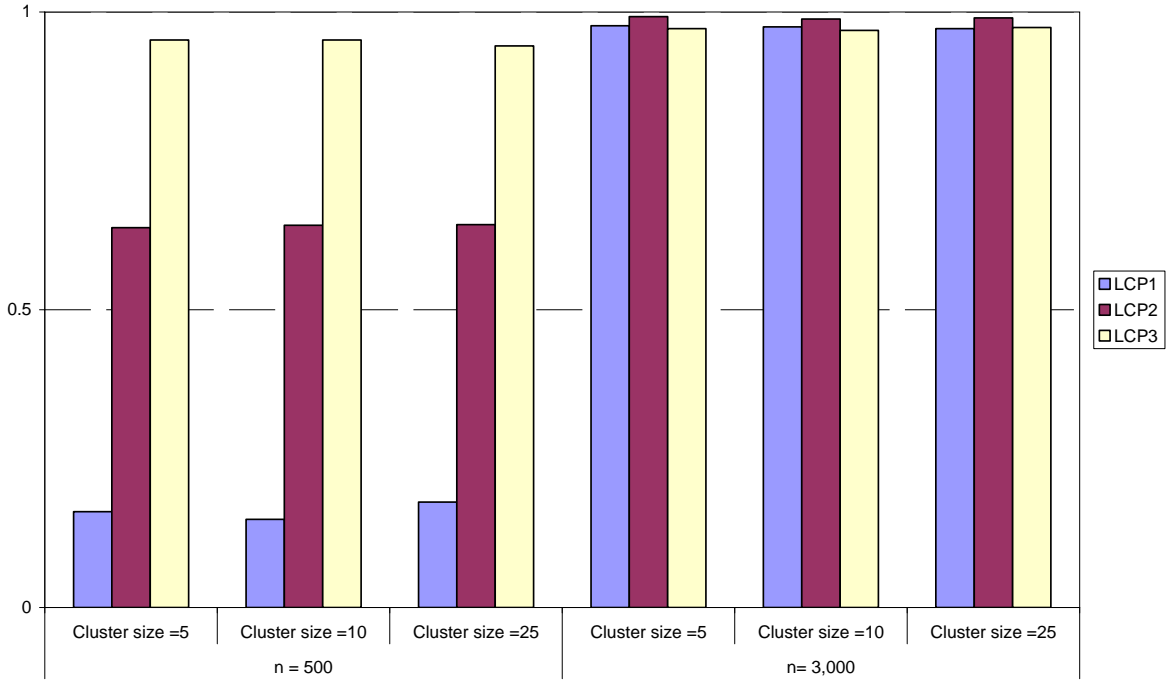
Figure 12. Accuracy for unequally weighted groups with item-specific error rate of .05 for Pseudo-Pearson chi-square (X^{2*})
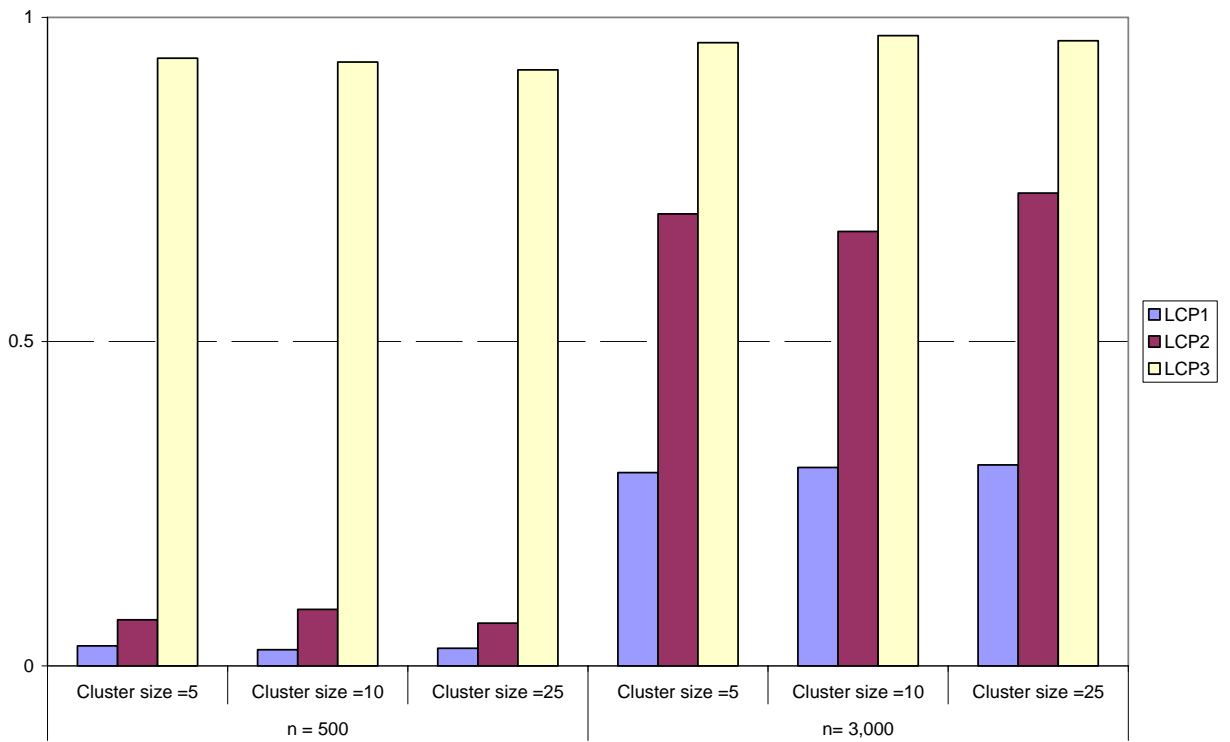
Figure 13. Accuracy for unequally weighted groups with item-specific error rate .25 for Pseudo-Pearson chi-square (X^{2*})
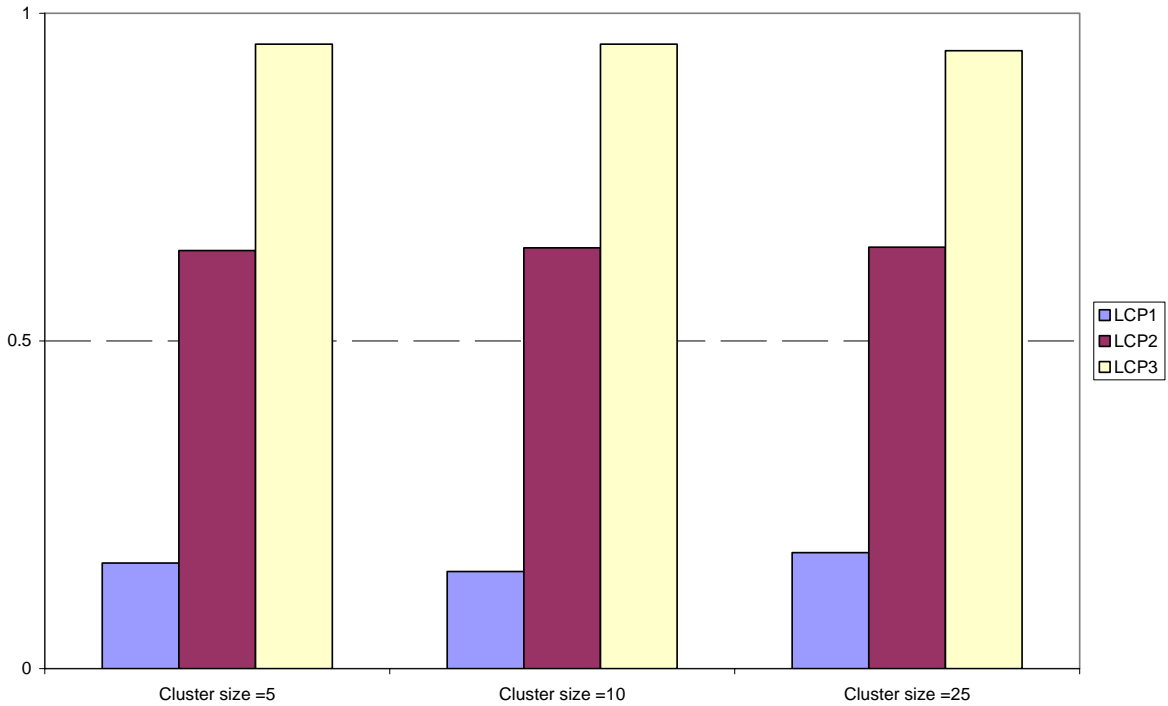
Figure 14. Accuracy for unequally weighted groups with sample size of 500 and item-specific error rate of .05 for Pseudo-Pearson chi-square (X^{2*})

No clear patterns emerged when the results were examined by cluster size. In the instances when item-specific error rates were high, some improvement was noted as cluster size increased, but no obvious pattern was observed.

The LC pattern used appears to have had some effect on the accuracy of the model comparison statistics. As expected X^{2*}, G^{2*} and I^{2*} had greater success when LC proportions were more variable. Moreover, the model comparison statistics were more accurate for large sample sizes. Figure 15, for instance, shows the results for cases with low LC variability and small sample sizes. In this case, the model comparison statistics were not accurate. However, Figures 16 and 17 show high rates of accuracy for large sample sizes and medium to high LC variability.
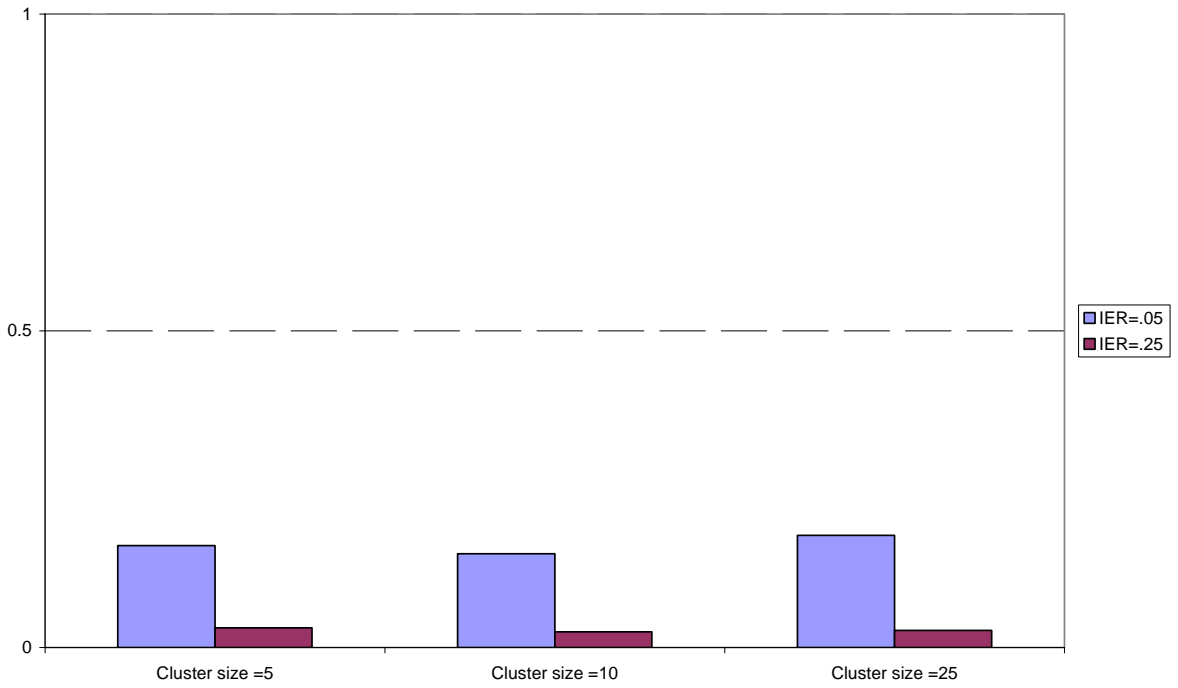
Figure 15. Accuracy for unequally weighted groups with first LC pattern and sample size of 500 for Pearson chi-square (X^{2*})
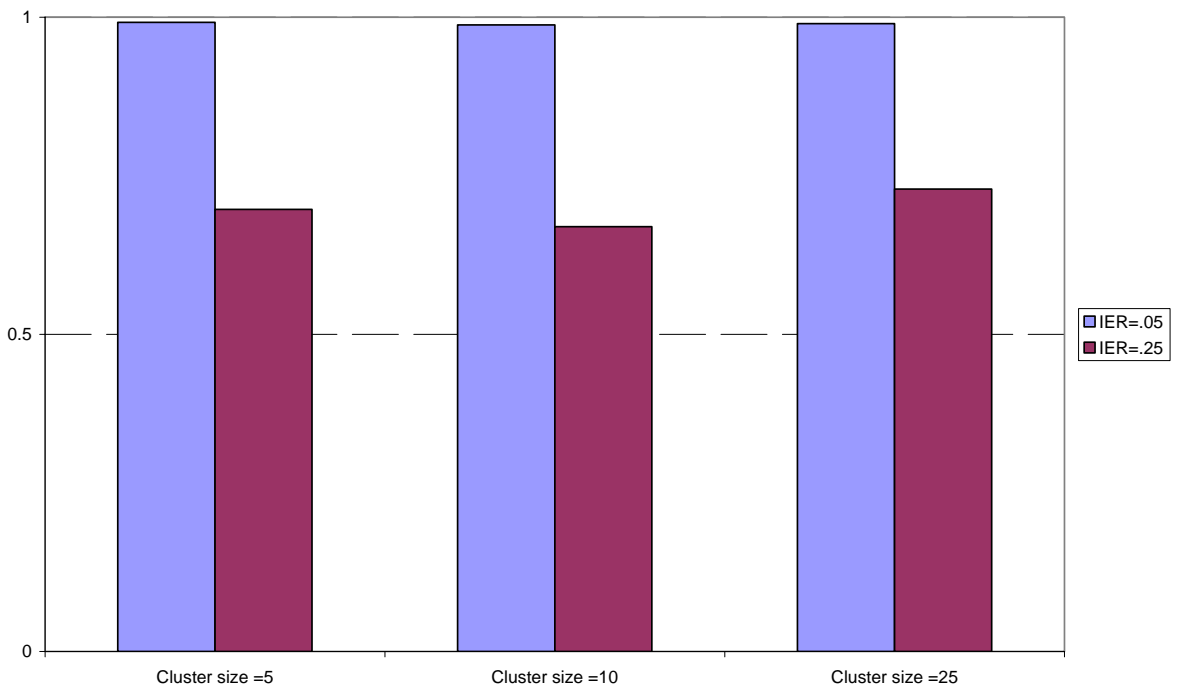
Figure 16. Accuracy for unequally weighted groups with second LC pattern and sample size of 3,000 for Pearson chi-square (X^{2*})
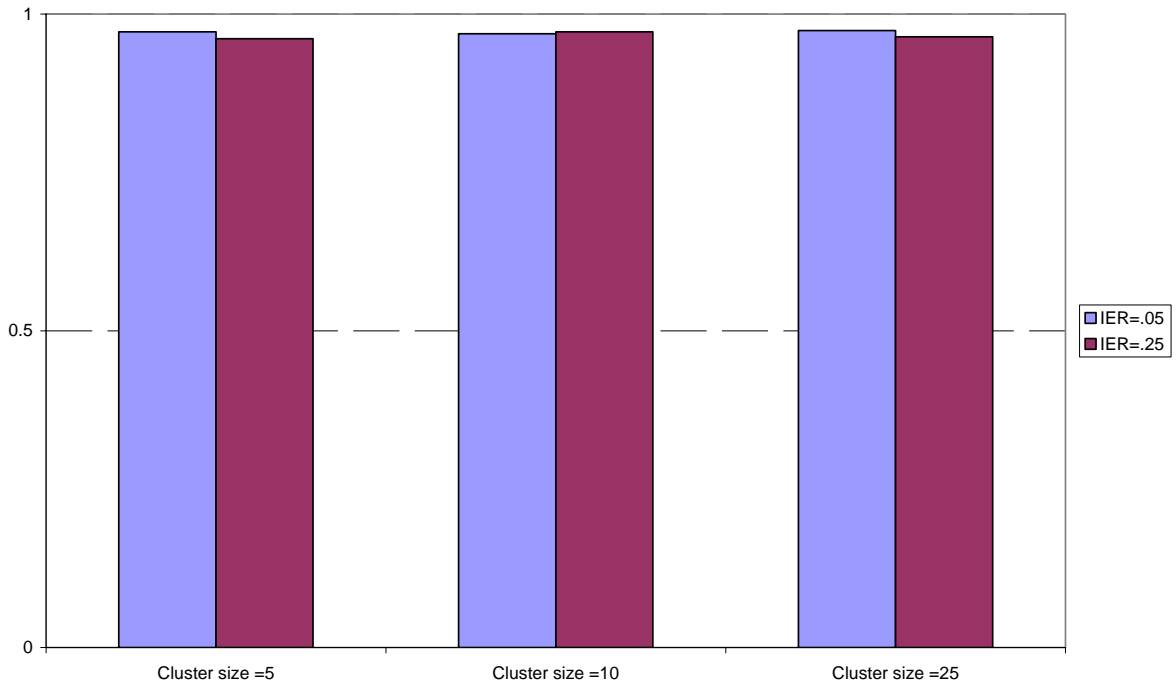
Figure 17. Accuracy for unequally weighted groups with third LC pattern and sample size of 3,000 for Pearson chi-square (X^{2*})

Also, item-specific error rates appeared to interact with these other factors in affecting the accuracy of the model comparison statistics. Figure 18 shows results for small sample sizes and medium LC variability. The model comparison statistics were only accurate in these instances when the item-specific error rates were low. This was also the case in Figure 19 where there was low LC variability and large sample sizes. Even though the group differences in LC proportions were less variable and the sample sizes were large, the same pattern was observed where the model comparison statistics were accurate only when item-specific error rates were low. Thus, the model comparison statistics had difficulty accurately detecting the correct model when sample sizes were small, item-specific error rates were high and differences in LC patterns were less variable.
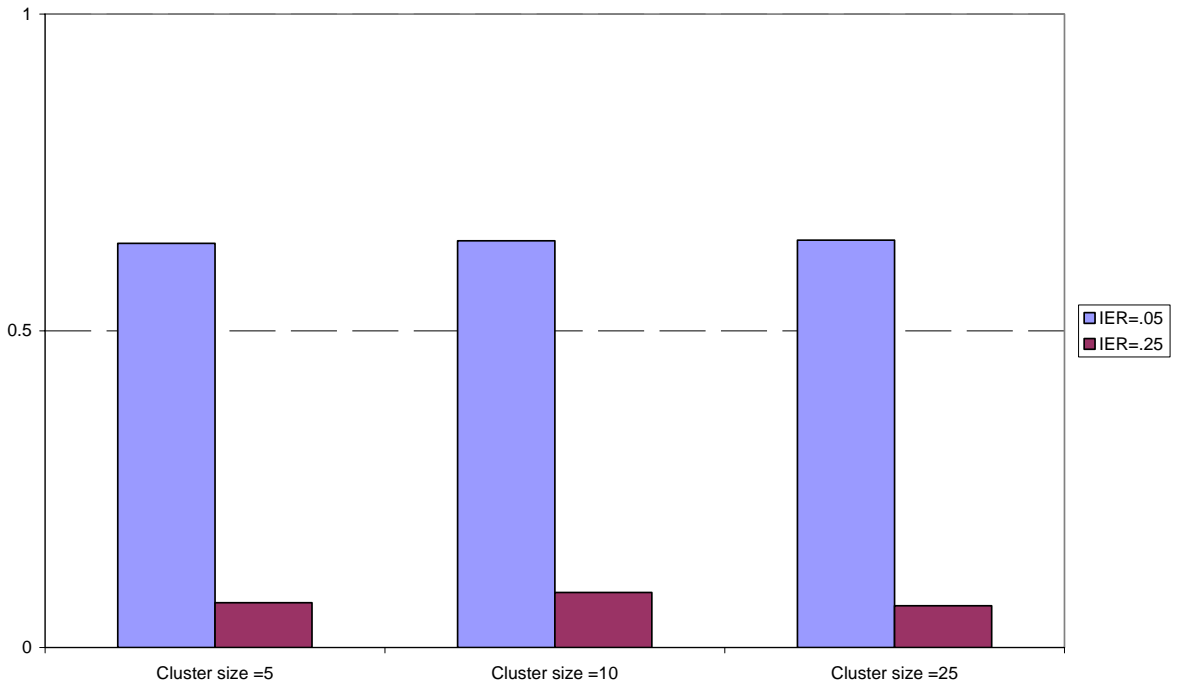
Figure 18. Accuracy for unequally weighted groups with second LC pattern and sample size of 500 for Pearson chi-square (X^{2*})
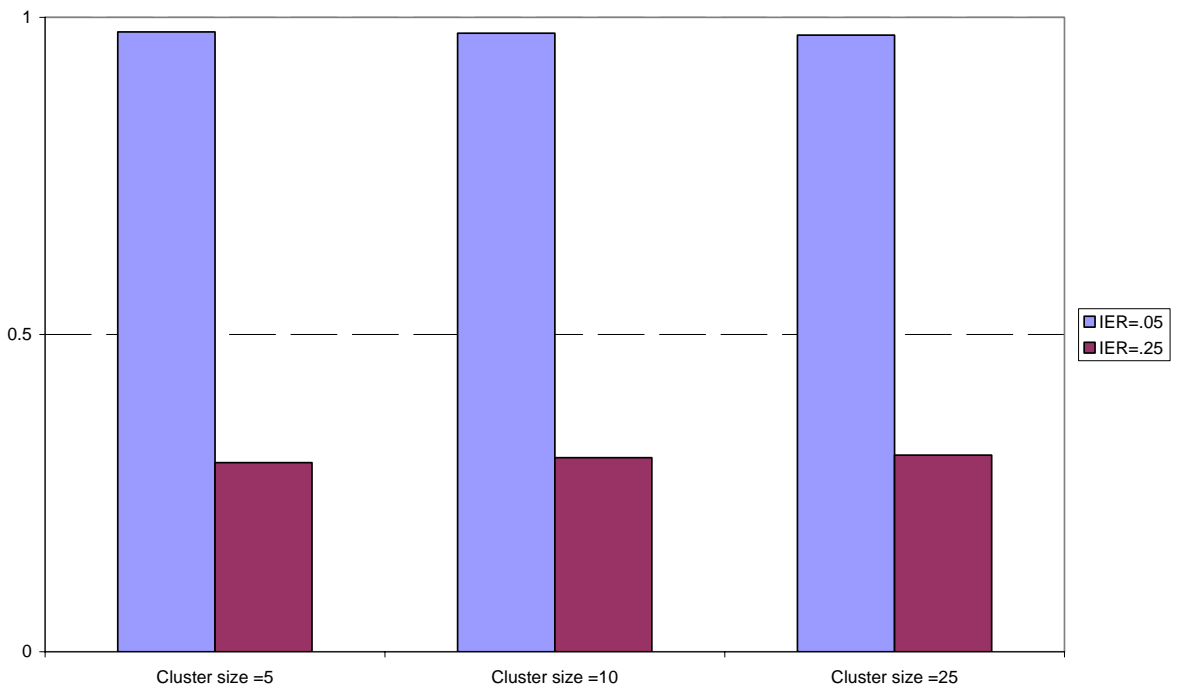
Figure 19. Accuracy for unequally weighted groups with first LC pattern and sample size of 3,000 for Pearson chi-square (X^{2*})

In summary:

- Sample size was an important factor in the accuracy of X^{2*}, G^{2*} and I^{2*} when using unequally weighted groups. X^{2*}, G^{2*} and I^{2*} were more accurate in detecting the correct model, especially when item-specific error rates were low and sample sizes were large.

- LC proportions of the unequally weighted groups had some effect on the accuracy of the model comparison statistics. However, this occurred mainly in combination with other important factors, such as sample size and item-specific error rates.

- Cluster size (ICC) only had a moderate effect on the accuracy of X^{2*}, G^{2*} and I^{2*} when high item-specific error rates were present in the data.

Overall Accuracy for X^2, G^2 and I^2

In examining the results from both equally and unequally weighted groups, several similarities and differences were found (Figure 20):

- All three chi-square and pseudo-chi-square model comparison statistics performed similarly.

- Sample size was an important factor in the accuracy of X^2, G^2 and I^2.

- The model comparison statistics were not accurate in detecting the correct model in some instances regardless of whether groups were equally or unequally weighted. When sample sizes were small, item-specific error rates were high and LC variability was low, the model comparison statistics did not perform well.

- When sample sizes were large, the model comparison statistics performed consistently well with equally weighted groups. This was not necessarily the case for the unequally weighted groups. When item-specific error rates were high, the model comparison statistics continued to perform poorly even in cases with large sample sizes.

- Cluster size (ICC) alone had little effect on the accuracy of X^2, G^2 and I^2 whether groups were equally or unequally weighted.
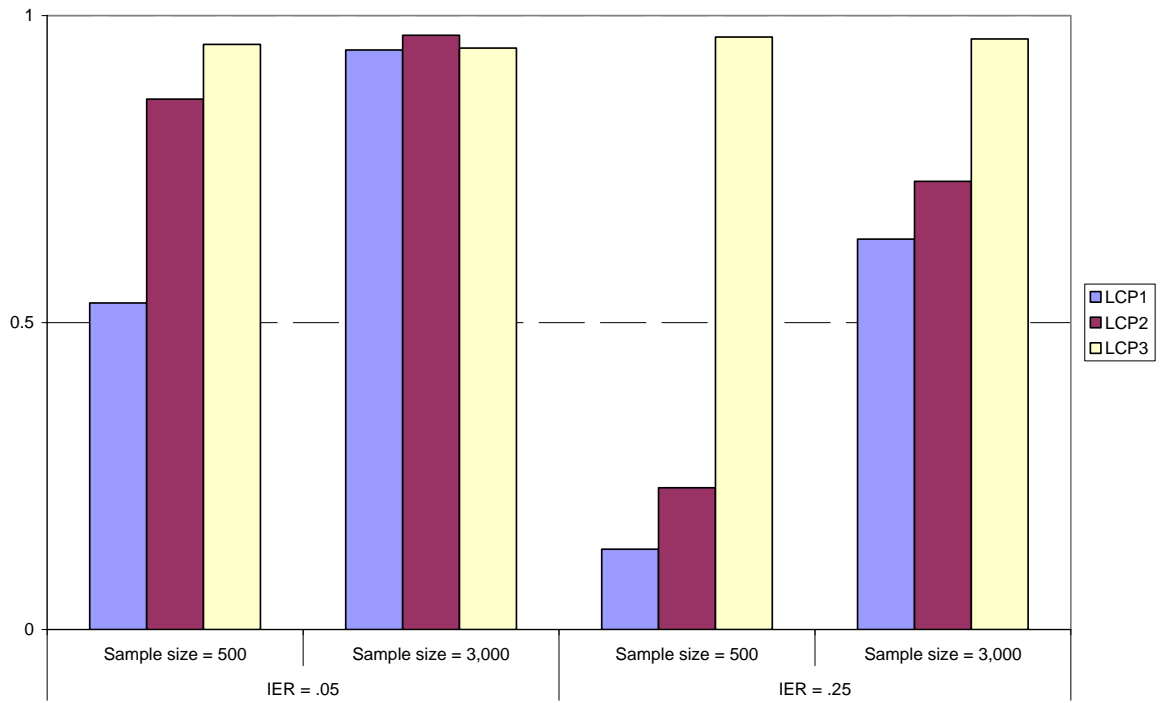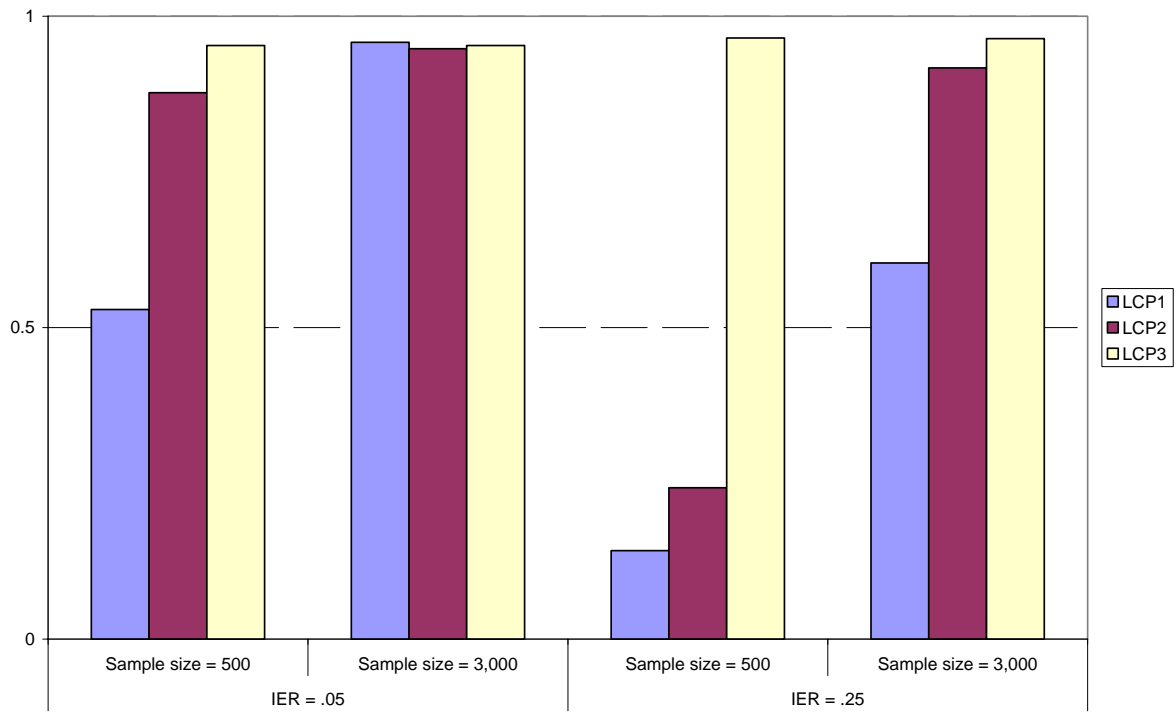

Figure 20. Accuracy for equally and unequally weighted groups with cluster size of 25 for Pearson chi-square (X^2)

Accuracy for Equally Weighted Groups for AIC, BIC and RIC

As with the chi-square statistics, the accuracy of the three model comparison statistics was assessed by varying the sample size of comparison groups between 500 and 3000 respondents. Table 13 shows the proportion of times out of 1000 where AIC, BIC and RIC selected the correct model. The highlighted cells indicate a proportion greater than .5. Darker shading indicates higher accuracy levels and the darkest cells indicate correct model selection occurred more than 90 percent of the time. Additional tables are available in Appendix A.

AIC demonstrated higher levels of accuracy in more cases than BIC and RIC. Although RIC did not perform as well as AIC, it demonstrated high accuracy in more cases than BIC. AIC was accurate in most cases with large sample sizes, while BIC and RIC showed relatively high accuracy in two-thirds of the cases. When sample sizes were small, similar levels of accuracy were observed in only two-thirds of the cases for AIC, half of the cases for RIC and one-third of the cases for BIC. When the statistics were inaccurate, 71 percent of the inaccurate results for AIC resulted in the selection of the homogeneous model. Ninety-six percent of the inaccurate results for BIC and 94 percent for RIC resulted in the selection of the homogeneous model.

Table 13. Accuracy for equally weighted groups for AIC, BIC and RIC

Sample Size	Cluster Size	LC Pattern	Item Specific Error Rate	AIC	BIC	RIC
500	5	1	.05	.569	.004	.157
			.25	.208	.000	.019
		2	.05	.814	.111	.637
			.25	.305	.000	.042
		3	.05	.901	1.0	.994
			.25	.852	.661	.967
	10	1	.05	.562	.007	.188
			.25	.210	.000	.024
		2	.05	.835	.112	.635
			.25	.319	.000	.043
		3	.05	.892	1.0	.995
			.25	.880	.666	.968
	25	1	.05	.574	.011	.231
			.25	.201	.000	.015
		2	.05	.841	.129	.662
			.25	.348	.001	.053
		3	.05	.873	1.0	.992
			.25	.871	.659	.968
3000	5	1	.05	.899	.702	.968
			.25	.846	.009	.167
		2	.05	.929	1.0	1.0
			.25	.902	.080	.514
		3	.05	.899	1.0	1.0
			.25	.802	1.0	.987
	10	1	.05	.923	.694	.969
			.25	.866	.004	.160
		2	.05	.916	1.0	1.0
			.25	.877	.082	.494
		3	.05	.902	1.0	1.0
			.25	.781	.999	.988
	25	1	.05	.905	.713	.964
			.25	.871	.008	.185
		2	.05	.910	1.0	1.0
			.25	.865	.044	.337
		3	.05	.916	1.0	1.0
			.25	.766	1.0	.991

KEY

☐ < .500

▫ .501 to .750

▪ .751 to .900

■ > .901

The item-specific error rates used also affected the accuracy of AIC, BIC and

RIC. AIC was accurate in all cases where item-specific error rates were low. In contrast,

when the item-specific error rates were high, as shown in Figure 21, AIC was only highly

accurate in cases where sample sizes were large and the differences in LC proportions were more variable.
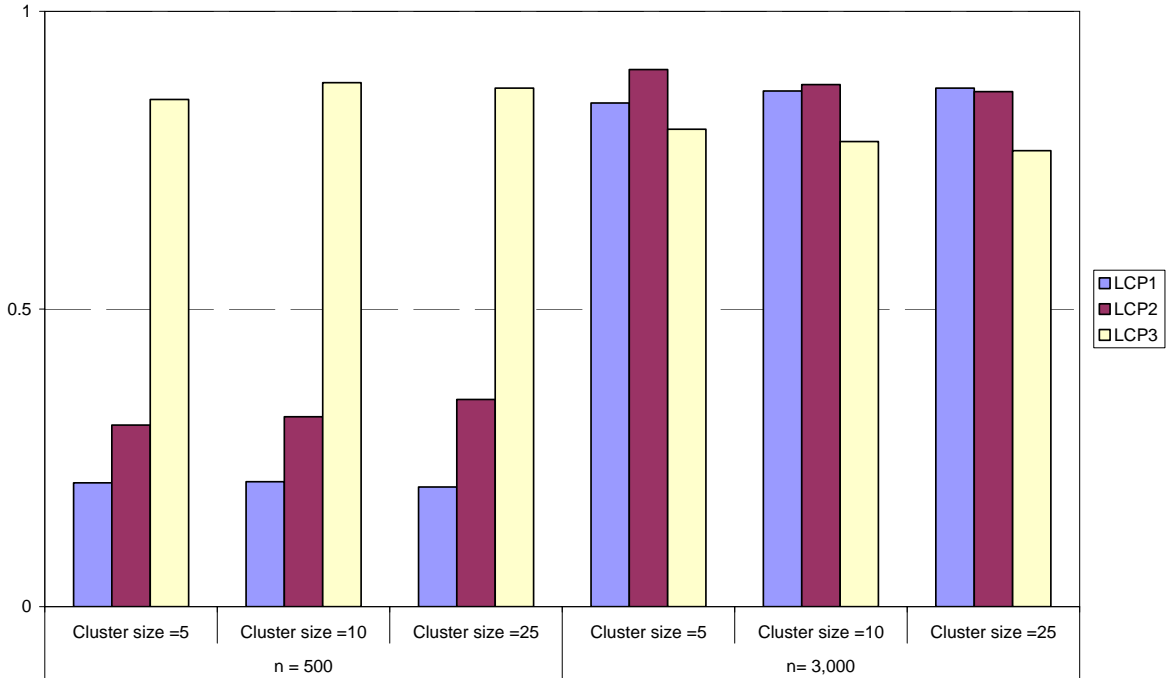
Figure 21. Accuracy for equally weighted groups with item-specific error rate of .25 for AIC

RIC was accurate in detecting the correct model in all cases where item-specific error rates were low and also sample sizes were large. When item-specific error rates were high, RIC was only accurate in cases with high LC variability.
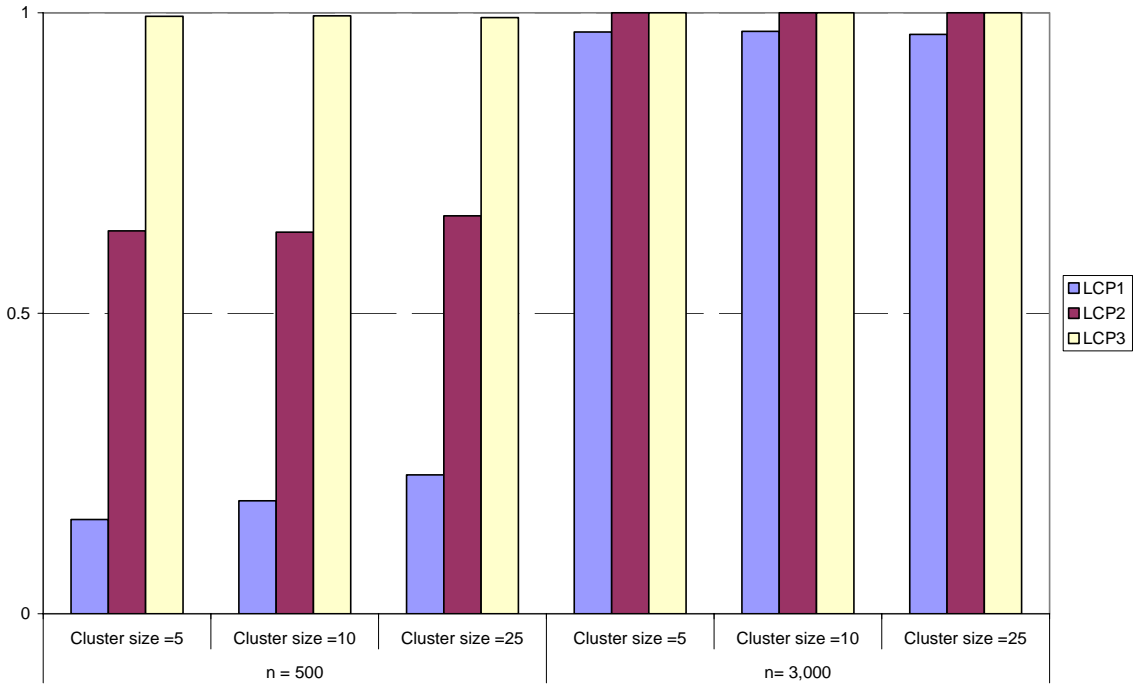
Figure 22. Accuracy for equally weighted groups with item-specific error rate of .05 for RIC

Similar to RIC, BIC was accurate in all cases where sample sizes were large and item-specific error rates were low. However, in cases where sample sizes were small and/or item-specific error rates were high, BIC was only accurate when there was low LC variability.
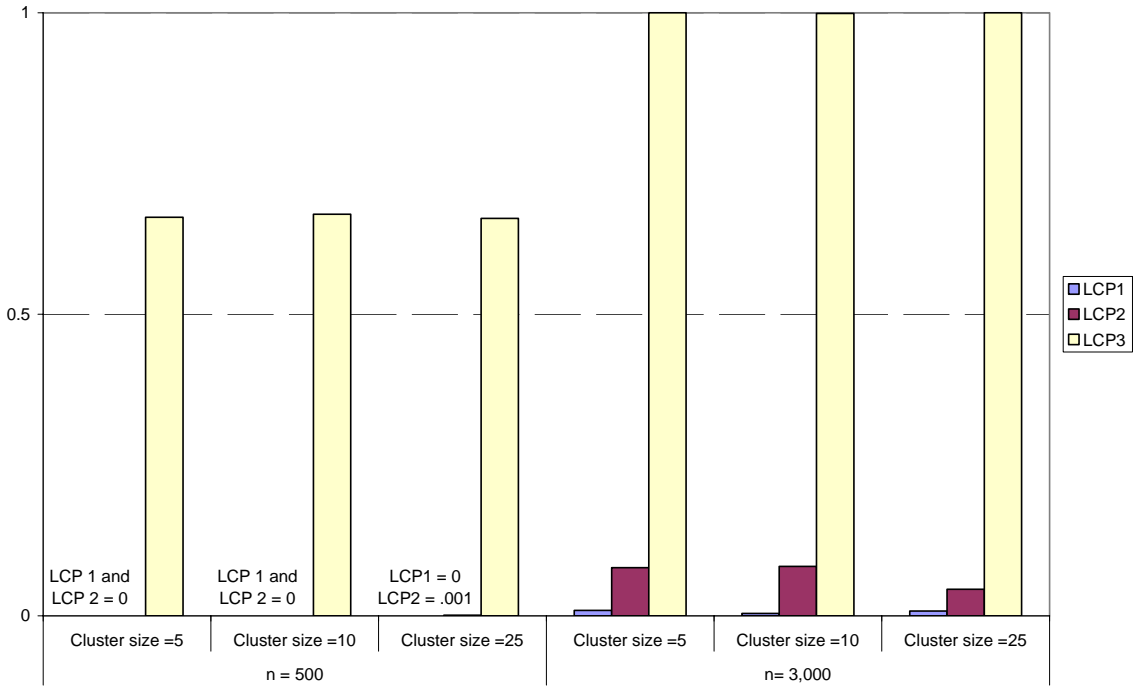
Figure 23. Accuracy for equally weighted groups with item-specific error rate of .25 for BIC

Results did not vary by cluster size. AIC, BIC and RIC produced mixed results depending on other factors unrelated to cluster size.
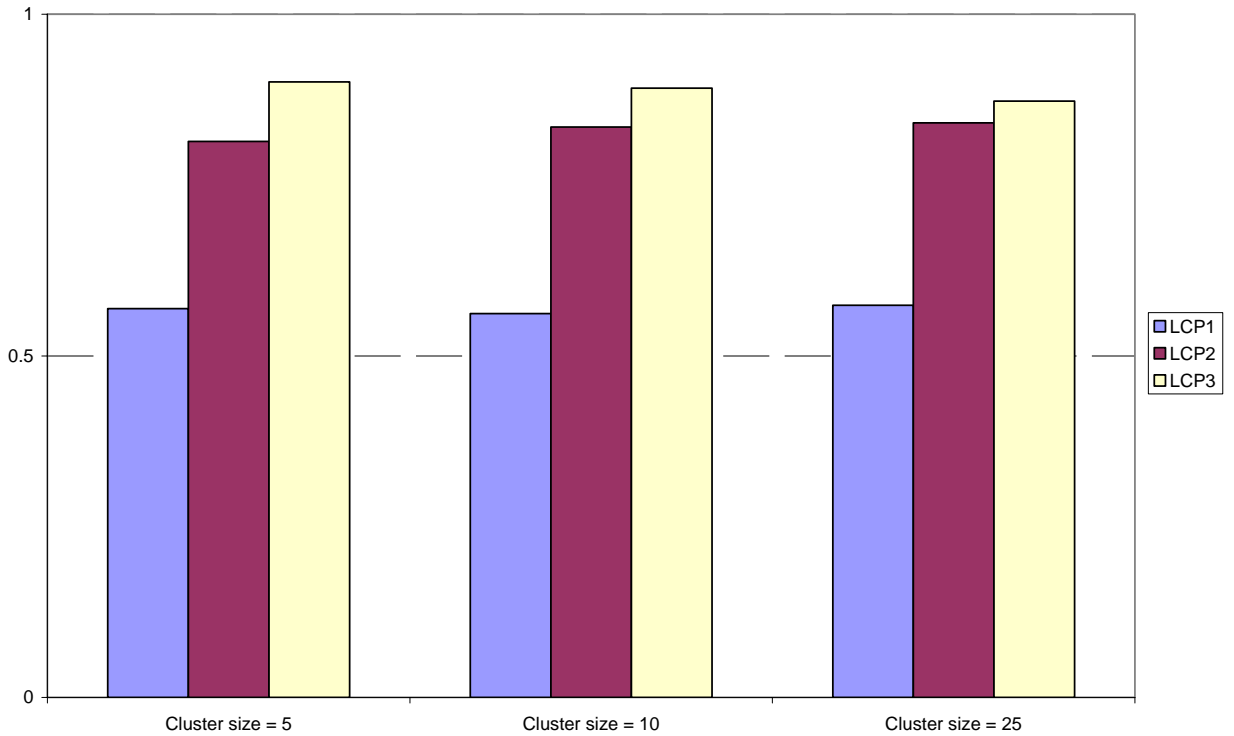
Figure 24. Accuracy for equally weighted groups with sample size of 500 and item-specific error rate of .05 for AIC

As expected, the model comparison statistics demonstrated higher levels of accuracy with high LC variability. Sample size in combination with LC variability were the most important factors in determining the accuracy of the model comparison statistics. In cases where the patterns of LC proportions were more variable, AIC demonstrated high levels of accuracy in all cases where sample sizes were large. BIC and RIC were often more accurate for large sample sizes and low item-specific error rates.
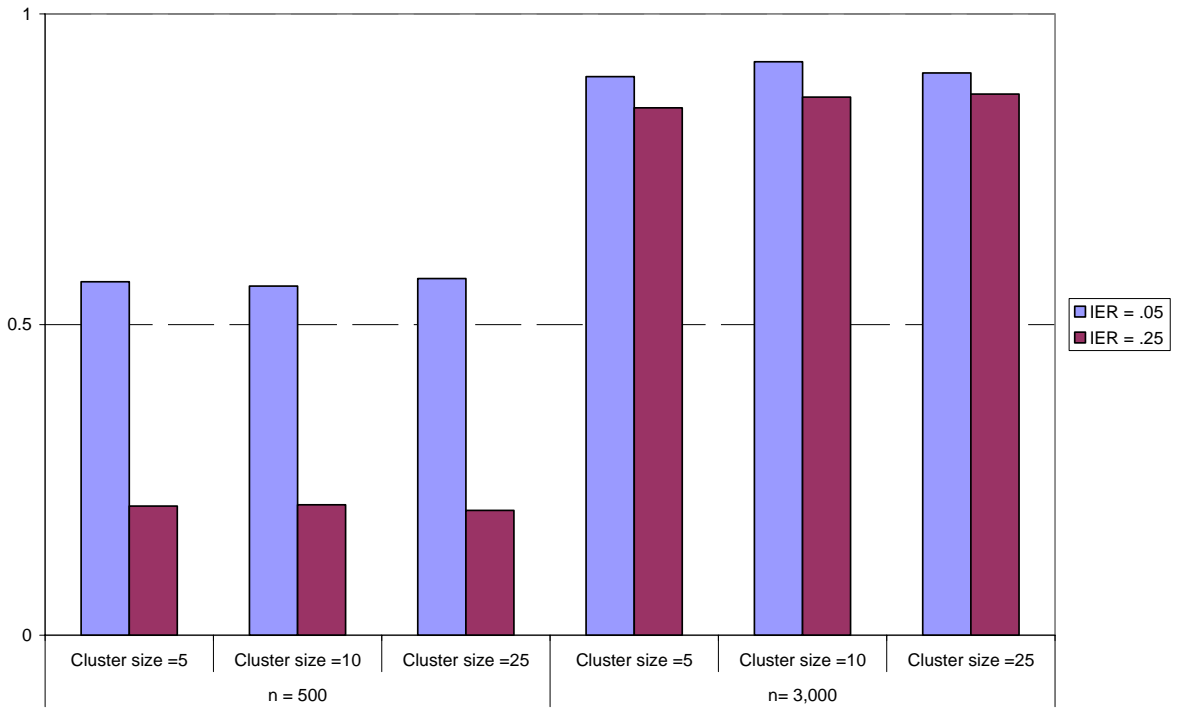
Figure 25. Accuracy for equally weighted groups with first LC pattern for AIC
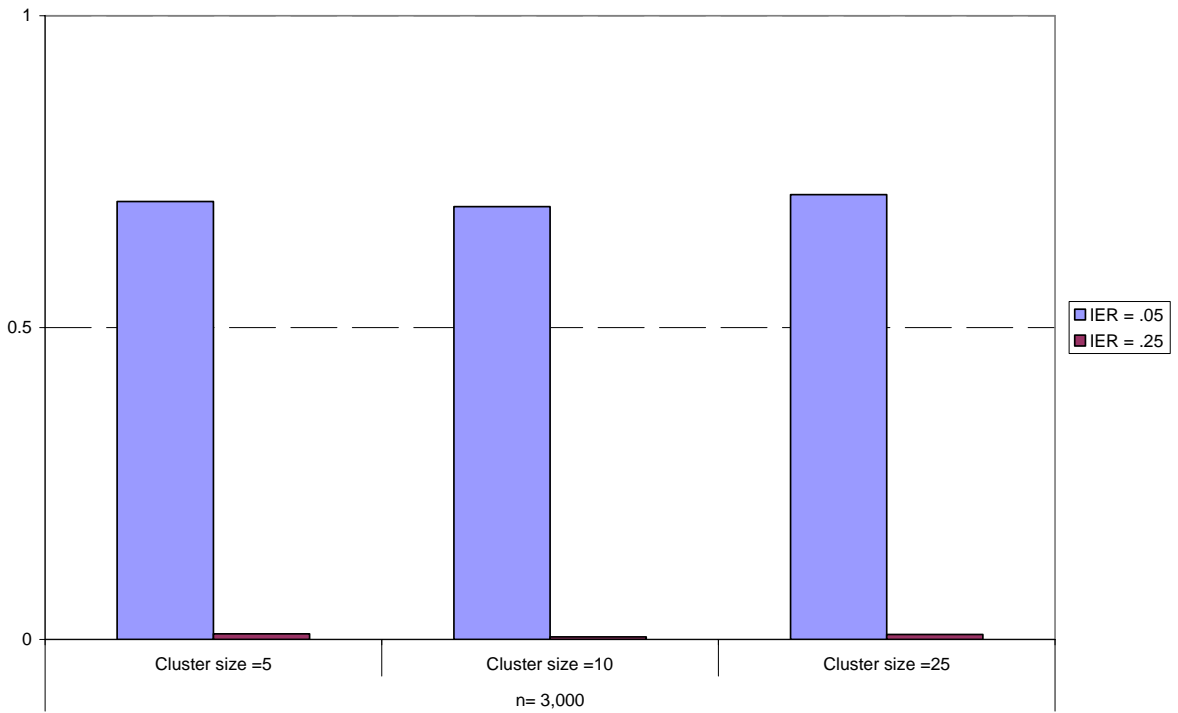
Figure 26. Accuracy for equally weighted groups with first LC pattern and sample size of 3,000 for BIC

As the LC variability became higher, RIC and BIC improved in accuracy. RIC was more accurate in cases where the item-specific error rates were low, despite the sample size. BIC only demonstrated high levels of accuracy in cases where sample sizes were large and item-specific error rates were low. All three model comparison statistics were accurate in the cases where LC variability was high.
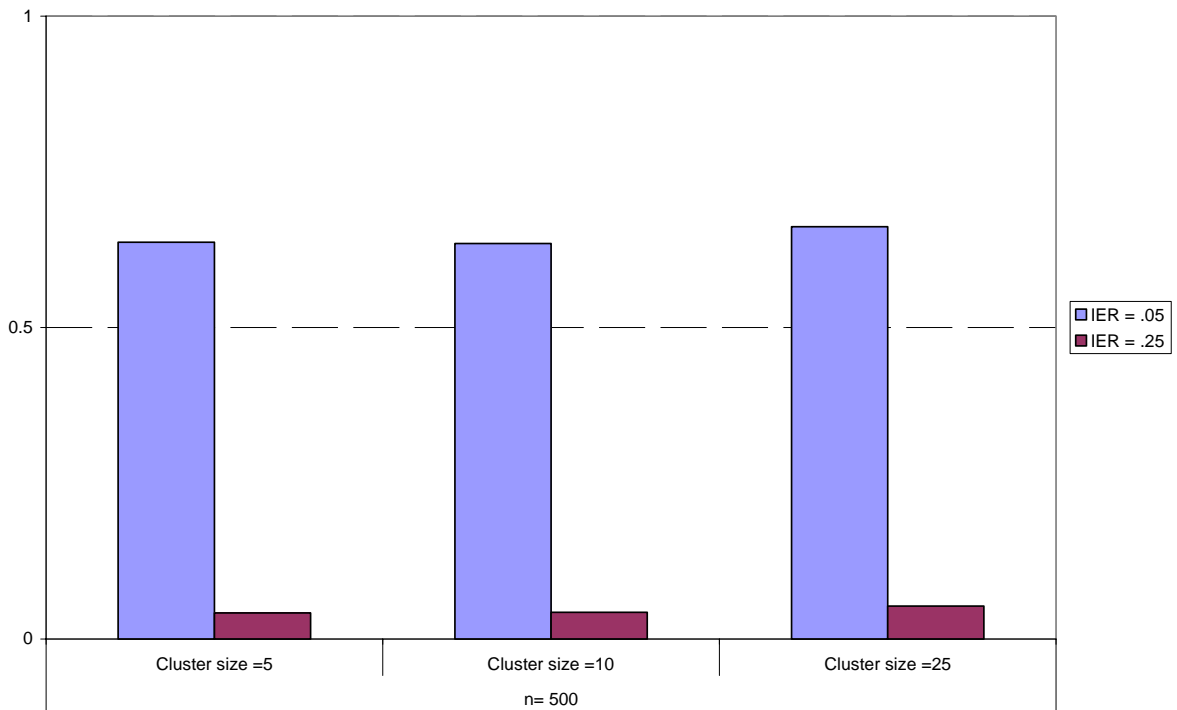
Figure 27. Accuracy for equally weighted groups with first LC pattern and sample size of 500 for RIC
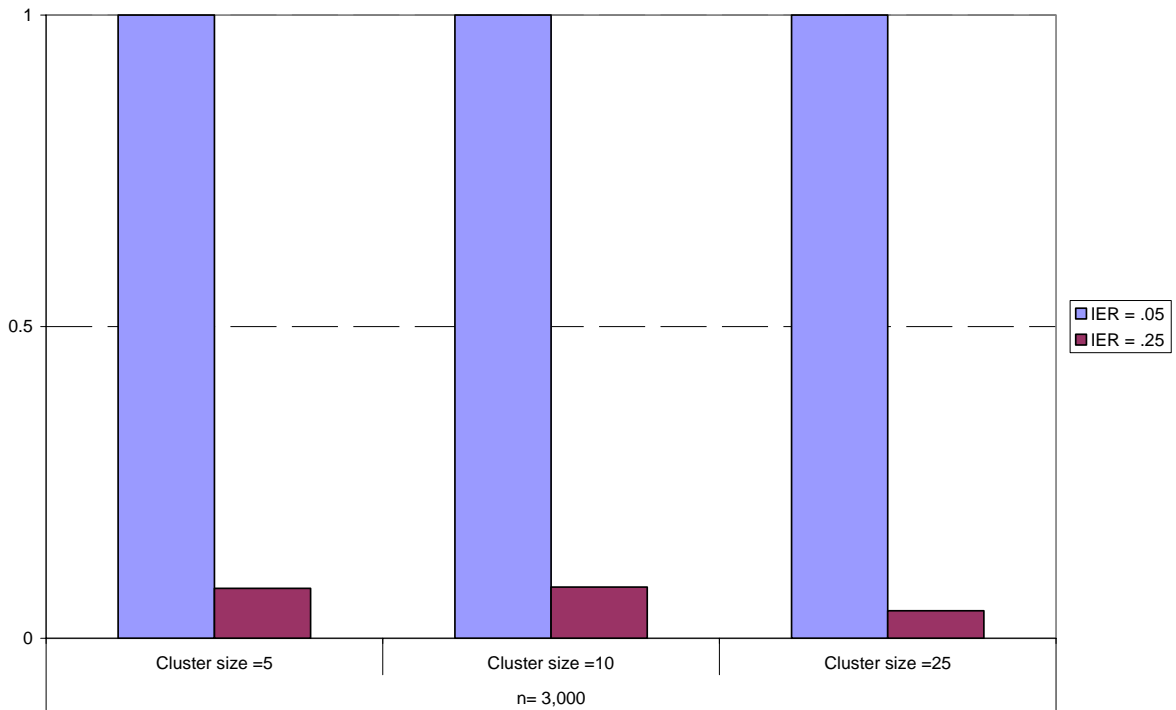
Figure 28. Accuracy for equally weighted groups with second LC pattern and sample size of 3,000 for BIC

In summary:

- As with the chi-square statistics, sample size was an important factor in the accuracy of the model comparison statistics. The model comparison statistics tended to be more accurate for large sample sizes.

- AIC performed well in more cases than BIC or RIC and RIC demonstrated high levels of accuracy in more cases than BIC.

- An important factor affecting the accuracy of AIC, BIC and RIC was the item-specific error rates. AIC was highly accurate in all cases where item-specific error rates were low and BIC and RIC showed high levels of accuracy in all cases where the item-specific error rates were low and sample sizes were large.

- As expected, model comparison statistics were more accurate for the third pattern of LC proportions. Sample size in combination with LC variability affected the accuracy of the model comparison statistics.

- Similar to the results of the chi-square statistics, cluster size (ICC) had virtually no effect on the accuracy of AIC, BIC and RIC.

Accuracy for Unequally Weighted Groups for AIC* BIC* and RIC*

As with the other model comparison statistics, a pseudo-likelihood was used to calculate AIC*, BIC* and RIC* for unequally weighted groups. When equally weighted groups were used, AIC was highly accurate in most cases where the sample sizes were large. However, when the groups were unequally weighted, Table 14 shows that AIC* was only accurate in instances when item-specific error rates were low that amounts to less than 80 percent of the cases where sample sizes were large when equally weighted groups were used. BIC* and RIC* also did not perform as well as when the groups were equally weighted. RIC* was somewhat accurate in less than two-thirds of the cases and BIC* was accurate in only half of the cases where sample sizes were large. When sample sizes were small, similar levels of accuracy were observed in only half of the cases for AIC*, one-quarter of the cases for RIC* and one-sixth of the cases for BIC*.

Table 14. Accuracy for unequally weighted groups for AIC*, BIC* and RIC*

Sample Size	Cluster Size	LC Pattern	Item Specific Error Rate	AIC*	BIC*	RIC*
500	5	1	.05	.261	.000	.008
			.25	.055	.000	.000
		2	.05	.702	.008	.210
			.25	.096	.000	.005
		3	.05	.949	.970	.999
			.25	.913	.098	.687
	10	1	.05	.240	.000	.013
			.25	.060	.000	.002
		2	.05	.691	.005	.201
			.25	.126	.000	.005
		3	.05	.945	.966	.997
			.25	.893	.078	.663
	25	1	.05	.275	.001	.023
			.25	.051	.000	.001
		2	.05	.695	.005	.247
			.25	.093	.000	.003
		3	.05	.933	.973	.998
			.25	.896	.088	.699
3000	5	1	.05	.971	.073	.563
			.25	.435	.000	.004
		2	.05	.983	.946	1.0
			.25	.760	.000	.039
		3	.05	.945	1.0	1.0
			.25	.894	1.0	.999
	10	1	.05	.972	.073	.568
			.25	.427	.000	.002
		2	.05	.977	.957	1.0
			.25	.734	.000	.032
		3	.05	.946	1.0	1.0
			.25	.899	1.0	.999
	25	1	.05	.972	.066	.543
			.25	.445	.000	.004
		2	.05	.985	.938	.999
			.25	.781	.000	.029
		3	.05	.955	1.0	1.0
			.25	.899	1.0	1.0

KEY

☐ < .500

▨ .501 to .750

▨ .751 to .900

▨ > .901

AIC*, BIC* and RIC* also did not perform as well with the unequally weighted

groups as with the equally weighted groups when item-specific error rates were high.

When the item-specific error rates were high, all three model comparison statistics were

not accurate in most cases despite other factors such as sample size and LC variability.

AIC* and RIC* continued to demonstrate high levels of accuracy in all cases where

sample sizes were large and item-specific error rates were low. However, BIC* did not

perform well when LC variability was low. When sample sizes were small, AIC* and

RIC* also did not detect the correct model in most cases when using the first pattern of

LC proportions.
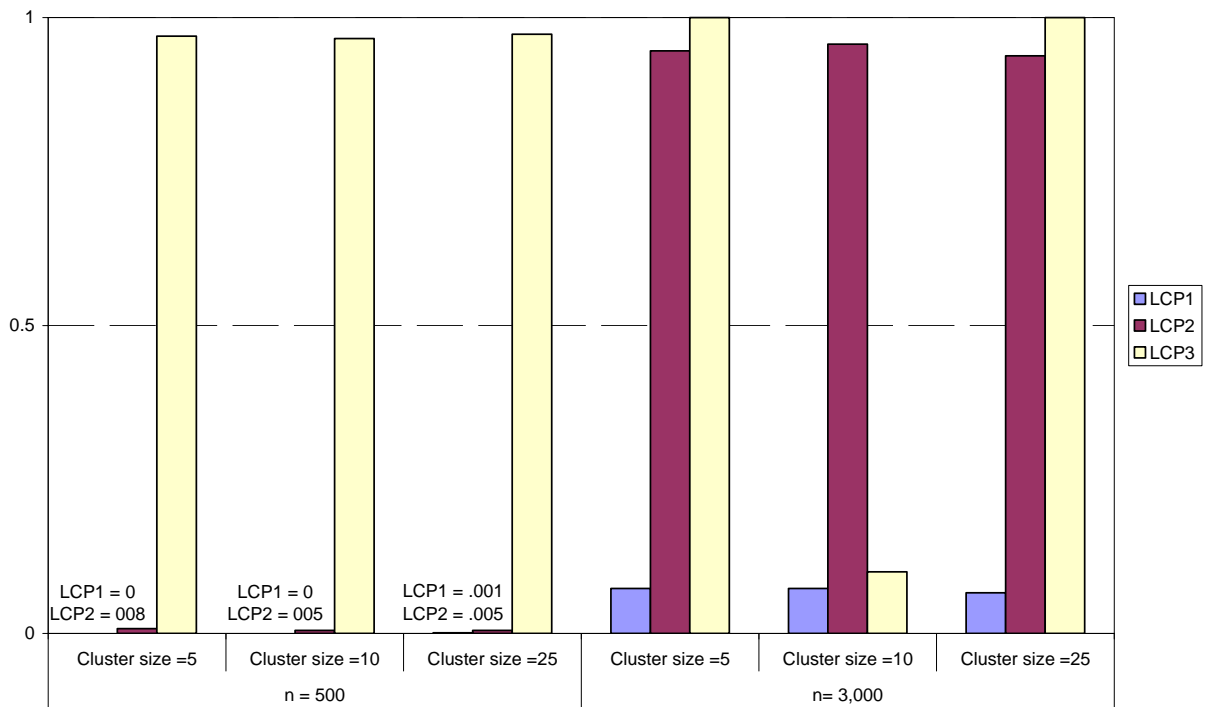
Figure 29. Accuracy for unequally weighted groups with item-specific error rate of .05 for BIC*
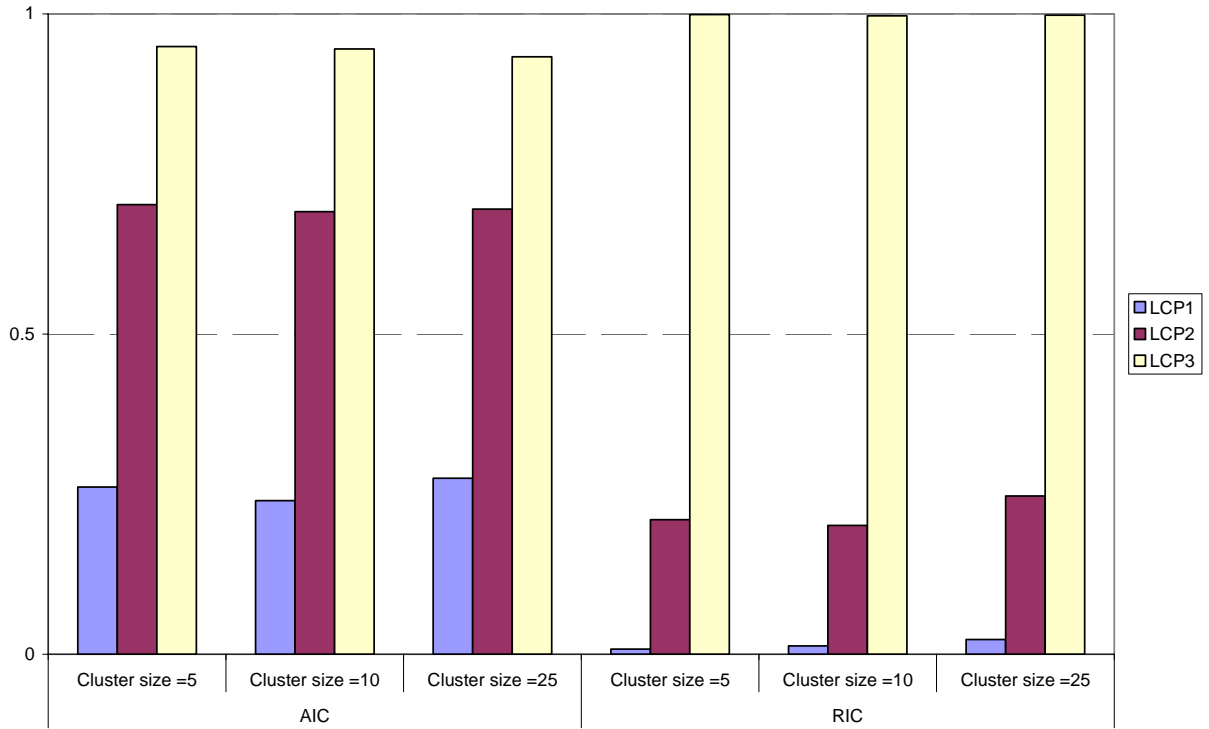
Figure 30. Accuracy for unequally weighted groups with item-specific error rate of .05 and sample size of 500 for AIC* and RIC*

Again, cluster size had little effect on the model comparison statistics' accuracy. The cases in which AIC*, BIC* and RIC* were accurate remained relatively similar across cluster sizes.
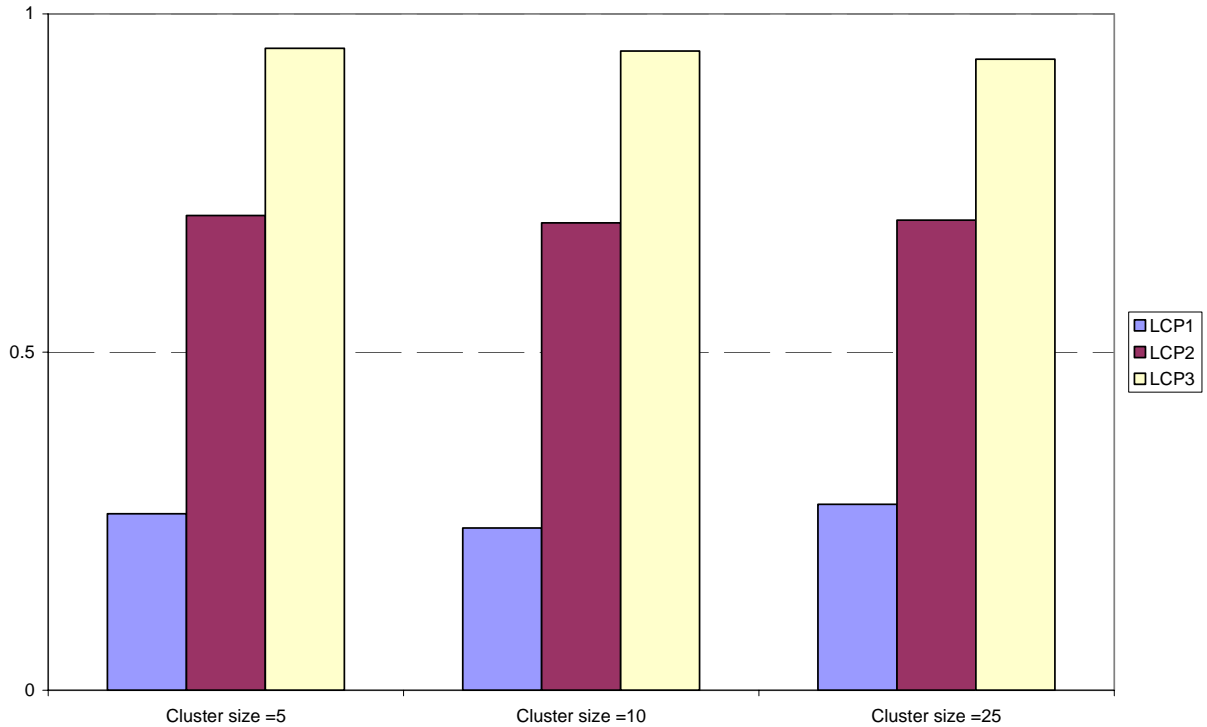
Figure 31. Accuracy for unequally weighted groups with item-specific error rate of .05 and sample size of 500 for AIC*

As with the equally weighted groups, the LC proportions had a notable effect on the performance of AIC*, BIC* and RIC*. The model comparison statistics again were more accurate when LC variability was high and sample sizes were large. In contrast, none of the model comparison statistics were highly accurate in any of the cases where LC variability was low and sample sizes were small. AIC* and RIC* improved in their accuracy when the sample sizes were large. However, BIC* remained inaccurate in all cases where LC variability was low.
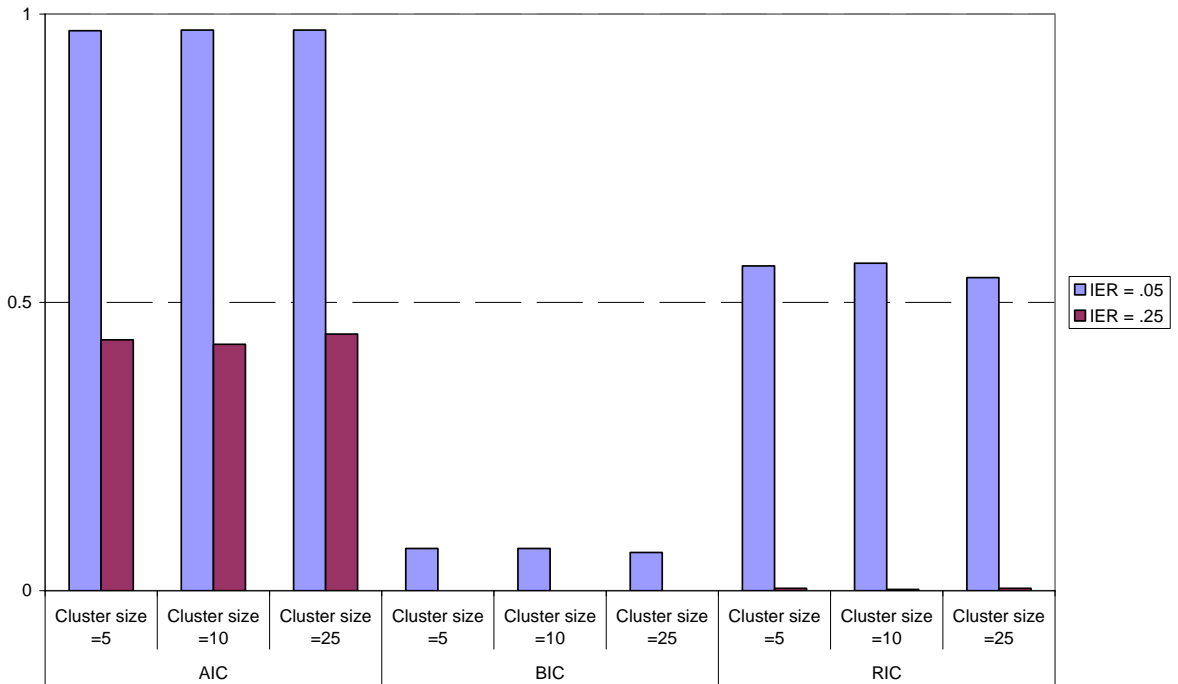
Figure 32. Accuracy for unequally weighted groups with first LC pattern and sample size of 3,000 for AIC*, BIC* and RIC*
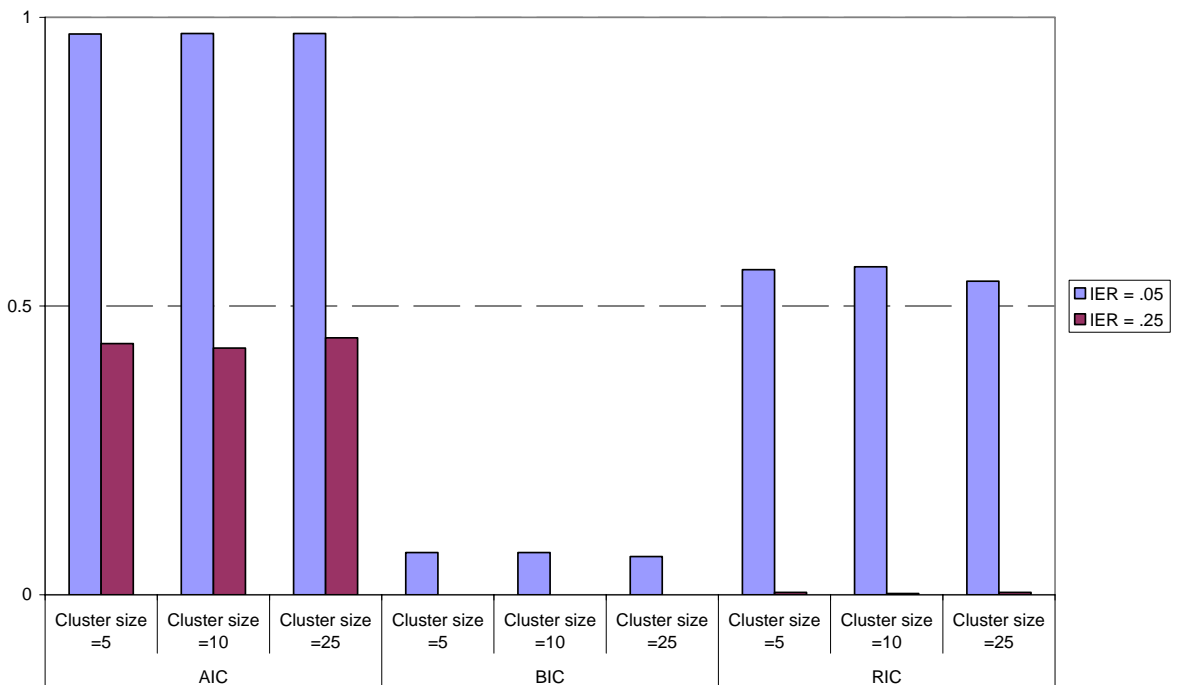
Figure 33. Accuracy for unequally weighted groups with third LC pattern and sample size of 500 for AIC*, BIC* and RIC*

When LC variability was medium (LCP = 2), AIC* was able to detect the correct model in cases where item-specific error rates were low and sample sizes were small. However, BIC* and RIC* remained inaccurate in all cases where sample sizes were small.

When LC variability was high as in Figure 34, all three model comparison statistics improved in accuracy. BIC* and RIC* were especially accurate when the sample sizes were large.
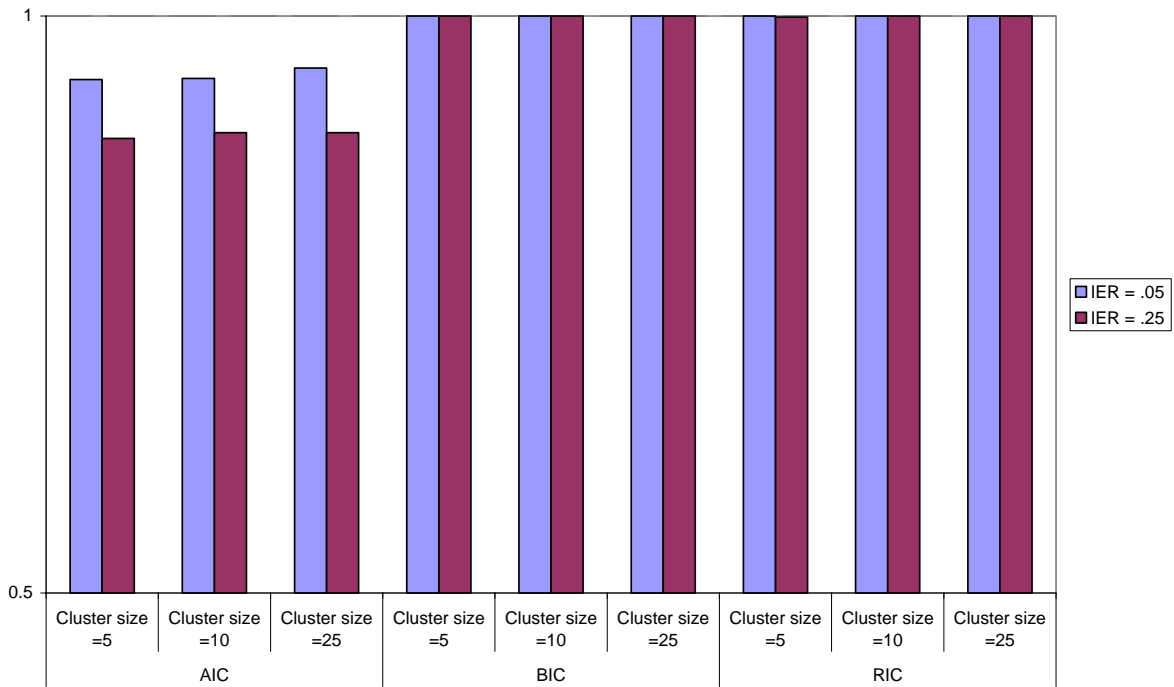
Figure 34. Accuracy for unequally weighted groups with third LC pattern and sample size of 3,000 for AIC*, BIC* and RIC*

In summary:

- AIC* and RIC* were accurate in all cases where sample sizes were large and item-specific error rates were low. BIC* also performed well in these cases except when the LC variability was low.

- In most cases, all three model comparison statistics were not accurate when item-specific error rates were high despite other factors such as sample size and LC variability.

- Improvement in accuracy was observed when LC variability was high for all three model comparison statistics. BIC* and RIC* were especially accurate in these cases when the sample sizes were large.

- As with the equally weighted groups, cluster size (ICC) had little effect on the accuracy of AIC*, BIC* and RIC*.

Overall Accuracy for AIC, BIC and RIC

The results of both equally and unequally weighted groups shown in Figure 35 suggest the following conclusions:

- AIC and BIC were more accurate when groups were equally weighted than when groups were unequally weighted. RIC retained the same level of accuracy for the use of equal or unequal sampling weights.

- As with the chi-square statistics, sample size affected the accuracy of AIC, BIC and RIC with both equally and unequally weighted groups. In all cases, the model comparison statistics were typically more accurate with large sample sizes.

- AIC was accurate in more cases than BIC and RIC and RIC was accurate in more cases than BIC.

- When item-specific error rates were high, AIC, BIC and RIC generally did not perform as well with the unequally weighted groups as with the equally

weighted groups. The item-specific error rates used had more of an influence on the model comparison statistics' accuracy with unequally weighted groups than with equally weighted groups.

- Cluster size (ICC) had little effect on the accuracy of AIC, BIC and RIC whether groups were equally or unequally weighted.
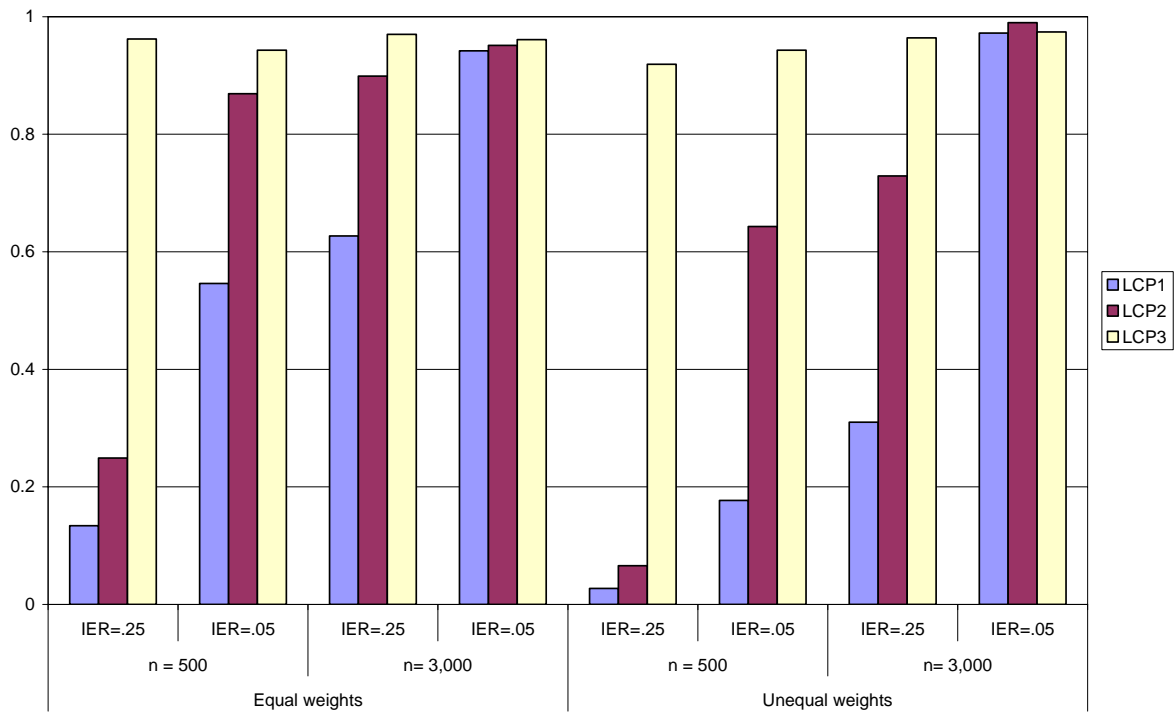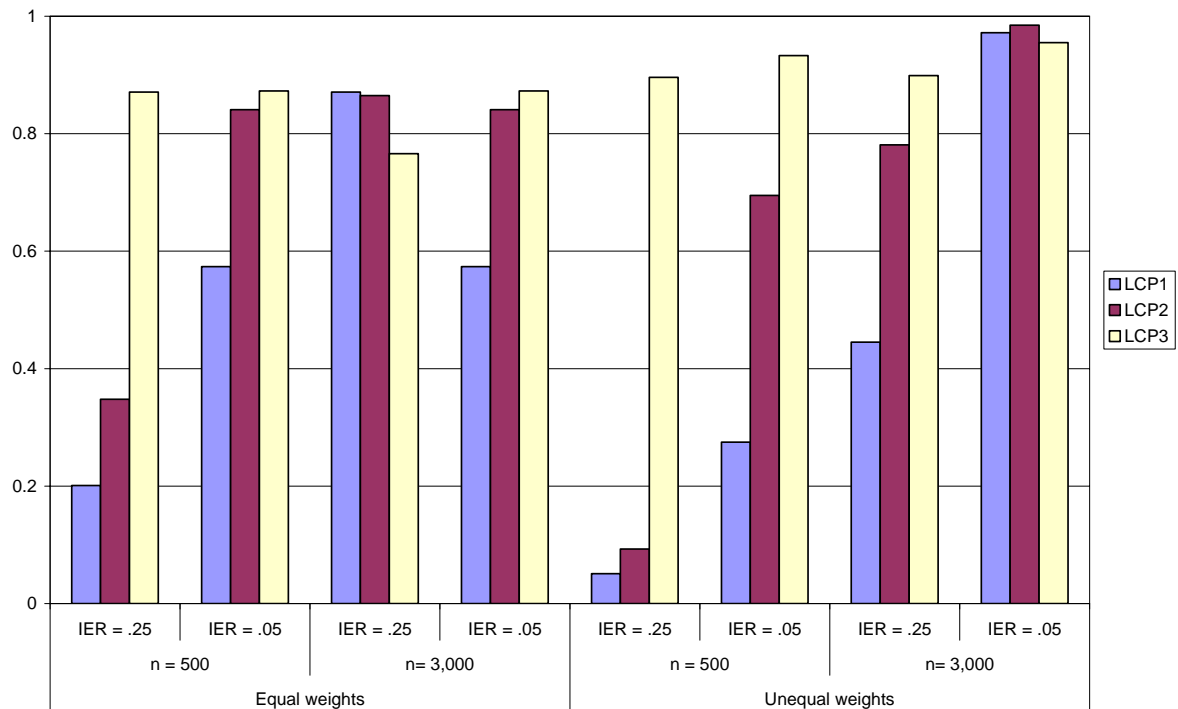

Figure 35. Accuracy for equally and unequally weighted groups with cluster size of 25 for AIC

Comparisons among Model Comparison Statistics

Figures 36 and 37 offer a comparison among X^2, G^2, I^2, AIC, BIC and RIC when groups are equally and unequally weighted. X^2, G^2, I^2 and AIC tended to perform similarly to one another and they were especially accurate when sample sizes were large. BIC and RIC did not perform as well as the other model comparison statistics under some of the same conditions. When item-specific error rates were high and LC variability was

low, BIC and RIC did not have as accurate model detection as the other model comparison statistics. BIC and RIC tended to perform at similar levels as the other model comparison statistics when LC variability was high, as in the third LC pattern.
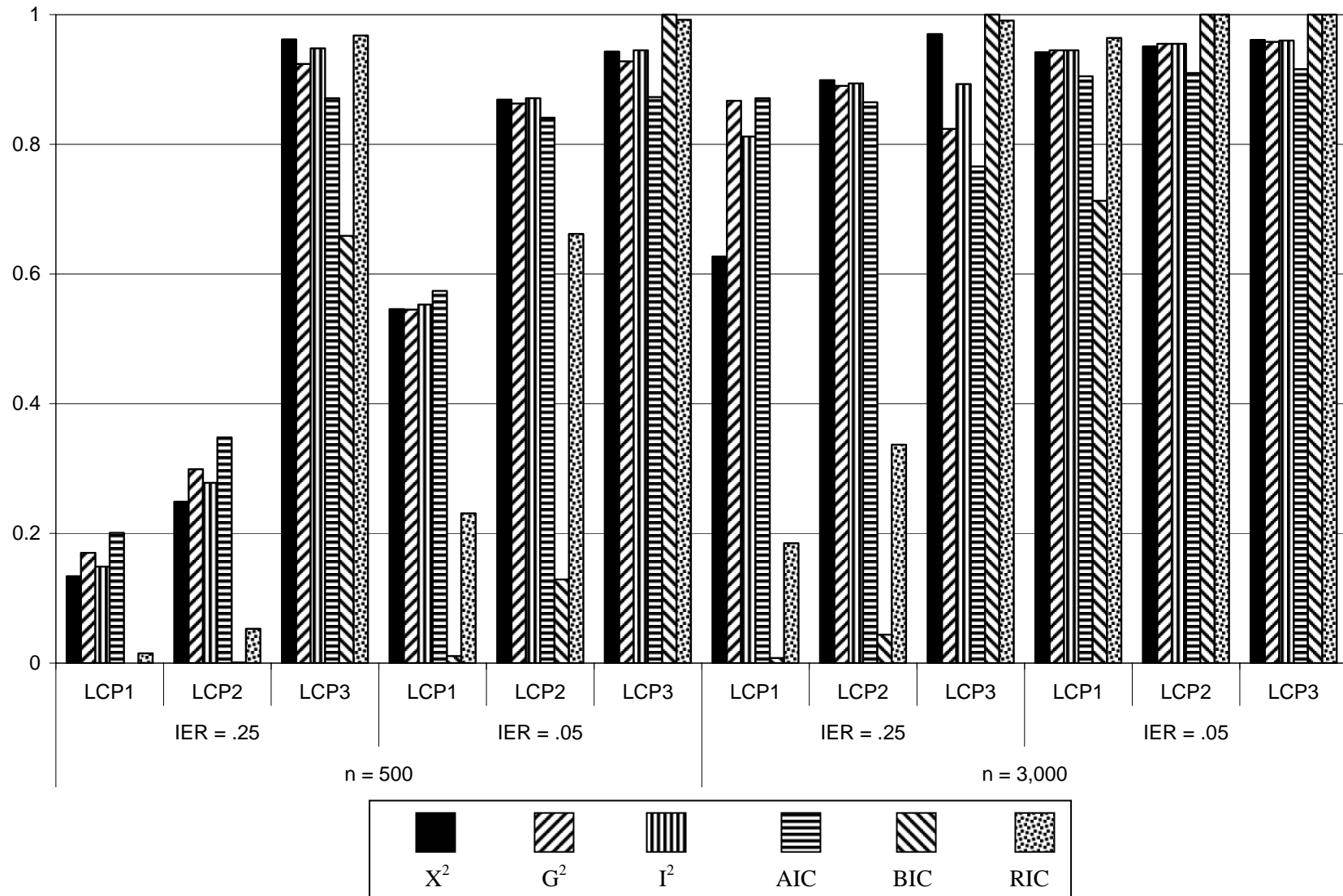
Figure 36. Accuracy for equally weighted groups with cluster size of 25 for X^2, G^2, I^2, AIC, BIC and RIC
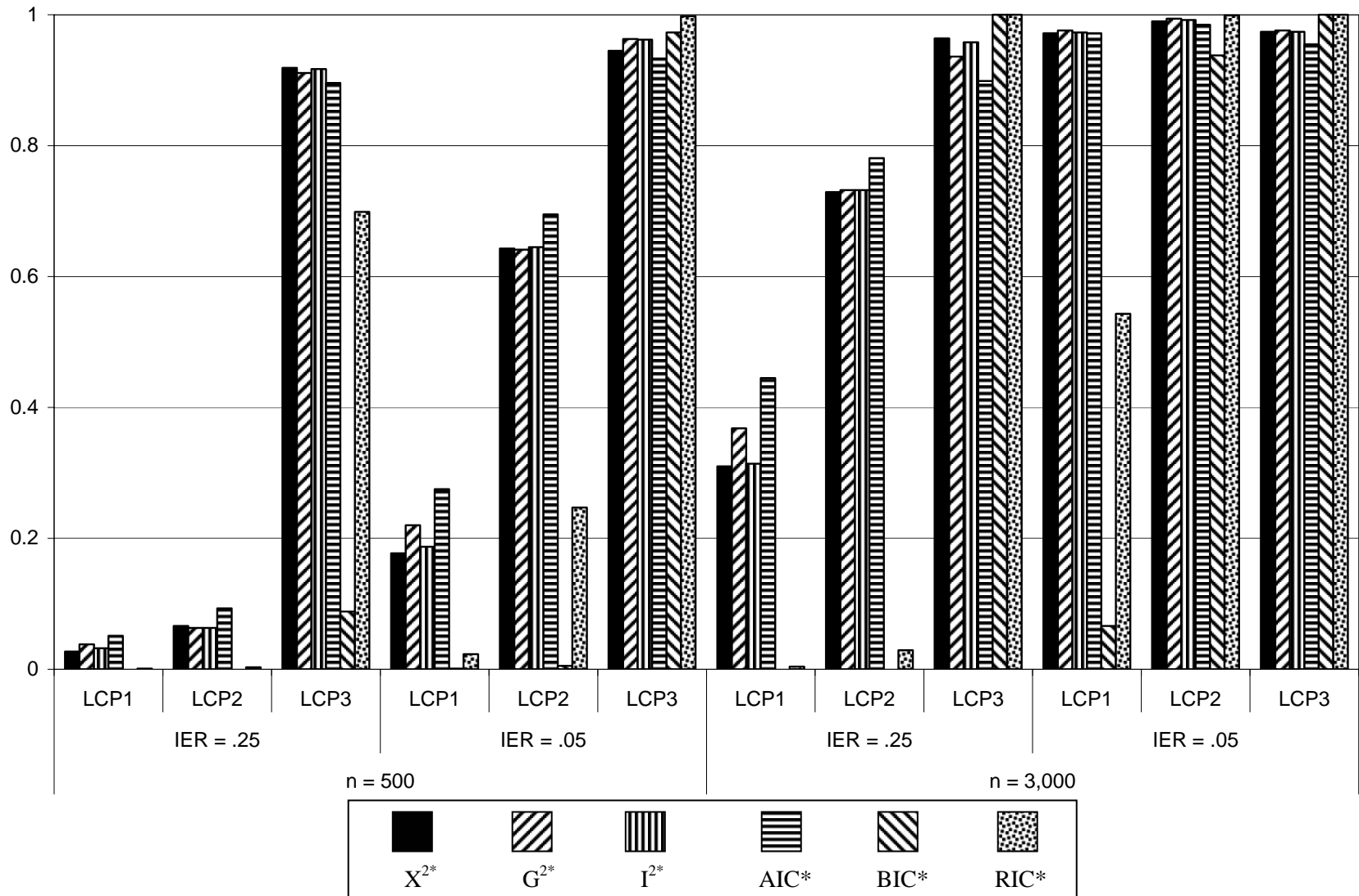
Figure 37. Accuracy for unequally weighted groups with cluster size of 25 for X^{2*}, G^{2*}, I^{2*}, AIC*, BIC* and RIC*

Figure 38 provides scatter plots of the results from X^2, G^2, I^2, AIC, BIC and RIC using equally weighted groups. The X^2, G^2 and I^2 results correlate well as demonstrated by the regression lines that capture most of the data points. These results were not surprising given the fact that these model comparison statistics performed equally well in the various simulated cases. The results of the scatter plot from the chi-square statistics and AIC again indicate that these model comparison statistics performed similarly under the various conditions of the cases. When comparing the results from AIC and BIC and from AIC and RIC, the results from these model comparison statistics do not correlate as well as the chi-square statistics and AIC. The lines drawn were unable to capture most of the data points, indicating a lack of correlation. Again, these results are not surprising given that AIC performed accurately in a number of cases where BIC and RIC did not.
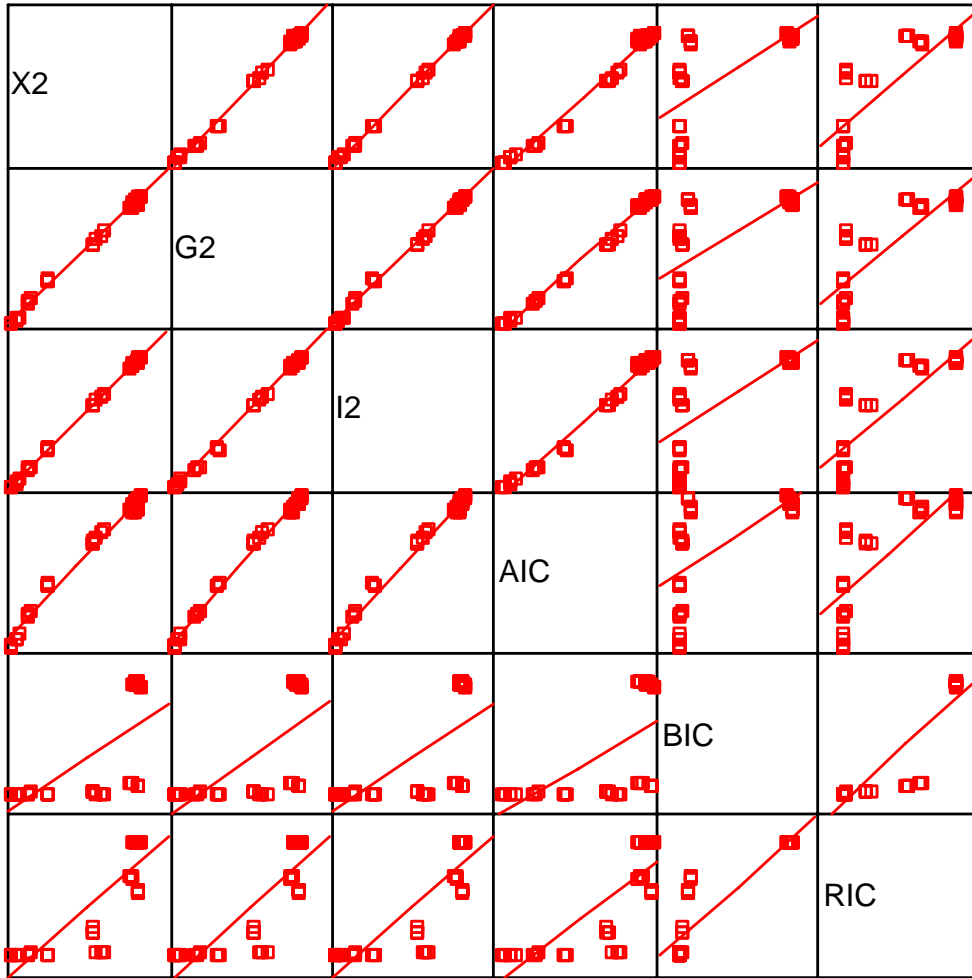
Figure 38: Accuracy for equally weighted groups for X^2, G^2, I^2, AIC, BIC and RIC

Figure 39 shows scatter plots of the results for X^{2*}, G^{2*}, I^{2*}, AIC*, BIC* and RIC* using unequally weighted groups. The results again indicate that the pseudo-chi-square model comparison statistics performed similarly under the various conditions of the cases. There is obvious correlation between these sets of model comparison statistics. Also, the pseudo-chi-square statistics and AIC* again show similar accuracy. In contrast, the results from the graphs of AIC* and BIC* and AIC* and RIC* imply less correlation between the sets of model comparison statistics.
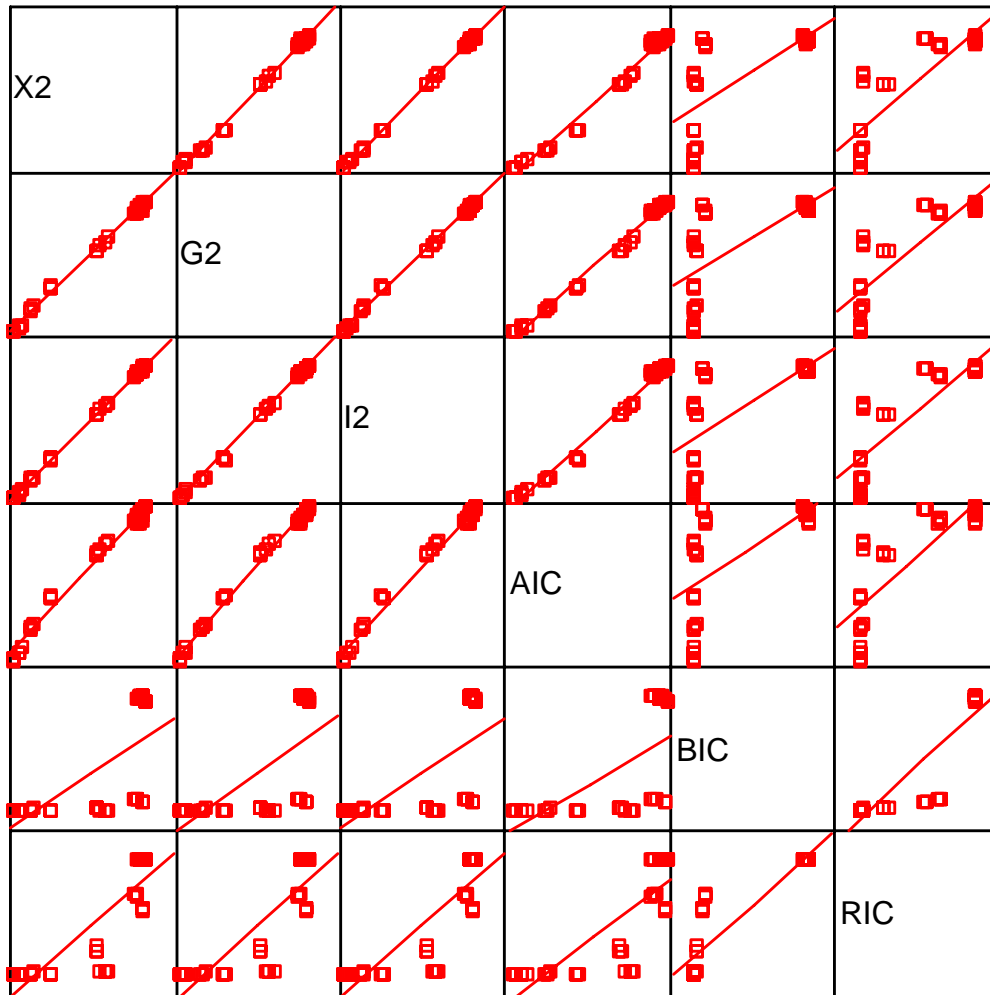
Figure 39: Accuracy for unequally weighted groups for X^{2*}, G^{2*}, I^{2*}, AIC*, BIC* and RIC*

Accuracy for Wald Statistic

The Wald statistic involves computing jackknifed variance estimates and consumes much longer computational time than other model comparison statistics. For this reason, the Wald statistic was only tested on a select number of cases. Cases were selected based on the results from the tests of the other model comparison statistics. Cases where the model comparison statistics were especially accurate or inaccurate were examined using the Wald statistic.

Table 15 shows the results of the Wald test. Cases where the sample size, item-specific error rates and LC proportions varied were examined. Cluster size was not considered because it appeared to have a negligible effect on the accuracy of the other model comparison statistics tested.

Table 15. Accuracy for Wald statistic for select cases

Sample Size	Item Specific Error Rates	Latent Class Proportions	Cluster Size	Weighting	Accuracy
3000	.05	1	25	Equal	1.0
3000	.05	2	25	Equal	1.0
3000	.05	3	25	Equal	.997
500	.05	3	25	Equal	.993
500	.25	1	25	Equal	.464
3000	.25	3	25	Equal	.387
500	.05	3	25	Unequal	.969
3000	.05	3	25	Unequal	1.0

Figure 40 offers a comparison of the results of the Wald statistic with the other model comparison statistics in the study. The Wald test performed similarly to X^2, G^2, I^2 and AIC. The Wald statistic demonstrated high levels of accuracy when sample sizes were large, item-specific error rates were low and LC variability was lower. These results were found when using both equally and unequally weighted groups. The only instances when the Wald statistic failed to select the correct model, were when item-specific error rates were high. Thus, high item-specific error rates may result in poor performance by the Wald test. However, other factors appear to have little effect on the accuracy of the Wald statistic.

Case	IER	LCP	Cluster	Wt	N	Wald	X^2	G^2	I^2	AIC	BIC	RIC
A	.05	1	25	Equal	3,000	1.000	0.942	0.945	0.945	0.905	0.713	0.964
B	.05	2	25	Equal	3,000	1.000	0.951	0.955	0.955	0.910	1.000	1.000
C	.05	3	25	Equal	3,000	0.997	0.961	0.958	0.960	0.916	1.000	1.000
D	.05	3	25	Equal	500	0.993	0.943	0.928	0.945	0.873	1.000	0.992
E	.25	1	25	Equal	500	0.464	0.134	0.170	0.149	0.201	0.000	0.015
F	.25	3	25	Equal	3,000	0.387	0.970	0.824	0.893	0.766	1.000	0.991
G	.05	3	25	Unequal	500	0.969	0.943	0.963	0.962	0.933	0.973	0.998
H	.05	3	25	Unequal	3,000	1.000	0.974	0.976	0.974	0.955	1.000	1.000
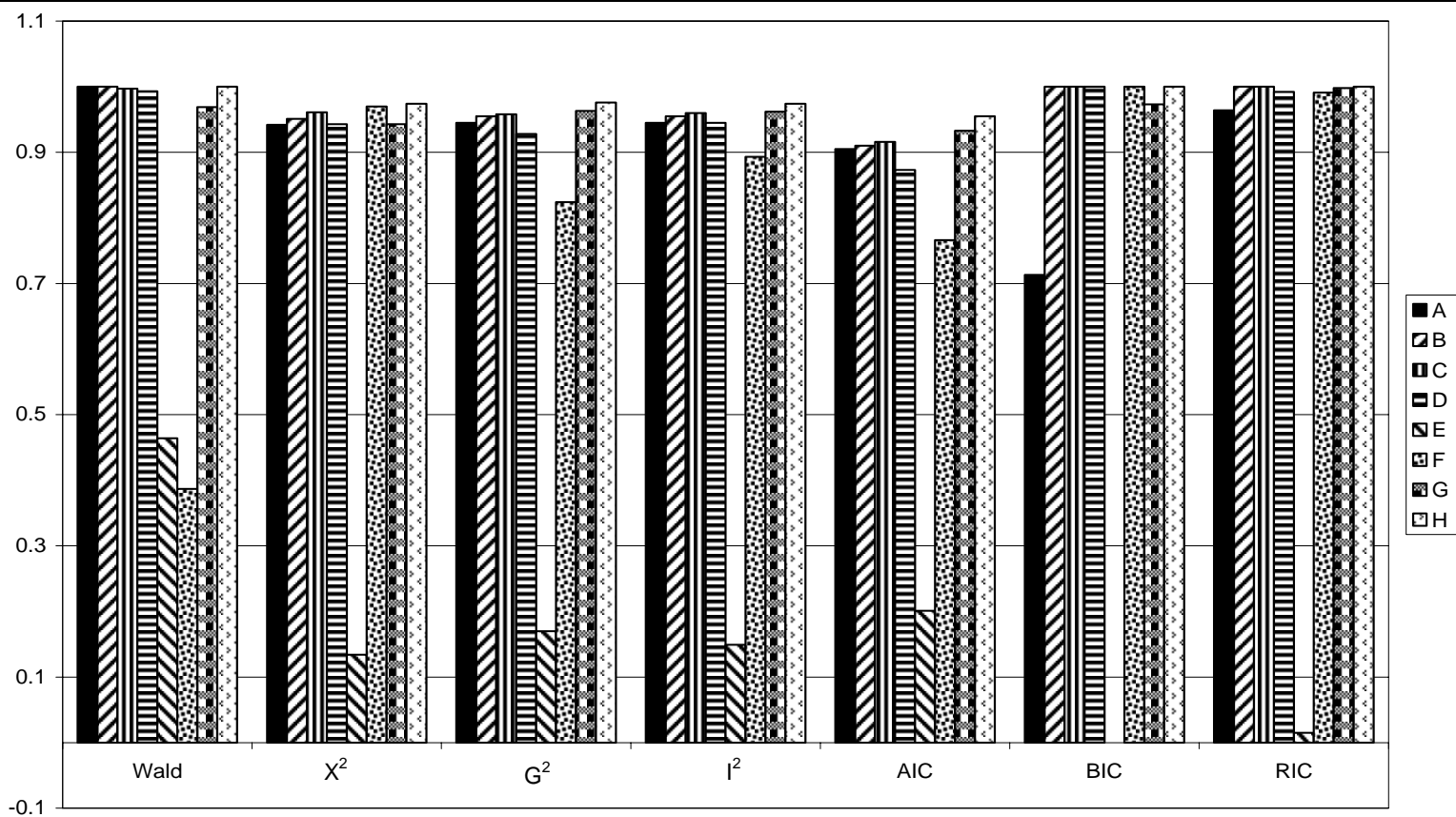
Figure 40. A comparison among X^2, G^2, I^2, AIC, BIC, RIC and Wald

Figure 41 provides scatter plots of the results from X^2, G^2, I^2, AIC, BIC, RIC and the Wald statistic using equally weighted groups. The relationships between the Wald statistic and the other statistics examined is somewhat obscured by the low number of cases in the scatter plot. However, the limited results of the Wald statistic and the chi-square statistics and AIC indicate that these model comparison statistics performed similarly in the cases examined. The results of the Wald statistic did not appear to be as correlated with the results of the BIC and RIC.
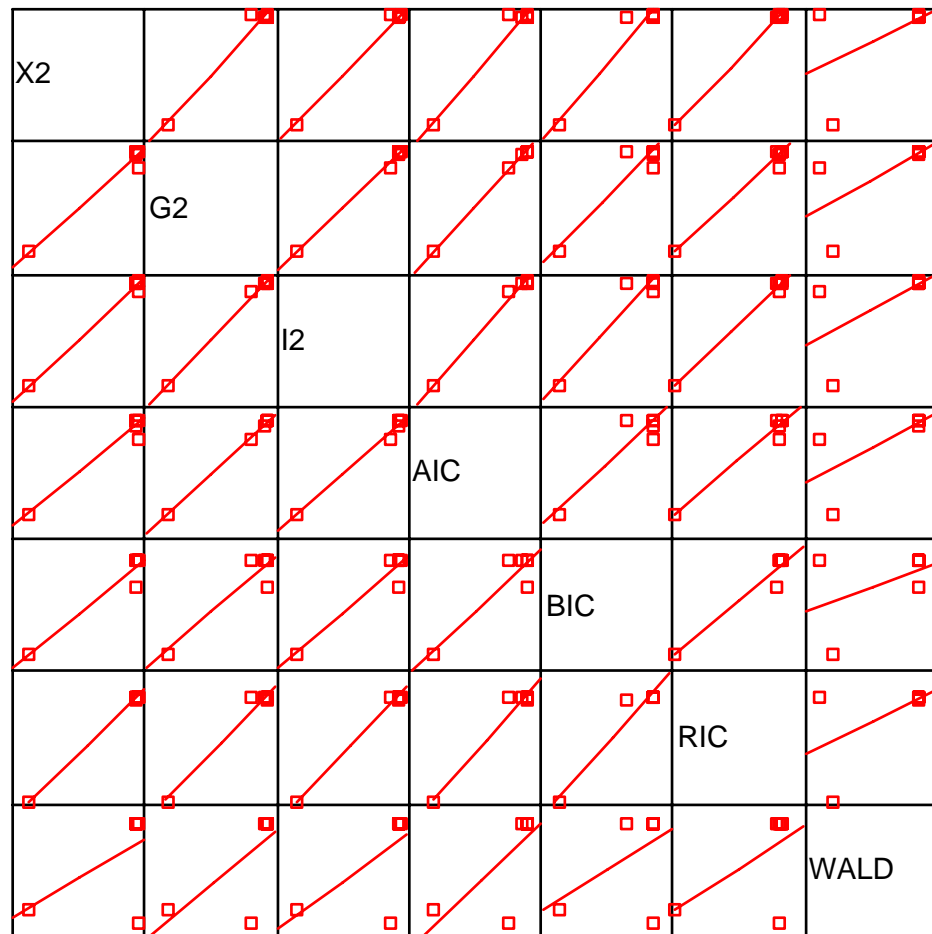
Figure 41: Accuracy for equally weighted groups for X^2, G^2, I^2, AIC, BIC, RIC and Wald

Figure 42 shows scatter plots of the results for X^{2*}, G^{2*}, I^{2*}, AIC*, BIC*, RIC* and the Wald statistic using unequally weighted groups. In this figure the results are limited to two data points because of the small number of cases examined with the Wald statistic. Again the results of the Wald statistic were correlated with the results of the pseudo-chi-square statistics and the AIC*. The results of the BIC* were also similar to those of the Wald statistic. The results of the RIC*, however, did not appear to be as correlated with the results of the Wald statistic.
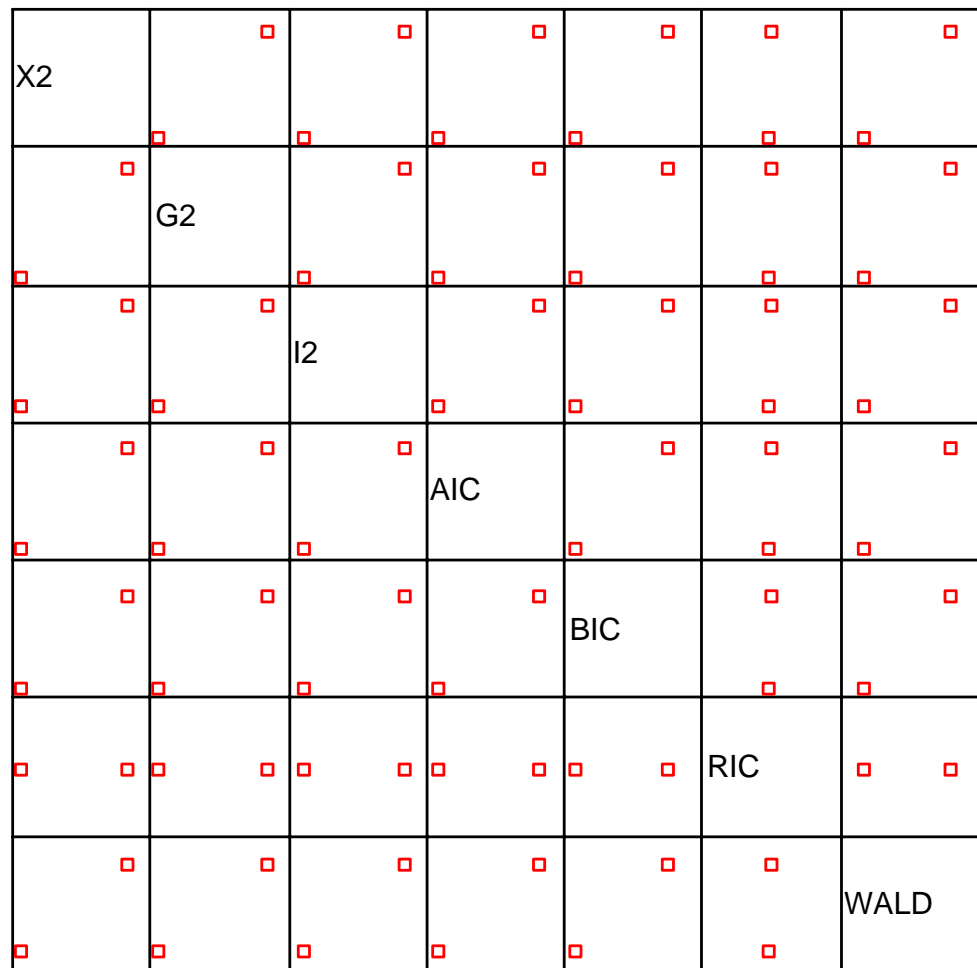
Figure 42: Accuracy for unequally weighted groups for X^{2*}, G^{2*}, I^{2*}, AIC*, BIC* RIC* and Wald

<u>NHSDA Simulation Results</u>

A simulation was performed based on the 1979 and 1988 NHSDA data. The first pattern of LC proportions (LCP1) were used as found in the analysis of the NHSDA data. In addition, since the two years of data were similar in size and sampled from the same size population, 100 clusters of size 80 were used in both groups and equal sampling weights were applied. The results of the NHSDA simulation at the .05 alpha level are reported in Table 16. The proportion of times out of 1000 where the model comparison statistic selected the correct model is shown.

Table 16. Accuracy for NHSDA Simulation

X^2	G^2	I^2	AIC	BIC	RIC	Wald
.941	.945	.941	.907	1.0	1.0	.987

As the table demonstrates, all of the model comparison statistics were highly accurate. The Wald statistic performed well, selecting the correct model in a large majority of the replications. These results are predictable based on the findings from the simulations reported above. In the NHSDA simulation, the sample sizes were large, the item-specific error rates were low and the groups were equally weighted.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

In this study, latent class scaling was used to study the research question: *Do patterns of drug use show change over time?* This question was explored using traditional LCA methods in LEM on the NHSDA data and the results indicated that indeed reported drug use patterns changed somewhat between 1979 and 1988. However, these findings can be considered questionable due to the use of complex survey sampling methods in collecting the NHSDA data. To address this issue, this study performed two analyses. First, how select model comparison statistics function when performing simultaneous latent structure analyses on complex sample survey data was examined. The characteristics of the samples were varied to understand how the model comparison statistics functioned under various conditions, including sample size, error level, ICC, LC variability and use of unequal sample weights. Second, data were simulated to be comparable to the 1979 and 1988 NHSDA and the accuracy of various model comparison statistics in detecting the correct model was tested.

Chi-Square Statistics

The three chi-square statistics (the Pearson statistic, X^2, the likelihood-ratio statistic, G^2 and the Read-Cressie statistic, I^2) were tested for their abilities to determine the best fitting model. Results showed that the chi-square statistics performed well under certain conditions. A large sample size was found to be essential to obtaining accurate

results with these model comparison statistics. The chi-square statistics also performed well with all of the ICCs in the simulations.

When samples included other types of characteristics, the chi-square statistics were not necessarily accurate at identifying the best fitting models. High item-specific error rates were the greatest barrier to identifying the correct model. When sample sizes were large, item-specific error rates tended to have less of an effect on the model comparison statistics' accuracy. However, for small sample sizes and unequally weighted groups, a high item-specific error rate often resulted in incorrect model identification. Further, when LC variability was low and, therefore, more difficult to detect, the chi-square statistics were not necessarily accurate.

Akaike Information Criterion (AIC) Statistic

AIC was also evaluated for its ability to determine the best fitting model. Results showed that AIC performed well under most conditions, especially when groups were equally weighted. As with the chi-square statistics, however, item-specific error rates appear to be an important determinant of AIC's accuracy. When item-specific error rates were high, AIC did not necessarily perform well. This was especially the case for small sample sizes and unequally weighted groups. When groups were unequally weighted, AIC had difficulty identifying the correct model when LC variability was low and item-specific error rates were high.

Rissanen Statistic (RIC)

RIC did not perform as well as AIC. It was an accurate model comparison statistic when item-specific error rates were low. Also, sample size did not have as much of an influence over its accuracy as some of the other model comparison statistics examined when groups were equally weighted. However, for large sample sizes, RIC was able to detect the more subtle variability between latent class structures that it could not detect for small sample sizes.

When groups were unequally weighted, sample size became an important determinant of RIC's accuracy. RIC performed well in twice as many cases with large sample sizes than with small sample sizes. Also, RIC was moderately accurate at detecting the correct model with low LC variability as long as sample sizes were large and item-specific error rates were low.

Item-specific error rates had the most obvious influence over the performance of RIC. In most cases, when item-specific error rates were high, RIC was unable to detect the correct model. The only exception to this rule, was when LC variability was high. Then, RIC was sometimes able to detect the correct model despite high item-specific error rates.

Bayesian Information Criterion (BIC) Statistic

Although conceptually similar to them, BIC did not perform as well as AIC and RIC. In fact, BIC was a poor performer in many of the simulations. For small sample sizes, BIC was not accurate with low LC variability. In fact, even for large sample sizes,

BIC was unable to detect the correct model with low LC variability when item-specific error rates were high.

When groups were unequally weighted, BIC was even less accurate. The model comparison statistic was unable to detect the correct model in a majority of simulations, especially for small sample sizes. When sample sizes were large, BIC was generally only able to detect the correct model in cases where LC variability was high.

Wald Statistic

Although fewer simulations were performed using the Wald statistic, some important conclusions can be drawn about its performance. The Wald statistic performed well in a majority of the simulations tested. The Wald statistic was very accurate despite low LC variability, small sample sizes, or unequal sampling weights. The only instance, which resulted in poor performance by the Wald statistic, was when high item-specific error rates were present in the sample. This result is consistent with the performance of the other model comparison statistics, which tended to perform poorly with high item-specific error rates.

Conclusions on the Use of Unequal Weighting and Clustering

Model comparison statistics routinely used for making inferences about populations may not produce accurate estimates when applied to data from complex survey designs. Typical model comparison statistics assume SRS. When this assumption is not met, the model comparison statistics may not produce accurate results. To address the problem of calculating valid model comparison statistics, this study investigated the

behavior of various model comparison statistics when performing simultaneous latent structure analyses on complex sample survey data by examining the effects of unequal sampling and clustering in the dataset.

Our results offer support for ignoring clustering when applying these model comparison statistics. This study explored typical ICC levels, such as .05 and .10 and atypically high ICC levels, such as .20. In all cases, ICC level did not appear to have an effect on the accuracy of the model comparison statistics in detecting the correct model. However, due to our use of Dirichlet values, we were unable to vary the ICC level while holding the cluster size constant. Therefore, this issue needs further scrutiny before drawing firm conclusions on the effects of ICC level.

When sampling across groups was largely unequal, the model comparison statistics did not perform as well as when sampling was the same across groups. Therefore, we would advise caution when applying these model comparison statistics to data with unequal weighting across groups. However, in this study the weighting was strongly skewed towards one group, so the model comparison statistics may be more accurate when the weighting ratio is less severe.

Analysis using the NHSDA Data

The second part of the study that involved developing a simulation based on the NHSDA data, showed that the chi-square statistics and AIC, BIC and RIC were accurate for use in detecting the correct model. In the preliminary analysis of the NHSDA data AIC, BIC and RIC selected the partial-homogeneous model as the best fitting model to the data.

As the LEM analysis showed, each of the latent classes increased slightly between 1979 and 1988, apart from the fourth latent class, which decreased more substantially. The fourth class is characterized by individuals who have used alcohol and tobacco, but have refrained from marijuana and harder "street drugs."

Based on the results of the simulation, it is possible to conclude that the preliminary analysis conducted with LEM is accurate and therefore, reported drug use patterns did change between 1979 and 1988. Most of the reported drug use classes increased slightly, while the group characterized by reported alcohol and tobacco use alone decreased more substantially.

Limitations of the Study

There were several limitations of the present research. This study determined the accuracy of a small number of model comparison statistics for a two-group comparison. Results may differ for the model comparison statistics examined when comparing more than two groups. Also, a number of other model comparison statistics could be examined in addition to those reviewed in this study. Additionally, this research only examined three common models used in simultaneous latent structure analysis, heterogeneous, partial-homogeneous and homogeneous. However, more complicated models were not examined and should be examined in future research. It is possible that the model comparison statistics examined may result in different levels of accuracy when tested with more complicated models.

The model comparison statistics examined were not accurate in all instances that were tested. For this reason, some guidelines are offered for using these model comparison statistics in LCA:

- All of the model comparison statistics performed best with large sample sizes (3000 observations per group). In several cases, many of the other factors affecting the model comparison statistics, including LC variability, item-specific error rates and weighting, were minimized for large sample sizes. For this reason, a large number of observations is preferred when performing any of these model comparison statistics.

- High item-specific error rates resulted in poor identification for most of the model comparison statistics. Therefore, caution is recommended when using these model comparison statistics when high item-specific error rates are suspected.

- ICC level appeared to have little effect on the model comparison statistics. For this reason, these model comparison statistics most likely can be used with high or low ICC levels.

- BIC was inaccurate at model identification in most of the simulations performed. So this model comparison statistic should be used with caution in simultaneous latent structure analysis.

Implications for Further Research

1. Future research should focus on testing the Wald statistic and its accuracy depending on various factors. Because the Wald test is time-consuming to calculate, more study should be dedicated to its performance under various conditions.

2. More investigation should be given to the issue of ICC levels and how they affects model comparison statistics. It may be possible to vary the ICC values while holding cluster size constant in a two class model because the conversion to Dirichlet values is unnecessary. Therefore, a future study may want to focus on the effects of ICC level in a two-class, simultaneous latent structure analysis.

3. It may be of interest to replicate data that mirrors past simultaneous latent structure analysis research to determine if conclusions are accurate. This would be especially useful in the field of drug use research where a number of studies have employed this and similar methods for studying reported drug use over time.

APPENDIX A: RESULTS TABLES

KEY TO TABLES

☐	< .500
▫	.501 to .750
▪	.751 to .900
■	> .901

Table A-1: Accuracy for equally weighted groups by sample size

Sample Size	Cluster Size	LC Pattern	Item Error Rate	X^2 α=.05	G^2 α=.05	I^2 α=.05	X^2 α=.01	G^2 α=.01	I^2 α=.01
500	5	1	.05	.532	.530	.541	.300	.300	.296
			.25	.131	.162	.146	.039	.047	.043
		2	.05	.864	.849	.862	.763	.770	.770
			.25	.231	.261	.246	.084	.103	.093
		3	.05	.953	.945	.952	.989	.984	.992
			.25	.965	.913	.949	.989	.979	.988
	10	1	.05	.529	.530	.533	.317	.334	.324
			.25	.142	.164	.145	.049	.063	.056
		2	.05	.877	.872	.881	.801	.798	.803
			.25	.243	.270	.257	.093	.116	.102
		3	.05	.953	.944	.958	.992	.989	.992
			.25	.965	.922	.943	.986	.970	.979
	25	1	.05	.546	.545	.553	.363	.360	.360
			.25	.134	.170	.149	.037	.050	.043
		2	.05	.869	.863	.871	.792	.789	.795
			.25	.249	.299	.278	.099	.119	.107
		3	.05	.943	.928	.945	.987	.987	.989
			.25	.962	.924	.948	.983	.967	.980
3000	5	1	.05	.944	.938	.943	.983	.985	.985
			.25	.636	.840	.791	.433	.723	.604
		2	.05	.968	.968	.970	.993	.995	.994
			.25	.930	.939	.942	.860	.922	.905
		3	.05	.947	.948	.945	.986	.983	.986
			.25	.962	.841	.898	.992	.921	.965
	10	1	.05	.958	.957	.958	.991	.991	.991
			.25	.604	.858	.811	.382	.720	.594
		2	.05	.948	.947	.950	.990	.992	.990
			.25	.917	.921	.927	.838	.904	.890
		3	.05	.953	.954	.953	.989	.989	.988
			.25	.964	.825	.907	.994	.917	.964
	25	1	.05	.942	.945	.945	.987	.986	.988
			.25	.627	.867	.812	.445	.738	.624
		2	.05	.951	.955	.955	.992	.992	.993
			.25	.899	.890	.894	.832	.823	.828
		3	.05	.961	.958	.960	.993	.995	.994
			.25	.970	.824	.893	.996	.918	.971

Table A-2: Accuracy for equally weighted groups with item error rate of .05

Sample Size	Cluster Size	LC Pattern	X^2 $\alpha=.05$	G^2 $\alpha=.05$	I^2 $\alpha=.05$	X^2 $\alpha=.01$	G^2 $\alpha=.01$	I^2 $\alpha=.01$
500	5	1	.532	.530	.541	.300	.300	.296
		2	.864	.849	.862	.763	.770	.770
		3	.953	.945	.952	.989	.984	.992
	10	1	.529	.530	.533	.317	.334	.324
		2	.877	.872	.881	.801	.798	.803
		3	.953	.944	.958	.992	.989	.992
	25	1	.546	.545	.553	.363	.360	.360
		2	.869	.863	.871	.792	.789	.795
		3	.943	.928	.945	.987	.987	.989
3000	5	1	.944	.938	.943	.983	.985	.985
		2	.968	.968	.970	.993	.995	.994
		3	.947	.948	.945	.986	.983	.986
	10	1	.958	.957	.958	.991	.991	.991
		2	.948	.947	.950	.990	.992	.990
		3	.953	.954	.953	.989	.989	.988
	25	1	.942	.945	.945	.987	.986	.988
		2	.951	.955	.955	.992	.992	.993
		3	.961	.958	.960	.993	.995	.994

Table A-3: Accuracy for equally weighted groups with item error rate of .25

Sample Size	Cluster Size	LC Pattern	X^2 $\alpha=.05$	G^2 $\alpha=.05$	I^2 $\alpha=.05$	X^2 $\alpha=.01$	G^2 $\alpha=.01$	I^2 $\alpha=.01$
500	5	1	.131	.162	.146	.039	.047	.043
		2	.231	.261	.246	.084	.103	.093
		3	.965	.913	.949	.989	.979	.988
	10	1	.142	.164	.145	.049	.063	.056
		2	.243	.270	.257	.093	.116	.102
		3	.965	.922	.943	.986	.970	.979
	25	1	.134	.170	.149	.037	.050	.043
		2	.249	.299	.278	.099	.119	.107
		3	.962	.924	.948	.983	.967	.980
3000	5	1	.636	.840	.791	.433	.723	.604
		2	.930	.939	.942	.860	.922	.905
		3	.962	.841	.898	.992	.921	.965
	10	1	.604	.858	.811	.382	.720	.594
		2	.917	.921	.927	.838	.904	.890
		3	.964	.825	.907	.994	.917	.964
	25	1	.627	.867	.812	.445	.738	.624
		2	.899	.890	.894	.832	.823	.828
		3	.970	.824	.893	.996	.918	.971

Table A-4: Accuracy for equally weighted groups with cluster size of 5

Item Error Rate	Sample Size	LC Pattern	X^2 α=.05	G^2 α=.05	I^2 α=.05	X^2 α=.01	G^2 α=.01	I^2 α=.01
.05	500	1	.532	.530	.541	.300	.300	.296
		2	.864	.849	.862	.763	.770	.770
		3	.953	.945	.952	.989	.984	.992
	3000	1	.944	.938	.943	.983	.985	.985
		2	.968	.968	.970	.993	.995	.994
		3	.947	.948	.945	.986	.983	.986
.25	500	1	.131	.162	.146	.039	.047	.043
		2	.231	.261	.246	.084	.103	.093
		3	.965	.913	.949	.989	.979	.988
	3000	1	.636	.840	.791	.433	.723	.604
		2	.930	.939	.942	.860	.922	.905
		3	.962	.841	.898	.992	.921	.965

Table A-5: Accuracy for equally weighted groups with cluster size of 10

Item Error Rate	Sample Size	LC Pattern	X^2 α=.05	G^2 α=.05	I^2 α=.05	X^2 α=.01	G^2 α=.01	I^2 α=.01
.05	500	1	.529	.530	.533	.317	.334	.324
		2	.877	.872	.881	.801	.798	.803
		3	.953	.944	.958	.992	.989	.992
	3000	1	.958	.957	.958	.991	.991	.991
		2	.948	.947	.950	.990	.992	.990
		3	.953	.954	.953	.989	.989	.988
.25	500	1	.142	.164	.145	.049	.063	.056
		2	.243	.270	.257	.093	.116	.102
		3	.965	.922	.943	.986	.970	.979
	3000	1	.604	.858	.811	.382	.720	.594
		2	.917	.921	.927	.838	.904	.890
		3	.964	.825	.907	.994	.917	.964

Table A-6: Accuracy for equally weighted groups with cluster size of 25

Item Error Rate	Sample Size	LC Pattern	X^2 α=.05	G^2 α=.05	I^2 α=.05	X^2 α=.01	G^2 α=.01	I^2 α=.01
.05	500	1	.546	.545	.553	.363	.360	.360
		2	.869	.863	.871	.792	.789	.795
		3	.943	.928	.945	.987	.987	.989
	3000	1	.942	.945	.945	.987	.986	.988
		2	.951	.955	.955	.992	.992	.993
		3	.961	.958	.960	.993	.995	.994
.25	500	1	.134	.170	.149	.037	.050	.043
		2	.249	.299	.278	.099	.119	.107
		3	.962	.924	.948	.983	.967	.980
	3000	1	.627	.867	.812	.445	.738	.624
		2	.899	.890	.894	.832	.823	.828
		3	.970	.824	.893	.996	.918	.971

Table A-7: Accuracy for equally weighted groups with LC pattern of 1 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	X^2 α=.05	G^2 α=.05	I^2 α=.05	X^2 α=.01	G^2 A=.01	I^2 A=.01
500	5	.05	.532	.530	.541	.300	.300	.296
		.25	.131	.162	.146	.039	.047	.043
	10	.05	.529	.530	.533	.317	.334	.324
		.25	.142	.164	.145	.049	.063	.056
	25	.05	.546	.545	.553	.363	.360	.360
		.25	.134	.170	.149	.037	.050	.043

Table A-8: Accuracy for equally weighted groups with LC pattern of 2 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	X^2 α=.05	G^2 α=.05	I^2 α=.05	X^2 α=.01	G^2 α=.01	I^2 α=.01
500	5	.05	.864	.849	.862	.763	.770	.770
		.25	.231	.261	.246	.084	.103	.093
	10	.05	.877	.872	.881	.801	.798	.803
		.25	.243	.270	.257	.093	.116	.102
	25	.05	.869	.863	.871	.792	.789	.795
		.25	.249	.299	.278	.099	.119	.107

Table A-9: Accuracy for equally weighted groups with LC pattern of 3 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	X^2 $\alpha=.05$	G^2 $\alpha=.05$	I^2 $\alpha=.05$	X^2 $\alpha=.01$	G^2 A=.01	I^2 $\alpha=.01$
500	5	.05	.953	.945	.952	.989	.984	.992
		.25	.965	.913	.949	.989	.979	.988
	10	.05	.953	.944	.958	.992	.989	.992
		.25	.965	.922	.943	.986	.970	.979
	25	.05	.943	.928	.945	.987	.987	.989
		.25	.962	.924	.948	.983	.967	.980

Table A-10: Accuracy for equally weighted groups with LC pattern of 1 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	X^2 $\alpha=.05$	G^2 $\alpha=.05$	I^2 $\alpha=.05$	X^2 $\alpha=.01$	G^2 $\alpha=.01$	I^2 $\alpha=.01$
3000	5	.05	.944	.938	.943	.983	.985	.985
		.25	.636	.840	.791	.433	.723	.604
	10	.05	.958	.957	.958	.991	.991	.991
		.25	.604	.858	.811	.382	.720	.594
	25	.05	.942	.945	.945	.987	.986	.988
		.25	.627	.867	.812	.445	.738	.624

Table A-11: Accuracy for equally weighted groups with LC pattern of 2 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	X^2 $\alpha=.05$	G^2 $\alpha=.05$	I^2 $\alpha=.05$	X^2 $\alpha=.01$	G^2 $\alpha=.01$	I^2 $\alpha=.01$
3000	5	.05	.968	.968	.970	.993	.995	.994
		.25	.930	.939	.942	.860	.922	.905
	10	.05	.948	.947	.950	.990	.992	.990
		.25	.917	.921	.927	.838	.904	.890
	25	.05	.951	.955	.955	.992	.992	.993
		.25	.899	.890	.894	.832	.823	.828

Table A-12: Accuracy for equally weighted groups with LC pattern of 3 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	X^2 $\alpha=.05$	G^2 $\alpha=.05$	I^2 $\alpha=.05$	X^2 $\alpha=.01$	G^2 $\alpha=.01$	I^2 $\alpha=.01$
3000	5	.05	.947	.948	.945	.986	.983	.986
		.25	.962	.841	.898	.992	.921	.965
	10	.05	.953	.954	.953	.989	.989	.988
		.25	.964	.825	.907	.994	.917	.964
	25	.05	.961	.958	.960	.993	.995	.994
		.25	.970	.824	.893	.996	.918	.971

Table A-13: Accuracy for unequally weighted groups by sample size

Sample Size	Cluster Size	LC Pattern	Item Error Rate	X^{2*} $\alpha=.05$	G^{2*} $\alpha=.05$	I^{2*} $\alpha=.05$	X^{2*} $\alpha=.01$	G^{2*} $\alpha=.01$	I^{2*} $\alpha=.01$
500	5	1	.05	.161	.194	.173	.044	.047	.040
			.25	.031	.041	.034	.004	.009	.004
		2	.05	.638	.641	.644	.401	.373	.387
			.25	.071	.072	.069	.014	.015	.014
		3	.05	.953	.974	.966	.990	.997	.997
			.25	.937	.928	.938	.849	.855	.856
	10	1	.05	.148	.187	.163	.040	.050	.040
			.25	.025	.042	.027	.001	.002	.002
		2	.05	.642	.624	.636	.420	.392	.414
			.25	.087	.085	.086	.017	.016	.016
		3	.05	.953	.963	.963	.987	.995	.994
			.25	.931	.911	.927	.843	.829	.835
	25	1	.05	.177	.220	.187	.054	.071	.055
			.25	.027	.038	.032	.005	.005	.005
		2	.05	.643	.641	.645	.453	.430	.442
			.25	.066	.063	.063	.019	.018	.019
		3	.05	.943	.963	.962	.985	.997	.993
			.25	.919	.911	.917	.857	.850	.849
3000	5	1	.05	.977	.984	.980	.961	.968	.963
			.25	.298	.360	.336	.111	.149	.130
		2	.05	.992	.996	.993	.999	1.0	.999
			.25	.697	.701	.708	.430	.441	.433
		3	.05	.972	.975	.973	.997	.996	.997
			.25	.961	.933	.959	.992	.981	.991
	10	1	.05	.975	.981	.979	.953	.957	.955
			.25	.306	.356	.331	.104	.154	.130
		2	.05	.988	.991	.990	.998	1.0	.998
			.25	.670	.684	.677	.424	.415	.417
		3	.05	.969	.974	.970	.999	.998	.999
			.25	.972	.934	.963	.993	.986	.993
	25	1	.05	.972	.976	.973	.943	.949	.945
			.25	.310	.368	.314	.103	.156	.129
		2	.05	.990	.994	.992	.999	1.0	1.0
			.25	.729	.732	.732	.472	.482	.478
		3	.05	.974	.976	.974	.995	.993	.995
			.25	.970	.824	.893	.996	.918	.971

Table A-14: Accuracy for unequally weighted groups with item error rate of .05

Sample Size	Cluster Size	LC Pattern	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
500	5	1	.161	.194	.173	.044	.047	.040
		2	.638	.641	.644	.401	.373	.387
		3	.953	.974	.966	.990	.997	.997
	10	1	.148	.187	.163	.040	.050	.040
		2	.642	.624	.636	.420	.392	.414
		3	.953	.963	.963	.987	.995	.994
	25	1	.177	.220	.187	.054	.071	.055
		2	.643	.641	.645	.453	.430	.442
		3	.943	.963	.962	.985	.997	.993
3000	5	1	.977	.984	.980	.961	.968	.963
		2	.992	.996	.993	.999	1.0	.999
		3	.972	.975	.973	.997	.996	.997
	10	1	.975	.981	.979	.953	.957	.955
		2	.988	.991	.990	.998	1.0	.998
		3	.969	.974	.970	.999	.998	.999
	25	1	.972	.976	.973	.943	.949	.945
		2	.990	.994	.992	.999	1.0	1.0
		3	.974	.976	.974	.995	.993	.995

Table A-15: Accuracy for unequally weighted groups with item error rate of .25

Sample Size	Cluster Size	LC Pattern	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
500	5	1	.031	.041	.034	.004	.009	.004
		2	.071	.072	.069	.014	.015	.014
		3	.937	.928	.938	.849	.855	.856
	10	1	.025	.042	.027	.001	.002	.002
		2	.087	.085	.086	.017	.016	.016
		3	.931	.911	.927	.843	.829	.835
	25	1	.027	.038	.032	.005	.005	.005
		2	.066	.063	.063	.019	.018	.019
		3	.919	.911	.917	.857	.850	.849
3000	5	1	.298	.360	.336	.111	.149	.130
		2	.697	.701	.708	.430	.441	.433
		3	.961	.933	.959	.992	.981	.991
	10	1	.306	.356	.331	.104	.154	.130
		2	.670	.684	.677	.424	.415	.417
		3	.972	.934	.963	.993	.986	.993
	25	1	.310	.368	.314	.103	.156	.129
		2	.729	.732	.732	.472	.482	.478
		3	.964	.936	.958	.997	.985	.995

Table A-16: Accuracy for unequally weighted groups with cluster size of 5

Item Error Rate	Sample Size	LC Pattern	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
.05	500	1	.161	.194	.173	.044	.047	.040
		2	.638	.641	.644	.401	.373	.387
		3	.953	.974	.966	.990	.997	.997
	3000	1	.977	.984	.980	.961	.968	.963
		2	.992	.996	.993	.999	1.0	.999
		3	.972	.975	.973	.997	.996	.997
.25	500	1	.031	.041	.034	.004	.009	.004
		2	.071	.072	.069	.014	.015	.014
		3	.937	.928	.938	.849	.855	.856
	3000	1	.298	.360	.336	.111	.149	.130
		2	.697	.701	.708	.430	.441	.433
		3	.961	.933	.959	.992	.981	.991

Table A-17: Accuracy for unequally weighted groups with cluster size of 10

Item Error Rate	Sample Size	LC Pattern	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
.05	500	1	.148	.187	.163	.040	.050	.040
		2	.642	.624	.636	.420	.392	.414
		3	.953	.963	.963	.987	.995	.994
	3000	1	.975	.981	.979	.953	.957	.955
		2	.988	.991	.990	.998	1.0	.998
		3	.969	.974	.970	.999	.998	.999
.25	500	1	.025	.042	.027	.001	.002	.002
		2	.087	.085	.086	.017	.016	.016
		3	.931	.911	.927	.843	.829	.835
	3000	1	.306	.356	.331	.104	.154	.130
		2	.670	.684	.677	.424	.415	.417
		3	.972	.934	.963	.993	.986	.993

Table A- 18: Accuracy for unequally weighted groups with cluster size of 25

Item Error Rate	Sample Size	LC Pattern	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
.05	500	1	.177	.220	.187	.054	.071	.055
		2	.643	.641	.645	.453	.430	.442
		3	.943	.963	.962	.985	.997	.993
	3000	1	.972	.976	.973	.943	.949	.945
		2	.990	.994	.992	.999	1.0	1.0
		3	.974	.976	.974	.995	.993	.995
.25	500	1	.931	.911	.927	.843	.829	.835
		2	.066	.063	.063	.019	.018	.019
		3	.919	.911	.917	.857	.850	.849
	3000	1	.310	.368	.314	.103	.156	.129
		2	.729	.732	.732	.472	.482	.478
		3	.964	.936	.958	.997	.985	.995

Table A-19: Accuracy for unequally weighted groups with LC pattern of 1 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
500	5	.05	.161	.194	.173	.044	.047	.040
		.25	.031	.041	.034	.004	.009	.004
	10	.05	.148	.187	.163	.040	.050	.040
		.25	.025	.042	.027	.001	.002	.002
	25	.05	.177	.220	.187	.054	.071	.055
		.25	.027	.038	.032	.005	.005	.005

Table A-20: Accuracy for unequally weighted groups with LC pattern of 2 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
500	5	.05	.638	.641	.644	.401	.373	.387
		.25	.071	.072	.069	.014	.015	.014
	10	.05	.642	.624	.636	.420	.392	.414
		.25	.087	.085	.086	.017	.016	.016
	25	.05	.643	.641	.645	.453	.430	.442
		.25	.066	.063	.063	.019	.018	.019

Table A-21: Accuracy for unequally weighted groups with LC pattern of 3 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
500	5	.05	.953	.974	.966	.990	.997	.997
		.25	.937	.928	.938	.849	.855	.856
	10	.05	.953	.963	.963	.987	.995	.994
		.25	.931	.911	.927	.843	.829	.835
	25	.05	.943	.963	.962	.985	.997	.993
		.25	.919	.911	.917	.857	.850	.849

Table A-22: Accuracy for unequally weighted groups with LC pattern of 1 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
3000	5	.05	.977	.984	.980	.961	.968	.963
		.25	.298	.360	.336	.111	.149	.130
	10	.05	.975	.981	.979	.953	.957	.955
		.25	.306	.356	.331	.104	.154	.130
	25	.05	.972	.976	.973	.943	.949	.945
		.25	.310	.368	.314	.103	.156	.129

Table A-23: Accuracy for unequally weighted groups with LC pattern of 2 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
3000	5	.05	.992	.996	.993	.999	1.0	.999
		.25	.697	.701	.708	.430	.441	.433
	10	.05	.988	.991	.990	.998	1.0	.998
		.25	.670	.684	.677	.424	.415	.417
	25	.05	.990	.994	.992	.999	1.0	1.0
		.25	.729	.732	.732	.472	.482	.478

Table A- 24: Accuracy for unequally weighted groups with LC pattern of 3 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	X^{2*} α=.05	G^{2*} α=.05	I^{2*} α=.05	X^{2*} α=.01	G^{2*} α=.01	I^{2*} α=.01
3000	5	.05	.972	.975	.973	.997	.996	.997
		.25	.961	.933	.959	.992	.981	.991
	10	.05	.969	.974	.970	.999	.998	.999
		.25	.972	.934	.963	.993	.986	.993
	25	.05	.974	.976	.974	.995	.993	.995
		.25	.964	.936	.958	.997	.985	.995

Table A-25: Accuracy for equally weighted groups

Sample Size	Cluster Size	LC Pattern	Item Error Rate	AIC	BIC	RIC
500	5	1	.05	.569	.004	.157
			.25	.208	.000	.019
		2	.05	.814	.111	.637
			.25	.305	.000	.042
		3	.05	.901	1.0	.994
			.25	.852	.661	.967
	10	1	.05	.562	.007	.188
			.25	.210	.000	.024
		2	.05	.835	.112	.635
			.25	.319	.000	.043
		3	.05	.892	1.0	.995
			.25	.880	.666	.968
	25	1	.05	.574	.011	.231
			.25	.201	.000	.015
		2	.05	.841	.129	.662
			.25	.348	.001	.053
		3	.05	.873	1.0	.992
			.25	.871	.659	.968
3000	5	1	.05	.899	.702	.968
			.25	.846	.009	.167
		2	.05	.929	1.0	1.0
			.25	.902	.080	.514
		3	.05	.899	1.0	1.0
			.25	.802	1.0	.987
	10	1	.05	.923	.694	.969
			.25	.866	.004	.160
		2	.05	.916	1.0	1.0
			.25	.877	.082	.494
		3	.05	.902	1.0	1.0
			.25	.781	.999	.988
	25	1	.05	.905	.713	.964
			.25	.871	.008	.185
		2	.05	.910	1.0	1.0
			.25	.865	.044	.337
		3	.05	.916	1.0	1.0
			.25	.766	1.0	.991

Table A-26: Accuracy for equally weighted groups with item error rate of .05

Sample Size	Cluster Size	LC Pattern	AIC	BIC	RIC
500	5	1	.569	.004	.157
		2	.814	.111	.637
		3	.901	1.0	.994
	10	1	.562	.007	.188
		2	.835	.112	.635
		3	.892	1.0	.995
	25	1	.574	.011	.231
		2	.841	.129	.662
		3	.873	1.0	.992
3000	5	1	.899	.702	.968
		2	.929	1.0	1.0
		3	.899	1.0	1.0
	10	1	.923	.694	.969
		2	.916	1.0	1.0
		3	.902	1.0	1.0
	25	1	.905	.713	.964
		2	.910	1.0	1.0
		3	.916	1.0	1.0

Table A-27: Accuracy for equally weighted groups with item error rate of .25

Sample Size	Cluster Size	LC Pattern	AIC	BIC	RIC
500	5	1	.208	.000	.019
		2	.305	.000	.042
		3	.852	.661	.967
	10	1	.210	.000	.024
		2	.319	.000	.043
		3	.880	.666	.968
	25	1	.201	.000	.015
		2	.348	.001	.053
		3	.871	.659	.968
3000	5	1	.846	.009	.167
		2	.902	.080	.514
		3	.802	1.0	.987
	10	1	.866	.004	.160
		2	.877	.082	.494
		3	.781	.999	.988
	25	1	.871	.008	.185
		2	.865	.044	.337
		3	.766	1.0	.991

Table A-28: Accuracy for equally weighted groups with cluster size of 5

Item Error Rate	Sample Size	LC Pattern	AIC	BIC	RIC
.05	500	1	.569	.004	.157
		2	.814	.111	.637
		3	.901	1.0	.994
	3000	1	.899	.702	.968
		2	.929	1.0	1.0
		3	.899	1.0	1.0
.25	500	1	.208	.000	.019
		2	.305	.000	.042
		3	.852	.661	.967
	3000	1	.846	.009	.167
		2	.902	.080	.514
		3	.802	1.0	.987

Table A-29: Accuracy for equally weighted groups with cluster size of 10

Item Error Rate	Sample Size	LC Pattern	AIC	BIC	RIC
.05	500	1	.562	.007	.188
		2	.835	.112	.635
		3	.892	1.0	.995
	3000	1	.923	.694	.969
		2	.916	1.0	1.0
		3	.902	1.0	1.0
.25	500	1	.210	.000	.024
		2	.319	.000	.043
		3	.880	.666	.968
	3000	1	.866	.004	.160
		2	.877	.082	.494
		3	.781	.999	.988

Table A-30: Accuracy for equally weighted groups with cluster size of 25

CPs	Sample Size	LC Pattern	AIC	BIC	RIC
.95	500	1	.574	.011	.231
		2	.841	.129	.662
		3	.873	1.0	.992
	3000	1	.574	.011	.231
		2	.841	.129	.662
		3	.873	1.0	.992
.75	500	1	.201	.000	.015
		2	.348	.001	.053
		3	.871	.659	.968
	3000	1	.871	.008	.185
		2	.865	.044	.337
		3	.766	1.0	.991

Table A-31: Accuracy for equally weighted groups with LC pattern of 1 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	AIC	BIC	RIC
500	5	.05	.569	.004	.157
		.25	.208	.000	.019
	10	.05	.562	.007	.188
		.25	.210	.000	.024
	25	.05	.574	.011	.231
		.25	.201	.000	.015

Table A-32: Accuracy for equally weighted groups with LC pattern of 2 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	AIC	BIC	RIC
500	5	.05	.814	.111	.637
		.25	.305	.000	.042
	10	.05	.835	.112	.635
		.25	.319	.000	.043
	25	.05	.841	.129	.662
		.25	.348	.001	.053

Table A-33: Accuracy for equally weighted groups with LC pattern of 3 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	AIC	BIC	RIC
500	5	.05	.901	1.0	.994
		.25	.852	.661	.967
	10	.05	.892	1.0	.995
		.25	.880	.666	.968
	25	.05	.873	1.0	.992
		.25	.871	.659	.968

Table A-34: Accuracy for equally weighted groups with LC pattern of 1 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	AIC	BIC	RIC
3000	5	.05	.899	.702	.968
		.25	.846	.009	.167
	10	.05	.923	.694	.969
		.25	.866	.004	.160
	25	.05	.905	.713	.964
		.25	.871	.008	.185

Table A-35: Accuracy for equally weighted groups with LC pattern of 2 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	AIC	BIC	RIC
3000	5	.05	.929	1.0	1.0
		.25	.902	.080	.514
	10	.05	.916	1.0	1.0
		.25	.877	.082	.494
	25	.05	.910	1.0	1.0
		.25	.865	.044	.337

Table A-36: Accuracy for equally weighted groups with LC pattern of 3 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	AIC	BIC	RIC
3000	5	.05	.899	1.0	1.0
		.25	.802	1.0	.987
	10	.05	.902	1.0	1.0
		.25	.781	.999	.988
	25	.05	.916	1.0	1.0
		.25	.766	1.0	.991

Table A-37: Accuracy for unequally weighted groups

Sample Size	Cluster Size	LC Pattern	Item Error Rate	AIC*	BIC*	RIC*
500	5	1	.05	.261	.000	.008
			.25	.055	.000	.000
		2	.05	.702	.008	.210
			.25	.096	.000	.005
		3	.05	.949	.970	.999
			.25	.913	.098	.687
	10	1	.05	.240	.000	.013
			.25	.060	.000	.002
		2	.05	.691	.005	.201
			.25	.126	.000	.005
		3	.05	.945	.966	.997
			.25	.893	.078	.663
	25	1	.05	.275	.001	.023
			.25	.051	.000	.001
		2	.05	.695	.005	.247
			.25	.093	.000	.003
		3	.05	.933	.973	.998
			.25	.896	.088	.699
3000	5	1	.05	.971	.073	.563
			.25	.435	.000	.004
		2	.05	.983	.946	1.0
			.25	.760	.000	.039
		3	.05	.945	1.0	1.0
			.25	.894	1.0	.999
	10	1	.05	.972	.073	.568
			.25	.427	.000	.002
		2	.05	.977	.957	1.0
			.25	.734	.000	.032
		3	.05	.946	1.0	1.0
			.25	.899	1.0	.999
	25	1	.05	.972	.066	.543
			.25	.445	.000	.004
		2	.05	.985	.938	.999
			.25	.781	.000	.029
		3	.05	.955	1.0	1.0
			.25	.899	1.0	1.0

Table A-38: Accuracy for unequally weighted groups with item error rate of .05

Sample Size	Cluster Size	LC Pattern	AIC*	BIC*	RIC*
500	5	1	.261	.000	.008
		2	.702	.008	.210
		3	.949	.970	.999
	10	1	.240	.000	.013
		2	.691	.005	.201
		3	.945	.966	.997
	25	1	.275	.001	.023
		2	.695	.005	.247
		3	.933	.973	.998
3000	5	1	.971	.073	.563
		2	.983	.946	1.0
		3	.945	1.0	1.0
	10	1	.972	.073	.568
		2	.977	.957	1.0
		3	.946	1.0	1.0
	25	1	.972	.066	.543
		2	.985	.938	.999
		3	.955	1.0	1.0

Table A-39: Accuracy for unequally weighted groups with item error rate of .25

Sample Size	Cluster Size	LC Pattern	AIC*	BIC*	RIC*
500	5	1	.055	.000	.000
		2	.096	.000	.005
		3	.913	.098	.687
	10	1	.060	.000	.002
		2	.126	.000	.005
		3	.893	.078	.663
	25	1	.051	.000	.001
		2	.093	.000	.003
		3	.896	.088	.699
3000	5	1	.435	.000	.004
		2	.760	.000	.039
		3	.894	1.0	.999
	10	1	.427	.000	.002
		2	.734	.000	.032
		3	.899	1.0	.999
	25	1	.445	.000	.004
		2	.781	.000	.029
		3	.899	1.0	1.0

Table A-40: Accuracy for unequally weighted groups with cluster size of 5

Item Error Rate	Sample Size	LC Pattern	AIC*	BIC*	RIC*
.05	500	1	.261	.000	.008
		2	.702	.008	.210
		3	.949	.970	.999
	3000	1	.971	.073	.563
		2	.983	.946	1.0
		3	.945	1.0	1.0
.25	500	1	.055	.000	.000
		2	.096	.000	.005
		3	.913	.098	.687
	3000	1	.435	.000	.004
		2	.760	.000	.039
		3	.894	1.0	.999

Table A-41: Accuracy for unequally weighted groups with cluster size of 10

Item Error Rate	Sample Size	LC Pattern	AIC*	BIC*	RIC*
.05	500	1	.240	.000	.013
		2	.691	.005	.201
		3	.945	.966	.997
	3000	1	.972	.073	.568
		2	.977	.957	1.0
		3	.946	1.0	1.0
.25	500	1	.060	.000	.002
		2	.126	.000	.005
		3	.893	.078	.663
	3000	1	.427	.000	.002
		2	.734	.000	.032
		3	.899	1.0	.999

Table A-42: Accuracy for unequally weighted groups with cluster size of 25

Item Error Rate	Sample Size	LC Pattern	AIC*	BIC*	RIC*
.05	500	1	.275	.001	.023
		2	.695	.005	.247
		3	.933	.973	.998
	3000	1	.972	.066	.543
		2	.985	.938	.999
		3	.955	1.0	1.0
.25	500	1	.051	.000	.001
		2	.093	.000	.003
		3	.896	.088	.699
	3000	1	.445	.000	.004
		2	.781	.000	.029
		3	.899	1.0	1.0

Table A-43: Accuracy for unequally weighted groups with LC pattern of 1 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	AIC*	BIC*	RIC*
500	5	.05	.261	.000	.008
		.25	.055	.000	.000
	10	.05	.240	.000	.013
		.25	.060	.000	.002
	25	.05	.275	.001	.023
		.25	.051	.000	.001

Table A-44: Accuracy for unequally weighted groups with LC pattern of 2 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	AIC*	BIC*	RIC*
500	5	.05	.702	.008	.210
		.25	.096	.000	.005
	10	.05	.691	.005	.201
		.25	.126	.000	.005
	25	.05	.695	.005	.247
		.25	.093	.000	.003

Table A-45: Accuracy for unequally weighted groups with LC pattern of 3 and sample size of 500

Sample Size	Cluster Size	Item Error Rate	AIC*	BIC*	RIC*
500	5	.05	.949	.970	.999
		.25	.913	.098	.687
	10	.05	.945	.966	.997
		.25	.893	.078	.663
	25	.05	.933	.973	.998
		.25	.896	.088	.699

Table A-46: Accuracy for unequally weighted groups with LC pattern of 1 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	AIC*	BIC*	RIC*
3000	5	.05	.971	.073	.563
		.25	.435	.000	.004
	10	.05	.972	.073	.568
		.25	.427	.000	.002
	25	.05	.972	.066	.543
		.25	.445	.000	.004

Table A-47: Accuracy for unequally weighted groups with LC pattern of 2 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	AIC*	BIC*	RIC*
3000	5	.05	.983	.946	1.0
		.25	.760	.000	.039
	10	.05	.977	.957	1.0
		.25	.734	.000	.032
	25	.05	.985	.938	.999
		.25	.781	.000	.029

Table A-48: Accuracy for unequally weighted groups with LC pattern of 3 and sample size of 3000

Sample Size	Cluster Size	Item Error Rate	AIC*	BIC*	RIC*
3000	5	.05	.945	1.0	1.0
		.25	.894	1.0	.999
	10	.05	.946	1.0	1.0
		.25	.899	1.0	.999
	25	.05	.955	1.0	1.0
		.25	.899	1.0	1.0

APPENDIX B-1: CHI-SQUARE, AIC, BIC AND RIC SAS PROGRAM

```
libname savelib "c:\final";
options nosymbolgen nomprint nomlogic noNOTES;
%let cp=.95;
DATA _NULL_;CALL SYMPUT('START',PUT(TIME(),TIME8.));RUN;
%put &START;
data blank16;input y;cards;
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16;run;
data blank16_2;input y2;cards;
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16;run;
%macro FIRST;%do  k=1 %to 1000;

/*CREATE DATA*/
data cmg1;do i = 1 to 100;          /*create Dirichlet values*/
gamma1=rangam(0,8);gamma2=rangam(0,3);gamma3=rangam(0,8);
gamma4=rangam(0,50);
gamma5=rangam(0,23);gamma6=rangam(0,8);gamma7=rangam(0,10);
gamma8=rangam(0,5);gamma9=rangam(0,10);gamma10=rangam(0,40);
gamma11=rangam(0,25);gamma12=rangam(0,10);
total1=sum(of gamma1-gamma6);
     d1=gamma1/total1;d2=gamma2/total1;d3=gamma3/total1;
     d4=gamma4/total1;d5=gamma5/total1;d6=gamma6/total1;
total2=sum(of gamma7-gamma12);
     d7=gamma7/total2;d8=gamma8/total2;d9=gamma9/total2;
     d10=gamma10/total2;d11=gamma11/total2;d12=gamma12/total2;

do j=1 to 80;     /*create clusters*/
     x=ranuni(0);
       if x le D1 then c=1;else if D1 lt x le D1+D2 then c=2;
else if D1+D2 lt x le D1+D2+D3 then c=3;
else if D1+D2+D3 lt x le D1+D2+D3+D4 then c=4;
else if (D1+D2+D3+D4) lt x le D1+D2+D3+D4+D5 then c=5;
else if x gt D1+D2+D3+D4+D5 then c=6;
     x6=ranuni(0);
       if x6 le D7 then c2=1;else if D7 lt x6 le D7+D8 then c2=2;
else if D7+D8 lt x6 le D7+D8+D9 then c2=3;
else if D7+D8+D9 lt x6 le D7+D8+D9+D10 then c2=4;
else if (D7+D8+D9+D10) lt x6 le D7+D8+D9+D10+D11 then c2=5;
else if x6 gt D7+D8+D9+D10+D11 then c2=6;output;end;end;
data cmg1(keep=V1 V2 V3 V4 V12 V22 V32 V42);set cmg1(keep=c c2);
     x2=ranuni(0);x3=ranuni(0);x4=ranuni(0);x5=ranuni(0);cp=&cp;
if c=1 then do;
if x2 le cp then v1=0; else v1=1;if x3 le cp then v2=0; else 2=1;
if x4 le cp then v3=0; else v3=1;if x5 le cp then v4=0; else 4=1;
end;
if c=2 then do;if x2 le cp then v1=0; else v1=1;if x3 le cp then
v2=1;
else v2=0;if x4 le cp then v3=0; else v3=1;if x5 le cp then v4=0;
else v4=1; end;
```

116

```
if c=3 then do;
if x2 le cp then v1=1; else v1=0;if x3 le cp then v2=0;else v2=1;
if x4 le cp then v3=0; else v3=1;if x5 le cp then v4=0;else v4=1;
end;
if c=4 then do;
if x2 le cp then v1=1; else v1=0;if x3 le cp then v2=1;else v2=0;
if x4 le cp then v3=0; else v3=1;if x5 le cp then v4=0;else v4=1;
end;
if c=5 then do;
if x2 le cp then v1=1; else v1=0;if x3 le cp then v2=1;else v2=0;
if x4 le cp then v3=1; else v3=0;if x5 le cp then v4=0;else v4=1;
end;
if c=6 then do;
if x2 le cp then v1=1; else v1=0;if x3 le cp then v2=1;else v2=0;
if x4 le cp then v3=1; else v3=0;if x5 le cp then v4=1;else v4=0;
end;
x7=ranuni(0);x8=ranuni(0);x9=ranuni(0);x10=ranuni(0);
if c2=1 then do;
if x7 le cp then v12=0; else v12=1;if x8 le cp then v22=0;
else v22=1;
if x9 le cp then v32=0; else v32=1;if x10 le cp then v42=0;
else v42=1; end;
if c2=2 then do;
if x7 le cp then v12=0; else v12=1;if x8 le cp then v22=1;
else v22=0;
if x9 le cp then v32=0; else v32=1;if x10 le cp then v42=0;
else v42=1; end;
if c2=3 then do;
if x7 le cp then v12=1; else v12=0;if x8 le cp then v22=0;
else v22=1;
if x9 le cp then v32=0; else v32=1;if x10 le cp then v42=0;
else v42=1; end;
if c2=4 then do;
if x7 le cp then v12=1; else v12=0;if x8 le cp then v22=1;
else v22=0;
if x9 le cp then v32=0; else v32=1;if x10 le cp then v42=0;
else v42=1; end;
if c2=5 then do;
if x7 le cp then v12=1; else v12=0;if x8 le cp then v22=1;
else v22=0;
if x9 le cp then v32=1; else v32=0;if x10 le cp then v42=0;
else v42=1; end;
if c2=6 then do;
if x7 le cp then v12=1; else v12=0;if x8 le cp then v22=1;
else v22=0;
if x9 le cp then v32=1; else v32=0;if x10 le cp then v42=1;
else v42=0; end;run;

%let LC1=.08;          /*HETEROGENEOUS MODEL*/
%let LC2=.03;%let LC3=.08;%let LC4=.50;%let LC5=.23;
%let LC6=.08;%let LC12=.10;
%let LC22=.05;%let LC32=.10;%let LC42=.40;%let LC52=.25;
```

```
%let LC62=.10;
%let CPA1=.95;%let CPB1=.95;%let CPC1=.95;%let CPD1=.95;
%let CPA2=.05;%let CPB2=.05;%let CPC2=.05;%let CPD2=.05;
%let CPA1_2=.95;%let CPB1_2=.95;%let CPC1_2=.95;%let CPD1_2=.95;
%let CPA2_2=.05;%let CPB2_2=.05;%let CPC2_2=.05;%let CPD2_2=.05;
data cmg1(keep=y y2); set cmg1;
%let cnt=0;%let endvar=0;%let j=25;
IF V1=0 AND V2=0 AND V3=0 AND V4=0 THEN Y=1; ELSE
IF V1=0 AND V2=0 AND V3=0 AND V4=1 THEN Y=2; ELSE
IF V1=0 AND V2=0 AND V3=1 AND V4=0 THEN y=3; ELSE
IF V1=0 AND V2=0 AND V3=1 AND V4=1 THEN y=4; ELSE
IF V1=0 AND V2=1 AND V3=0 AND V4=0 THEN y=5; ELSE
IF V1=0 AND V2=1 AND V3=0 AND V4=1 THEN y=6; ELSE
IF V1=0 AND V2=1 AND V3=1 AND V4=0 THEN y=7; ELSE
IF V1=0 AND V2=1 AND V3=1 AND V4=1 THEN y=8; ELSE
IF V1=1 AND V2=0 AND V3=0 AND V4=0 THEN y=9; ELSE
IF V1=1 AND V2=0 AND V3=0 AND V4=1 THEN y=10; ELSE
IF V1=1 AND V2=0 AND V3=1 AND V4=0 THEN y=11; ELSE
IF V1=1 AND V2=0 AND V3=1 AND V4=1 THEN y=12; ELSE
IF V1=1 AND V2=1 AND V3=0 AND V4=0 THEN y=13; ELSE
IF V1=1 AND V2=1 AND V3=0 AND V4=1 THEN y=14; ELSE
IF V1=1 AND V2=1 AND V3=1 AND V4=0 THEN y=15; ELSE
IF V1=1 AND V2=1 AND V3=1 AND V4=1 THEN y=16;
            IF V12=0 AND V22=0 AND V32=0 AND V42=0 THEN Y2=1; ELSE
            IF V12=0 AND V22=0 AND V32=0 AND V42=1 THEN Y2=2; ELSE
            IF V12=0 AND V22=0 AND V32=1 AND V42=0 THEN Y2=3; ELSE
            IF V12=0 AND V22=0 AND V32=1 AND V42=1 THEN Y2=4; ELSE
            IF V12=0 AND V22=1 AND V32=0 AND V42=0 THEN Y2=5; ELSE
            IF V12=0 AND V22=1 AND V32=0 AND V42=1 THEN Y2=6; ELSE
            IF V12=0 AND V22=1 AND V32=1 AND V42=0 THEN Y2=7; ELSE
            IF V12=0 AND V22=1 AND V32=1 AND V42=1 THEN Y2=8; ELSE
            IF V12=1 AND V22=0 AND V32=0 AND V42=0 THEN Y2=9; ELSE
            IF V12=1 AND V22=0 AND V32=0 AND V42=1 THEN Y2=10;ELSE
            IF V12=1 AND V22=0 AND V32=1 AND V42=0 THEN Y2=11;ELSE
            IF V12=1 AND V22=0 AND V32=1 AND V42=1 THEN Y2=12;ELSE
            IF V12=1 AND V22=1 AND V32=0 AND V42=0 THEN Y2=13;ELSE
            IF V12=1 AND V22=1 AND V32=0 AND V42=1 THEN Y2=14;ELSE
            IF V12=1 AND V22=1 AND V32=1 AND V42=0 THEN Y2=15;ELSE
            IF V12=1 AND V22=1 AND V32=1 AND V42=1 THEN Y2=16;RUN;
data cmg2; set cmg1;notagain1=0; notagain2=0;
data blank16;set blank16; notagain1=0; notagain2=0;
data blank16_2;set blank16_2; notagain1=0; notagain2=0;run;
PROC FREQ NOPRINT data=cmg2; TABLES Y /
OUT=YFREQ(DROP=PERCENT);RUN;
PROC SORT DATA=CMG2; BY Y; run;
DATA CMG3;MERGE YFREQ(RENAME=(COUNT=CNT1))
CMG2;BY Y;if first.Y;run;
data cmg3;MERGE CMG3 blank16; BY Y;RUN;
data cmg3 (drop=y2);SET CMG3; if cnt1=. then cnt1=0;RUN;
PROC SORT DATA=CMG2; BY Y2; run;
PROC FREQ  NOPRINT data=cmg2; TABLES Y2 /
OUT=Y2FREQ(DROP=PERCENT);RUN;
```

```
DATA CMG32(keep=y2 cnt2);MERGE Y2FREQ(RENAME=(COUNT=cnt2))
CMG2;BY Y2;if first.Y2;run;
proc sort data=cmg32; by y2;
proc sort data=blank16_2; by y2; run;
data cmg32;MERGE CMG32 blank16_2; BY Y2;RUN;
DATA CMG3; SET CMG3;YH=Y; all=1;
DATA CMG32; SET CMG32;YH=Y2; all=1;
proc SORT data=CMG3; BY YH;
proc SORT data=CMG32; BY YH;run;
DATA CMG3; MERGE CMG3 CMG32; BY YH;run;
%macro cm;
%do %until (&endvar=1);
%let OLDLC1 = &LC1;%let OLDLC2 = &LC2;%let OLDLC3 = &LC3;
%let OLDLC4 = &LC4;%let OLDLC5 = &LC5;%let OLDLC6 = &LC6;
%let OLDCPA1 = &CPA1;%let OLDCPB1 = &CPB1;
%let OLDCPC1 = &CPC1;%let OLDCPD1 = &CPD1;
data cmg3; set cmg3(keep=cnt1 y y2 yh all cnt2 notagain1
notagain2);
retain grpcnt1 0 bigl1 0 bigl2 0 bigl3 0 bigl4 0 bigl5 0 bigl6 0;
    if Y=1 then do;
P1=&LC1*&CPA1*&CPB1*&CPC1*&CPD1;P2=&LC2*&CPA1*&CPB2*&CPC1*&CPD1;
P3=&LC3*&CPA2*&CPB1*&CPC1*&CPD1;P4=&LC4*&CPA2*&CPB2*&CPC1*&CPD1;
P5=&LC5*&CPA2*&CPB2*&CPC2*&CPD1;P6=&LC6*&CPA2*&CPB2*&CPC2*&CPD2;
end;else if Y=2 then do;
P1=&LC1*&CPA1*&CPB1*&CPC1*&CPD2;P2=&LC2*&CPA1*&CPB2*&CPC1*&CPD2;
P3=&LC3*&CPA2*&CPB1*&CPC1*&CPD2;P4=&LC4*&CPA2*&CPB2*&CPC1*&CPD2;
P5=&LC5*&CPA2*&CPB2*&CPC2*&CPD2;P6=&LC6*&CPA2*&CPB2*&CPC2*&CPD1;
end;else if Y=3 then do;
P1=&LC1*&CPA1*&CPB1*&CPC2*&CPD1;P2=&LC2*&CPA1*&CPB2*&CPC2*&CPD1;
P3=&LC3*&CPA2*&CPB1*&CPC2*&CPD1;P4=&LC4*&CPA2*&CPB2*&CPC2*&CPD1;
P5=&LC5*&CPA2*&CPB2*&CPC1*&CPD1;P6=&LC6*&CPA2*&CPB2*&CPC1*&CPD2;
end;else if Y=4 then do;
P1=&LC1*&CPA1*&CPB1*&CPC2*&CPD2;P2=&LC2*&CPA1*&CPB2*&CPC2*&CPD2;
P3=&LC3*&CPA2*&CPB1*&CPC2*&CPD2;P4=&LC4*&CPA2*&CPB2*&CPC2*&CPD2;
P5=&LC5*&CPA2*&CPB2*&CPC1*&CPD2;P6=&LC6*&CPA2*&CPB2*&CPC1*&CPD1;
end;else if Y=5 then do;
P1=&LC1*&CPA1*&CPB2*&CPC1*&CPD1;P2=&LC2*&CPA1*&CPB1*&CPC1*&CPD1;
P3=&LC3*&CPA2*&CPB2*&CPC1*&CPD1;P4=&LC4*&CPA2*&CPB1*&CPC1*&CPD1;
P5=&LC5*&CPA2*&CPB1*&CPC2*&CPD1;P6=&LC6*&CPA2*&CPB1*&CPC2*&CPD2;
end;else if Y=6 then do;
P1=&LC1*&CPA1*&CPB2*&CPC1*&CPD2;P2=&LC2*&CPA1*&CPB1*&CPC1*&CPD2;
P3=&LC3*&CPA2*&CPB2*&CPC1*&CPD2;P4=&LC4*&CPA2*&CPB1*&CPC1*&CPD2;
P5=&LC5*&CPA2*&CPB1*&CPC2*&CPD2;P6=&LC6*&CPA2*&CPB1*&CPC2*&CPD1;
end;else if Y=7 then do;
P1=&LC1*&CPA1*&CPB2*&CPC2*&CPD1;P2=&LC2*&CPA1*&CPB1*&CPC2*&CPD1;
P3=&LC3*&CPA2*&CPB2*&CPC2*&CPD1;P4=&LC4*&CPA2*&CPB1*&CPC2*&CPD1;
P5=&LC5*&CPA2*&CPB1*&CPC1*&CPD1;P6=&LC6*&CPA2*&CPB1*&CPC1*&CPD2;
end;else if Y=8 then do;
P1=&LC1*&CPA1*&CPB2*&CPC2*&CPD2;P2=&LC2*&CPA1*&CPB1*&CPC2*&CPD2;
P3=&LC3*&CPA2*&CPB2*&CPC2*&CPD2;P4=&LC4*&CPA2*&CPB1*&CPC2*&CPD2;
P5=&LC5*&CPA2*&CPB1*&CPC1*&CPD2;P6=&LC6*&CPA2*&CPB1*&CPC1*&CPD1;
```

```
end;else if Y=9 then do;
P1=&LC1*&CPA2*&CPB1*&CPC1*&CPD1;P2=&LC2*&CPA2*&CPB2*&CPC1*&CPD1;
P3=&LC3*&CPA1*&CPB1*&CPC1*&CPD1;P4=&LC4*&CPA1*&CPB2*&CPC1*&CPD1;
P5=&LC5*&CPA1*&CPB2*&CPC2*&CPD1;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD2;
end;else if Y=10 then do;
P1=&LC1*&CPA2*&CPB1*&CPC1*&CPD2;P2=&LC2*&CPA2*&CPB2*&CPC1*&CPD2;
P3=&LC3*&CPA1*&CPB1*&CPC1*&CPD2;P4=&LC4*&CPA1*&CPB2*&CPC1*&CPD2;
P5=&LC5*&CPA1*&CPB2*&CPC2*&CPD2;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD1;
end;else if Y=11 then do;
P1=&LC1*&CPA2*&CPB1*&CPC2*&CPD1;P2=&LC2*&CPA2*&CPB2*&CPC2*&CPD1;
P3=&LC3*&CPA1*&CPB1*&CPC2*&CPD1;P4=&LC4*&CPA1*&CPB2*&CPC2*&CPD1;
P5=&LC5*&CPA1*&CPB2*&CPC1*&CPD1;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD1;
end;else if Y=12 then do;
P1=&LC1*&CPA2*&CPB1*&CPC2*&CPD2;P2=&LC2*&CPA2*&CPB2*&CPC2*&CPD2;
P3=&LC3*&CPA1*&CPB1*&CPC2*&CPD2;P4=&LC4*&CPA1*&CPB2*&CPC2*&CPD2;
P5=&LC5*&CPA1*&CPB2*&CPC1*&CPD2;P6=&LC6*&CPA1*&CPB2*&CPC1*&CPD1;
end;else if Y=13 then do;
P1=&LC1*&CPA2*&CPB2*&CPC1*&CPD1;P2=&LC2*&CPA2*&CPB1*&CPC1*&CPD1;
P3=&LC3*&CPA1*&CPB2*&CPC1*&CPD1;P4=&LC4*&CPA1*&CPB1*&CPC1*&CPD1;
P5=&LC5*&CPA1*&CPB1*&CPC2*&CPD1;P6=&LC6*&CPA1*&CPB1*&CPC2*&CPD2;
end;else if Y=14 then do;
P1=&LC1*&CPA2*&CPB2*&CPC1*&CPD2;P2=&LC2*&CPA2*&CPB1*&CPC1*&CPD2;
P3=&LC3*&CPA1*&CPB2*&CPC1*&CPD2;P4=&LC4*&CPA1*&CPB1*&CPC1*&CPD2;
P5=&LC5*&CPA1*&CPB1*&CPC2*&CPD2;P6=&LC6*&CPA1*&CPB1*&CPC2*&CPD1;
end;else if Y=15 then do;
P1=&LC1*&CPA2*&CPB2*&CPC2*&CPD1;P2=&LC2*&CPA2*&CPB1*&CPC2*&CPD1;
P3=&LC3*&CPA1*&CPB2*&CPC2*&CPD1;P4=&LC4*&CPA1*&CPB1*&CPC2*&CPD1;
P5=&LC5*&CPA1*&CPB1*&CPC1*&CPD1;P6=&LC6*&CPA1*&CPB1*&CPC1*&CPD2;
end;else if Y=16 then do;
P1=&LC1*&CPA2*&CPB2*&CPC2*&CPD2;P2=&LC2*&CPA2*&CPB1*&CPC2*&CPD2;
P3=&LC3*&CPA1*&CPB2*&CPC2*&CPD2;P4=&LC4*&CPA1*&CPB1*&CPC2*&CPD2;
P5=&LC5*&CPA1*&CPB1*&CPC1*&CPD2;P6=&LC6*&CPA1*&CPB1*&CPC1*&CPD1;
end;TOTP1=P1+P2+P3+P4+P5+P6;
retain A1LC1 0 A1LC2 0 A1LC3 0 A1LC4 0 A1LC5 0 A1LC6 0;
     L1=(CNT1*P1)/TOTP1;L2=(CNT1*P2)/TOTP1;L3=(CNT1*P3)/TOTP1;
     L4=(CNT1*P4)/TOTP1;L5=(CNT1*P5)/TOTP1;L6=(CNT1*P6)/TOTP1;
if Y ge 9 then do;A1LC1+L1;A1LC2+L2;end;
if Y le 8 then do;A1LC3+L3;A1LC4+L4;A1LC5+L5;A1LC6+L6;end;
if Y IN (5,6,7,8,13,14,15,16)then do;B1LC1+L1;B1LC3+L3;end;
if Y IN (1,2,3,4,9,10,11,12)then do;
     B1LC2+L2;B1LC4+L4;B1LC5+L5;B1LC6+L6;end;
if Y IN (3,4,7,8,11,12,15,16)then do;
     C1LC1+L1;C1LC2+L2;C1LC3+L3;C1LC4+L4;END;
if Y IN (1,2,5,6,9,10,13,14)then do;C1LC5+L5;C1LC6+L6;END;
if Y IN (2,4,6,8,10,12,14,16)then do;
     D1LC1+L1;D1LC2+L2;D1LC3+L3;D1LC4+L4;D1LC5+L5;END;
if Y IN (1,3,5,7,9,11,13,15)then do;D1LC6+L6;END;
grpcnt1+cnt1;bigl1+l1;
bigl2+l2;bigl3+l3;bigl4+l4;bigl5+l5;bigl6+l6;
all=1;yh=y;run;
%let OLDLC12=&LC12;%let OLDLC22=&LC22;%let OLDLC32=&LC32;
%let OLDLC42=&LC42;%let OLDLC52=&LC52;%let OLDLC62=&LC62;
```

```
%let OLDCPA1_2=&CPA1_2;%let OLDCPB1_2=&CPB1_2;
%let OLDCPC1_2=&CPC1_2;%let OLDCPD1_2=&CPD1_2;
data cmg32; set CMG32(keep = y2 cnt2 notagain1 notagain2 yh all);
retain grpcnt2 0 bigl12 0 bigl22 0 bigl32 0 bigl42 0 bigl52 0
bigl62 0;
     if Y2=1 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;end;
     else if Y2=2 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
     else if Y2=3 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;end;
     else if Y2=4 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;end;
     else if Y2=5 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;end;
     else if Y2=6 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;end;
     else if Y2=7 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
```

```
P52=&LC52*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;end;
     else if Y2=8 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;end;
     else if Y2=9 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;end;
     else if Y2=10 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
     else if Y2=11 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
     else if Y2=12 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;end;
     else if Y2=13 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;end;
     else if Y2=14 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;end;
     else if Y2=15 then do;
```

```
P12=&LC12*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;end;
     else if Y2=16 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;end;
TOTP2=P12+P22+P32+P42+P52+P62;IF CNT2=. THEN CNT2=0;
retain A1LC12 0 A1LC22 0 A1LC32 0 A1LC42 0 A1LC52 0 A1LC62 0;
L12=(CNT2*P12)/TOTP2;L22=(CNT2*P22)/TOTP2;L32=(CNT2*P32)/TOTP2;
L42=(CNT2*P42)/TOTP2;L52=(CNT2*P52)/TOTP2;L62=(CNT2*P62)/TOTP2;
if Y2 ge 9 then do;A1LC12+L12;A1LC22+L22;end;
if Y2 le 8 then do;
     A1LC32+L32;A1LC42+L42;A1LC52+L52;A1LC62+L62;end;
if Y2 IN (5,6,7,8,13,14,15,16)then do;
     B1LC12+L12;B1LC32+L32;end;
if Y2 IN (1,2,3,4,9,10,11,12)then do;
     B1LC22+L22;B1LC42+L42;B1LC52+L52;B1LC62+L62;end;
if Y2 IN (3,4,7,8,11,12,15,16)then do;
     C1LC12+L12;C1LC22+L22;C1LC32+L32;C1LC42+L42;END;
if Y2 IN (1,2,5,6,9,10,13,14)then do;
     C1LC52+L52;C1LC62+L62;END;
if Y2 IN (2,4,6,8,10,12,14,16)then do;
     D1LC12+L12;D1LC22+L22;D1LC32+L32;D1LC42+L42;D1LC52+L52;END;
if Y2 IN (1,3,5,7,9,11,13,15)then do;D1LC62+L62;END;grpcnt2+cnt2;
bigl12+l12;bigl22+l22;bigl32+l32;
bigl42+l42;bigl52+l52;bigl62+l62;
all=1;yh=y2;run;proc SORT data=CMG3; BY YH; run;
proc SORT data=CMG32; BY YH; run;DATA CMG3; MERGE CMG3 CMG32;
BY YH; run;
data cm4(keep=y y2 lc1-lc6 cpa1 cpb1 cpc1 cpd1 cpa2 cpb2 cpc2
cpd2
lc12 lc22 lc32 lc42 lc52 lc62 cpa1_2 cpb1_2 cpc1_2 cpd1_2 cpa2_2
cpb2_2
cpc2_2 cpd2_2 cnt1 cnt2 grpcnt1 grpcnt2 totp1 totp2 yh);
set cmg3; by all;if last.all;
LC1=BIGL1/GRPCNT1;LC2=BIGL2/GRPCNT1;LC3=BIGL3/GRPCNT1;
LC4=BIGL4/GRPCNT1;LC5=BIGL5/GRPCNT1;LC6=BIGL6/GRPCNT1;
LCA2=A1LC1+A1LC2+A1LC3+A1LC4+A1LC5+A1LC6;
LCB2=B1LC1+B1LC2+B1LC3+B1LC4+B1LC5+B1LC6;
LCC2=C1LC1+C1LC2+C1LC3+C1LC4+C1LC5+C1LC6;
LCD2=D1LC1+D1LC2+D1LC3+D1LC4+D1LC5+D1LC6;
CPA2=LCA2/GRPCNT1;CPB2=LCB2/GRPCNT1;
CPC2=LCC2/GRPCNT1;CPD2=LCD2/GRPCNT1;
CPA1=(1-CPA2);CPB1=(1-CPB2);CPC1=(1-CPC2);CPD1=(1-CPD2);
%if &endvar=0 %then %do;
```

```
      call symput('lc1',lc1);call symput('lc2',lc2);
      call symput('lc3',lc3);call symput('lc4',lc4);
      call symput('lc5',lc5);call symput('lc6',lc6);
      call symput('cpa1',cpa1);call symput('cpb1',cpb1);
      call symput('cpc1',cpc1);call symput('cpd1',cpd1);
      call symput('cpa2',cpa2);call symput('cpb2',cpb2);
      call symput('cpc2',cpc2);call symput('cpd2',cpd2);%end;
ablc1=abs(&oldlc1-lc1);ablc2=abs(&oldlc2-lc2);
ablc3=abs(&oldlc3-lc3);ablc4=abs(&oldlc4-lc4);
ablc5=abs(&oldlc5-lc5);aba1=abs(&oldcpa1-cpa1);
abb1=abs(&oldcpb1-cpb1);abc1=abs(&oldcpc1-cpc1);
abd1=abs(&oldcpd1-cpd1);
TOTABS=ABLC1+ABLC2+ABLC3+ABLC4+ABLC5+ABA1+ABB1+ABC1+ABD1;
IF TOTABS LE .0001 AND NOTAGAIN1 NE 1 THEN DO;NOTAGAIN1=1;
call symput('lc1',lc1);call symput('lc2',lc2);
call symput('lc3',lc3);call symput('lc4',lc4);
call symput('lc5',lc5);call symput('lc6',lc6);
call symput('cpa1',cpa1);call symput('cpb1',cpb1);
call symput('cpc1',cpc1);call symput('cpd1',cpd1);
call symput('TOTP1',TOTP1);call symput('CNT1',CNT1);
call symput('GRPCNT1',GRPCNT1);END;
LC12=BIGL12/GRPCNT2;LC22=BIGL22/GRPCNT2;LC32=BIGL32/GRPCNT2;
LC42=BIGL42/GRPCNT2;LC52=BIGL52/GRPCNT2;LC62=BIGL62/GRPCNT2;
      LCA22=A1LC12+A1LC22+A1LC32+A1LC42+A1LC52+A1LC62;
      LCB22=B1LC12+B1LC22+B1LC32+B1LC42+B1LC52+B1LC62;
      LCC22=C1LC12+C1LC22+C1LC32+C1LC42+C1LC52+C1LC62;
      LCD22=D1LC12+D1LC22+D1LC32+D1LC42+D1LC52+D1LC62;
CPA2_2=LCA22/GRPCNT2;CPB2_2=LCB22/GRPCNT2;
CPC2_2=LCC22/GRPCNT2;CPD2_2=LCD22/GRPCNT2;
CPA1_2=(1-CPA2_2);CPB1_2=(1-CPB2_2);
CPC1_2=(1-CPC2_2);CPD1_2=(1-CPD2_2);
%if (&endvar=0) %then %do;
call symput('lc12',lc12);call symput('lc22',lc22);
call symput('lc32',lc32);call symput('lc42',lc42);
call symput('lc52',lc52);call symput('lc62',lc62);
call symput('cpa1_2',cpa1_2);call symput('cpb1_2',cpb1_2);
call symput('cpc1_2',cpc1_2);call symput('cpd1_2',cpd1_2);
call symput('cpa2_2',cpa2_2);call symput('cpb2_2',cpb2_2);
call symput('cpc2_2',cpc2_2);call symput('cpd2_2',cpd2_2);%end;
ablc12=abs(&oldlc12-lc12);ablc22=abs(&oldlc22-lc22);
ablc32=abs(&oldlc32-lc32);ablc42=abs(&oldlc42-lc42);
ablc52=abs(&oldlc52-lc52);ablc62=abs(&oldlc62-lc62);
aba12=abs(&oldcpa1_2-cpa1_2);abb12=abs(&oldcpb1_2-cpb1_2);
abc12=abs(&oldcpc1_2-cpc1_2);abd12=abs(&oldcpd1_2-cpd1_2);
TOTABS2=sum(ABLC12,ABLC22,ABLC32,ABLC42,ABLC52,ABA12,ABB12,ABC12,
ABD12);
IF TOTABS2 LE .0001 AND NOTAGAIN2 NE 1 THEN DO;NOTAGAIN2=1;
call symput('lc12',lc12);call symput('lc22',lc22);
call symput('lc32',lc32);call symput('lc42',lc42);
call symput('lc52',lc52);call symput('lc62',lc62);
call symput('cpa1_2',cpa1_2);call symput('cpb1_2',cpb1_2);
call symput('cpc1_2',cpc1_2);call symput('cpd1_2',cpd1_2);
```

```
call symput('TOTP2',TOTP2);call symput('CNT2',CNT2);
call symput('GRPCNT2',GRPCNT2);end;
     IF NOTAGAIN1=1 AND NOTAGAIN2=1 THEN DO;
CALL SYMPUT ('endvar',1);END;run;
%if (&endvar^=1) %then %let cnt=%eval(&cnt+1);
%if (&cnt>=500) %then %do;%let endvar=1;%end;%end;
%mend cm;%cm
data WHET1 (keep=lc1-lc5 cpa1 cpb1 cpc1 cpd1)
     WHET2 (keep=lc12 LC22 LC32 LC42 LC52 cpa1_2 cpb1_2 cpc1_2
cpd1_2);set cm4;run;
DATA HOMO (KEEP=CNT1 CNT2 YH GRPCNT1 GRPCNT2 ALL NOTAGAIN1);set
cmg3;run;


/*CHI-SQ/AIC FOR HETERO*/
data chi1(keep=all cnt1 totp1 y); set cmg3;if cnt1=.
then cnt1=0;all=1;
data chi2(keep=all grpcnt1 BIGCNT);
set cm4;BIGCNT=GRPCNT1+GRPCNT2;all=1;run;
data hetchi1;merge chi1 chi2; by all;run;
data hetchiTS1(KEEP=expx1 TCHI2_1 TG2_1 LN1 RC1 GRPCNT1 BIGCNT
CNT1 all); set hetchi1;
EXPX1=TOTP1*GRPCNT1;
CHI21=((EXPX1-CNT1)**2)/EXPX1;TCHI2_1+chi21;
if cnt1 ne 0 then expg1=log(cnt1/expx1);
else expg1=.;retain g21 tg2_1 ln1;G21=2*CNT1*EXPG1;TG2_1+g21;
PROP1=GRPCNT1/BIGCNT;TOTPP1=PROP1*TOTP1;lg1=log(totpP1);
LL1=lg1*cnt1;LN1+LL1;I=CNT1/EXPX1;
I=(I**.6667)-1;RC11=CNT1*I;RC1+RC11;RUN;
data chi1(keep=all cnt2 totp2); set cmg32;if cnt2=.
then cnt2=0;all=1;
data chi2(keep=all grpcnt2 BIGCNT);set cm4;
BIGCNT=GRPCNT1+GRPCNT2;all=1;run;
data hetchi2;merge chi1 chi2; by all;run;
data hetchiTS2(KEEP=expx2 TCHI2_2 TG2_2 LN2 RC2 GRPCNT2 BIGCNT
all cnt2);
set hetchi2;EXPX2=TOTP2*GRPCNT2;CHI22=((EXPX2-CNT2)**2)/EXPX2;
TCHI2_2+chi22;If cnt2 ne 0 then expg2=log(cnt2/expx2);
else expg2=.;retain g22 tg2_2 ln2;G22=2*CNT2*EXPG2;TG2_2+g22;
PROP2=GRPCNT2/BIGCNT;TOTPP2=PROP2*TOTP2;lg2=log(totpP2);
LL2=lg2*cnt2;LN2+LL2;I=CNT2/EXPX2;I=(I**.6667)-1;
RC22=CNT2*I;RC2+RC22;RUN;
proc sort data=hetchiTS1; by all;proc sort data=hetchiTS2;
by all;
DATA hetchi; RETAIN TCHI2_1 TCHI2_2 TG2_1 TG2_2 LN1 LN2 RC1 RC2
GRPCNT1 GRPCNT2 all;Merge hetchiTS1 hetchiTS2; BY ALL;RUN;
proc sort data=hetchi;by all;run;
DATA HETCHI(KEEP=HETCHI2 HETG2 HETLN HETRC BIGCNT); SET HETCHI;
by all;
IF LAST.ALL;RC1=1.7999999*RC1;RC2=1.7999999*RC2;
HETCHI2=TCHI2_1+TCHI2_2;HETG2=TG2_1+TG2_2;HETLN=LN1+LN2;
HETRC=RC1+RC2;RUN;
```

```
DATA HET_ABR(KEEP=HETCHI2 HETG2 HETRC HETAIC HETBIC HETRIC
CNT_HET); SET hetchi;HETAIC=(-2*HETLN)+36;
HETBIC=(-2*HETLN)+LOG(BIGCNT)*18;HETR=(BIGCNT+2)/24;
HETRIC=(-2*HETLN)+LOG(HETR)*18;CNT_HET=&CNT.;run;

%let LC1=.08;           /*PARTIAL MODEL*/
%let LC2=.03;%let LC3=.08;%let LC4=.50;%let LC5=.23;
%let LC6=.08;%let LC12=.10;
%let LC22=.05;%let LC32=.10;%let LC42=.40;%let LC52=.25;
%let LC62=.10;
%let CPA1=.95;%let CPB1=.95;%let CPC1=.95;%let CPD1=.95;
%let CPA2=.05;%let CPB2=.05;%let CPC2=.05;%let CPD2=.05;
%let CPA1_2=.95;%let CPB1_2=.95;%let CPC1_2=.95;%let CPD1_2=.95;
%let CPA2_2=.05;%let CPB2_2=.05;%let CPC2_2=.05;%let CPD2_2=.05;
%let H_LC1=.09;   /*HOMO SPECS*/
%let H_LC2=.04;%let H_LC3=.09;%let H_LC4=.45;%let H_LC5=.24;
%let H_LC6=.09;
%let cnt=0;%let endvar=0;%let j=25;
%macro cm;
%do %until (&endvar=1);
%let OLDLC1 = &LC1;%let OLDLC2 = &LC2;%let OLDLC3 = &LC3;
%let OLDLC4 = &LC4;%let OLDLC5 = &LC5;%let OLDLC6 = &LC6;
%let OLDCPA1 = &CPA1;%let OLDCPB1 = &CPB1;
%let OLDCPC1 = &CPC1;%let OLDCPD1 = &CPD1;
data pcmg3; set cmg3(keep=cnt1 y y2 yh all cnt2 notagain1
notagain2);
retain grpcnt1 0 bigl1 0 bigl2 0 bigl3 0 bigl4 0 bigl5 0 bigl6 0;
    if Y=1 then do;
P1=&LC1*&CPA1*&CPB1*&CPC1*&CPD1;P2=&LC2*&CPA1*&CPB2*&CPC1*&CPD1;
P3=&LC3*&CPA2*&CPB1*&CPC1*&CPD1;P4=&LC4*&CPA2*&CPB2*&CPC1*&CPD1;
P5=&LC5*&CPA2*&CPB2*&CPC2*&CPD1;P6=&LC6*&CPA2*&CPB2*&CPC2*&CPD2;
end;else if Y=2 then do;
P1=&LC1*&CPA1*&CPB1*&CPC1*&CPD2;P2=&LC2*&CPA1*&CPB2*&CPC1*&CPD2;
P3=&LC3*&CPA2*&CPB1*&CPC1*&CPD2;P4=&LC4*&CPA2*&CPB2*&CPC1*&CPD2;
P5=&LC5*&CPA2*&CPB2*&CPC2*&CPD2;P6=&LC6*&CPA2*&CPB2*&CPC2*&CPD1;
end;else if Y=3 then do;
P1=&LC1*&CPA1*&CPB1*&CPC2*&CPD1;P2=&LC2*&CPA1*&CPB2*&CPC2*&CPD1;
P3=&LC3*&CPA2*&CPB1*&CPC2*&CPD1;P4=&LC4*&CPA2*&CPB2*&CPC2*&CPD1;
P5=&LC5*&CPA2*&CPB2*&CPC1*&CPD1;P6=&LC6*&CPA2*&CPB2*&CPC1*&CPD2;
end;else if Y=4 then do;
P1=&LC1*&CPA1*&CPB1*&CPC2*&CPD2;P2=&LC2*&CPA1*&CPB2*&CPC2*&CPD2;
P3=&LC3*&CPA2*&CPB1*&CPC2*&CPD2;P4=&LC4*&CPA2*&CPB2*&CPC2*&CPD2;
P5=&LC5*&CPA2*&CPB2*&CPC1*&CPD2;P6=&LC6*&CPA2*&CPB2*&CPC1*&CPD1;
end;else if Y=5 then do;
P1=&LC1*&CPA1*&CPB2*&CPC1*&CPD1;P2=&LC2*&CPA1*&CPB1*&CPC1*&CPD1;
P3=&LC3*&CPA2*&CPB2*&CPC1*&CPD1;P4=&LC4*&CPA2*&CPB1*&CPC1*&CPD1;
P5=&LC5*&CPA2*&CPB1*&CPC2*&CPD1;P6=&LC6*&CPA2*&CPB1*&CPC2*&CPD2;
end;else if Y=6 then do;
P1=&LC1*&CPA1*&CPB2*&CPC1*&CPD2;P2=&LC2*&CPA1*&CPB1*&CPC1*&CPD2;
P3=&LC3*&CPA2*&CPB2*&CPC1*&CPD2;P4=&LC4*&CPA2*&CPB1*&CPC1*&CPD2;
P5=&LC5*&CPA2*&CPB1*&CPC2*&CPD2;P6=&LC6*&CPA2*&CPB1*&CPC2*&CPD1;
```

```
end;else if Y=7 then do;
P1=&LC1*&CPA1*&CPB2*&CPC2*&CPD1;P2=&LC2*&CPA1*&CPB1*&CPC2*&CPD1;
P3=&LC3*&CPA2*&CPB2*&CPC2*&CPD1;P4=&LC4*&CPA2*&CPB1*&CPC2*&CPD1;
P5=&LC5*&CPA2*&CPB1*&CPC1*&CPD1;P6=&LC6*&CPA2*&CPB1*&CPC1*&CPD2;
end;else if Y=8 then do;
P1=&LC1*&CPA1*&CPB2*&CPC2*&CPD2;P2=&LC2*&CPA1*&CPB1*&CPC2*&CPD2;
P3=&LC3*&CPA2*&CPB2*&CPC2*&CPD2;P4=&LC4*&CPA2*&CPB1*&CPC2*&CPD2;
P5=&LC5*&CPA2*&CPB1*&CPC1*&CPD2;P6=&LC6*&CPA2*&CPB1*&CPC1*&CPD1;
end;else if Y=9 then do;
P1=&LC1*&CPA2*&CPB1*&CPC1*&CPD1;P2=&LC2*&CPA2*&CPB2*&CPC1*&CPD1;
P3=&LC3*&CPA1*&CPB1*&CPC1*&CPD1;P4=&LC4*&CPA1*&CPB2*&CPC1*&CPD1;
P5=&LC5*&CPA1*&CPB2*&CPC2*&CPD1;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD2;
end;else if Y=10 then do;
P1=&LC1*&CPA2*&CPB1*&CPC1*&CPD2;P2=&LC2*&CPA2*&CPB2*&CPC1*&CPD2;
P3=&LC3*&CPA1*&CPB1*&CPC1*&CPD2;P4=&LC4*&CPA1*&CPB2*&CPC1*&CPD2;
P5=&LC5*&CPA1*&CPB2*&CPC2*&CPD2;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD1;
end;else if Y=11 then do;
P1=&LC1*&CPA2*&CPB1*&CPC2*&CPD1;P2=&LC2*&CPA2*&CPB2*&CPC2*&CPD1;
P3=&LC3*&CPA1*&CPB1*&CPC2*&CPD1;P4=&LC4*&CPA1*&CPB2*&CPC2*&CPD1;
P5=&LC5*&CPA1*&CPB2*&CPC1*&CPD1;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD1;
end;else if Y=12 then do;
P1=&LC1*&CPA2*&CPB1*&CPC2*&CPD2;P2=&LC2*&CPA2*&CPB2*&CPC2*&CPD2;
P3=&LC3*&CPA1*&CPB1*&CPC2*&CPD2;P4=&LC4*&CPA1*&CPB2*&CPC2*&CPD2;
P5=&LC5*&CPA1*&CPB2*&CPC1*&CPD2;P6=&LC6*&CPA1*&CPB2*&CPC1*&CPD1;
end;else if Y=13 then do;
P1=&LC1*&CPA2*&CPB2*&CPC1*&CPD1;P2=&LC2*&CPA2*&CPB1*&CPC1*&CPD1;
P3=&LC3*&CPA1*&CPB2*&CPC1*&CPD1;P4=&LC4*&CPA1*&CPB1*&CPC1*&CPD1;
P5=&LC5*&CPA1*&CPB1*&CPC2*&CPD1;P6=&LC6*&CPA1*&CPB1*&CPC2*&CPD2;
end;else if Y=14 then do;
P1=&LC1*&CPA2*&CPB2*&CPC1*&CPD2;P2=&LC2*&CPA2*&CPB1*&CPC1*&CPD2;
P3=&LC3*&CPA1*&CPB2*&CPC1*&CPD2;P4=&LC4*&CPA1*&CPB1*&CPC1*&CPD2;
P5=&LC5*&CPA1*&CPB1*&CPC2*&CPD2;P6=&LC6*&CPA1*&CPB1*&CPC2*&CPD1;
end;else if Y=15 then do;
P1=&LC1*&CPA2*&CPB2*&CPC2*&CPD1;P2=&LC2*&CPA2*&CPB1*&CPC2*&CPD1;
P3=&LC3*&CPA1*&CPB2*&CPC2*&CPD1;P4=&LC4*&CPA1*&CPB1*&CPC2*&CPD1;
P5=&LC5*&CPA1*&CPB1*&CPC1*&CPD1;P6=&LC6*&CPA1*&CPB1*&CPC1*&CPD2;
end;else if Y=16 then do;
P1=&LC1*&CPA2*&CPB2*&CPC2*&CPD2;P2=&LC2*&CPA2*&CPB1*&CPC2*&CPD2;
P3=&LC3*&CPA1*&CPB2*&CPC2*&CPD2;P4=&LC4*&CPA1*&CPB1*&CPC2*&CPD2;
P5=&LC5*&CPA1*&CPB1*&CPC1*&CPD2;P6=&LC6*&CPA1*&CPB1*&CPC1*&CPD1;
end;PTOTP1=P1+P2+P3+P4+P5+P6;
retain A1LC1 0 A1LC2 0 A1LC3 0 A1LC4 0 A1LC5 0 A1LC6 0;
     L1=(CNT1*P1)/PTOTP1;L2=(CNT1*P2)/PTOTP1;
     L3=(CNT1*P3)/PTOTP1;L4=(CNT1*P4)/PTOTP1;
     L5=(CNT1*P5)/PTOTP1;L6=(CNT1*P6)/PTOTP1;
if Y ge 9 then do;A1LC1+L1;A1LC2+L2;end;
if Y le 8 then do;A1LC3+L3;A1LC4+L4;A1LC5+L5;A1LC6+L6;end;
if Y IN (5,6,7,8,13,14,15,16)then do;B1LC1+L1;B1LC3+L3;end;
if Y IN (1,2,3,4,9,10,11,12)then do;
B1LC2+L2;B1LC4+L4;B1LC5+L5;B1LC6+L6;end;
if Y IN (3,4,7,8,11,12,15,16)then do;
C1LC1+L1;C1LC2+L2;C1LC3+L3;C1LC4+L4;END;
```

```
if Y IN (1,2,5,6,9,10,13,14)then do;C1LC5+L5;C1LC6+L6;END;
if Y IN (2,4,6,8,10,12,14,16)then do;
D1LC1+L1;D1LC2+L2;D1LC3+L3;D1LC4+L4;D1LC5+L5;END;
if Y IN (1,3,5,7,9,11,13,15)then do;D1LC6+L6;END;
grpcnt1+cnt1;bigl1+l1;bigl2+l2bigl3+l3;
bigl4+l4;bigl5+l5;bigl6+l6;
all=1;yh=y;run;
%let OLDLC12=&LC12;%let OLDLC22=&LC22;%let OLDLC32=&LC32;
%let OLDLC42=&LC42;%let OLDLC52=&LC52;%let OLDLC62=&LC62;
      %let OLDCPA1_2=&CPA1_2;     %let OLDCPB1_2=&CPB1_2;
      %let OLDCPC1_2=&CPC1_2;     %let OLDCPD1_2=&CPD1_2;
data Pcmg32; set CMG32(keep = y2 cnt2 notagain1 notagain2 yh
all);
retain grpcnt2 0 bigl12 0 bigl22 0 bigl32 0 bigl42 0 bigl52 0
bigl62 0;
     if Y2=1 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;end;
     else if Y2=2 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
     else if Y2=3 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;end;
     else if Y2=4 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;end;
     else if Y2=5 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;end;
     else if Y2=6 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
```

```
P22=&LC22*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;end;
     else if Y2=7 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;end;
     else if Y2=8 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;end;
     else if Y2=9 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;end;
     else if Y2=10 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
     else if Y2=11 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
     else if Y2=12 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;end;
     else if Y2=13 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
```

```
P52=&LC52*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;end;
     else if Y2=14 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;end;
     else if Y2=15 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;end;
     else if Y2=16 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;end;
PTOTP2=P12+P22+P32+P42+P52+P62;
retain A1LC12 0 A1LC22 0 A1LC32 0 A1LC42 0 A1LC52 0 A1LC62 0;
     L12=(CNT2*P12)/PTOTP2;L22=(CNT2*P22)/PTOTP2;
     L32=(CNT2*P32)/PTOTP2;L42=(CNT2*P42)/PTOTP2;
     L52=(CNT2*P52)/PTOTP2;L62=(CNT2*P62)/PTOTP2;
if Y2 ge 9 then do;A1LC12+L12;A1LC22+L22;end;
if Y2 le 8 then do;
A1LC32+L32;A1LC42+L42;A1LC52+L52;A1LC62+L62;end;
if Y2 IN (5,6,7,8,13,14,15,16)then do;B1LC12+L12;B1LC32+L32;end;
if Y2 IN (1,2,3,4,9,10,11,12)then do;
B1LC22+L22;B1LC42+L42;B1LC52+L52;B1LC62+L62;end;
if Y2 IN (3,4,7,8,11,12,15,16)then do;
C1LC12+L12;C1LC22+L22;C1LC32+L32;C1LC42+L42;END;
if Y2 IN (1,2,5,6,9,10,13,14)then do;C1LC52+L52;C1LC62+L62;END;
if Y2 IN (2,4,6,8,10,12,14,16)then do;
D1LC12+L12;D1LC22+L22;D1LC32+L32;D1LC42+L42;D1LC52+L52;END;
if Y2 IN (1,3,5,7,9,11,13,15)then do;D1LC62+L62;END;
grpcnt2+cnt2;bigl12+l12;
bigl22+l22;bigl32+l32;bigl42+l42;bigl52+l52;bigl62+l62;
all=1;yh=y2;run;proc SORT data=PCMG3; BY YH; run;proc SORT
data=PCMG32; BY YH; run;
DATA PCMG3; MERGE PCMG3 PCMG32; BY YH; run;
DATA pcmg3; set pcmg3; bigcnt=grpcnt1+grpcnt2; run;
data Pcm4(keep=y y2 lc1-lc6 cpa1 cpb1 cpc1 cpd1 cpa2 cpb2 cpc2
cpd2
lc12 lc22 lc32 lc42 lc52 lc62 cpa1_2 cpb1_2 cpc1_2 cpd1_2 cpa2_2
cpb2_2 cpc2_2 cpd2_2
cnt1 cnt2 grpcnt1 grpcnt2 bigcnt ptotp1 ptotp2 yh);
set Pcmg3; by all;if last.all;
```

```
LC1=BIGL1/GRPCNT1;LC2=BIGL2/GRPCNT1;LC3=BIGL3/GRPCNT1;
LC4=BIGL4/GRPCNT1;LC5=BIGL5/GRPCNT1;LC6=BIGL6/GRPCNT1;
      LCA2=A1LC1+A1LC2+A1LC3+A1LC4+A1LC5+A1LC6+
     A1LC12+A1LC22+A1LC32+A1LC42+A1LC52+A1LC62;
      LCB2=B1LC1+B1LC2+B1LC3+B1LC4+B1LC5+B1LC6+
     B1LC12+B1LC22+B1LC32+B1LC42+B1LC52+B1LC62;
      LCC2=C1LC1+C1LC2+C1LC3+C1LC4+C1LC5+C1LC6+
     C1LC12+C1LC22+C1LC32+C1LC42+C1LC52+C1LC62;
      LCD2=D1LC1+D1LC2+D1LC3+D1LC4+D1LC5+D1LC6+
     D1LC12+D1LC22+D1LC32+D1LC42+D1LC52+D1LC62;
CPA2=LCA2/BIGCNT;CPB2=LCB2/BIGCNT;
CPC2=LCC2/BIGCNT;CPD2=LCD2/BIGCNT;
CPA1=(1-CPA2);CPB1=(1-CPB2);
CPC1=(1-CPC2);CPD1=(1-CPD2);
%if &endvar=0 %then %do;
call symput('lc1',lc1);call symput('lc2',lc2);
call symput('lc3',lc3);call symput('lc4',lc4);
call symput('lc5',lc5);call symput('lc6',lc6);
call symput('cpa1',cpa1);call symput('cpb1',cpb1);
call symput('cpc1',cpc1);call symput('cpd1',cpd1);
call symput('cpa2',cpa2);call symput('cpb2',cpb2);
call symput('cpc2',cpc2);call symput('cpd2',cpd2);%end;
ablc1=abs(&oldlc1-lc1);ablc2=abs(&oldlc2-lc2);
ablc3=abs(&oldlc3-lc3);ablc4=abs(&oldlc4-lc4);
ablc5=abs(&oldlc5-lc5);
    aba1=abs(&oldcpa1-cpa1);abb1=abs(&oldcpb1-cpb1);
    abc1=abs(&oldcpc1-cpc1);abd1=abs(&oldcpd1-cpd1);
TOTABS=ABLC1+ABLC2+ABLC3+ABLC4+ABLC5+ABA1+ABB1+ABC1+ABD1;
IF TOTABS LE .0001 AND NOTAGAIN1 NE 1 THEN DO;NOTAGAIN1=1;
    call symput('PTOTP1',PTOTP1);call symput('CNT1',CNT1);
    call symput('GRPCNT1',GRPCNT1);END;
LC12=BIGL12/GRPCNT2;LC22=BIGL22/GRPCNT2;
LC32=BIGL32/GRPCNT2;LC42=BIGL42/GRPCNT2;
LC52=BIGL52/GRPCNT2;LC62=BIGL62/GRPCNT2;
CPA1_2=CPA1;CPB1_2=CPB1;CPC1_2=CPC1;CPD1_2=CPD1;
CPA2_2=CPA2;CPB2_2=CPB2;CPC2_2=CPC2;CPD2_2=CPD2;
%if (&endvar=0) %then %do;
    call symput('lc12',lc12);call symput('lc22',lc22);
    call symput('lc32',lc32);call symput('lc42',lc42);
    call symput('lc52',lc52);call symput('lc62',lc62);
    call symput('cpa1_2',cpa1_2);call symput('cpb1_2',cpb1_2);
    call symput('cpc1_2',cpc1_2);call symput('cpd1_2',cpd1_2);
    call symput('cpa2_2',cpa2_2);call symput('cpb2_2',cpb2_2);
    call symput('cpc2_2',cpc2_2);
    call symput('cpd2_2',cpd2_2);%end;
ablc12=abs(&oldlc12-lc12);ablc22=abs(&oldlc22-lc22);
ablc32=abs(&oldlc32-lc32);ablc42=abs(&oldlc42-lc42);
ablc52=abs(&oldlc52-lc52);ablc62=abs(&oldlc62-lc62);
aba12=abs(&oldcpa1_2-cpa1_2);abb12=abs(&oldcpb1_2-cpb1_2);
abc12=abs(&oldcpc1_2-cpc1_2);abd12=abs(&oldcpd1_2-cpd1_2);
TOTABS2=sum(ABLC12,ABLC22,ABLC32,ABLC42,ABLC52,ABA12,ABB12,ABC12,
ABD12);
```

```
IF TOTABS2 LE .0001 AND NOTAGAIN2 NE 1 THEN DO;NOTAGAIN2=1;
call symput('PTOTP2',PTOTP2);call symput('CNT2',CNT2);
call symput('GRPCNT2',GRPCNT2);end;
IF NOTAGAIN1=1 AND NOTAGAIN2=1 THEN DO;
CALL SYMPUT ('endvar',1);END;run;
%if (&endvar^=1) %then %let cnt=%eval(&cnt+1);
%if (&cnt>=500) %then %do;%let endvar=1;%end;%end;
%mend cm;%cm


/*CHI-SQ/AIC FOR PARTIAL*/
data chi1(keep=all cnt1 ptotp1 y); set pcmg3;if cnt1=.
then cnt1=0;all=1;
data chi2(keep=all grpcnt1 bigcnt); set pcm4;all=1;run;
data parchi1;merge chi1 chi2; by all;run;
data parchiTS1(KEEP=expx1 TCHI2_1 TG2_1 LN1 RC1 GRPCNT1 BIGCNT
CNT1 all); set parchi1;
EXPX1=PTOTP1*GRPCNT1;CHI21=((EXPX1-CNT1)**2)/EXPX1;
TCHI2_1+chi21;If cnt1 ne 0 then expg1=log(cnt1/expx1);
else expg1=.;
retain g21 tg2_1 ln1;
G21=2*CNT1*EXPG1;TG2_1+g21;PROP1=GRPCNT1/BIGCNT;
TOTPP1=PROP1*PTOTP1;lg1=log(totpP1);LL1=lg1*cnt1;LN1+LL1;
I=CNT1/EXPX1;I=(I**.6667)-1;RC11=CNT1*I;RC1+RC11;RUN;
data chi1(keep=all cnt2 ptotp2);set Pcmg32;if cnt2=.
then cnt2=0;all=1;
data chi2(keep=all grpcnt2 bigcnt);set Pcm4;all=1;run;
data parchi2;merge chi1 chi2; by all;run;
data parchiTS2(KEEP=TCHI2_2 TG2_2 LN2 RC2 GRPCNT2 BIGCNT all);
set parchi2;
EXPX2=PTOTP2*GRPCNT2;CHI22=((EXPX2-CNT2)**2)/EXPX2;TCHI2_2+chi22;
If cnt2 ne 0 then expg2=log(cnt2/expx2);else expg2=.;
retain g22 tg2_2 ln2;
G22=2*CNT2*EXPG2;TG2_2+g22;PROP2=GRPCNT2/BIGCNT;
TOTPP2=PROP2*PTOTP2;
lg2=log(totpP2);LL2=lg2*cnt2;LN2+LL2;I=CNT2/EXPX2;I=(I**.6667)-1;
RC22=CNT2*I;RC2+RC22;RUN;
proc sort data=parchiTS1; by all;proc sort data=parchiTS2;
by all;
DATA parchi; RETAIN TCHI2_1 TCHI2_2 TG2_1 TG2_2 LN1 LN2 RC1 RC2
GRPCNT1 GRPCNT2 all;Merge parchiTS1 parchiTS2; BY ALL;RUN;
proc sort data=parchi; by all;run;
DATA PARCHI(KEEP=PARCHI2 PARG2 PARLN PARRC BIGCNT); SET PARCHI;
by all;
IFLAST.ALL;RC1=1.7999999*RC1;RC2=1.7999999*RC2;
PARCHI2=TCHI2_1+TCHI2_2;
PARG2=TG2_1+TG2_2;PARLN=LN1+LN2;PARRC=RC1+RC2;RUN;
DATA PAR_ABR(KEEP=PARCHI2 PARG2 PARRC PARAIC PARBIC PARRIC
CNT_PAR); SET parchi;
PARAIC=(-2*PARLN)+28;
PARBIC=(-2*PARLN)+LOG(BIGCNT)*14;PARR=(BIGCNT+2)/24;
PARRIC=(-2*PARLN)+LOG(PARR)*14;CNT_PAR=&CNT.;run;
```

```
%let H_LC1=.09;   /*HOMOGENEOUS MODEL*/
%let H_LC2=.04;%let H_LC3=.09;%let H_LC4=.45;%let H_LC5=.24;
%let H_LC6=.09;
%let H_CPA1=.95;%let H_CPB1=.95;%let H_CPC1=.95;%let H_CPD1=.95;
%let H_CPA2=.05;%let H_CPB2=.05;%let H_CPC2=.05;%let H_CPD2=.05;
%let cnt=0;%let endvar=0;%let j=25;
data homo1; set homo;cnth=sum(cnt1,cnt2);IF CNTH=.
THEN CNTH=0;RUN;
%macro cm;
%do %until (&endvar=1);
%let OLDLC1 = &H_LC1;%let OLDLC2 = &H_LC2;%let OLDLC3 = &H_LC3;
%let OLDLC4 = &H_LC4;%let OLDLC5 = &H_LC5;%let OLDLC6 = &H_LC6;
%let OLDCPA1 = &H_CPA1;%let OLDCPB1 = &H_CPB1;
%let OLDCPC1 = &H_CPC1;%let OLDCPD1 = &H_CPD1;
data HOMO1; set HOMO1(keep=cntH CNT1 CNT2 GRPCNT1 GRPCNT2 yh all
notagain1);
retain bigcnth 0 bigl1 0 bigl2 0 bigl3 0 bigl4 0 bigl5 0 bigl6 0;
if YH=1 then do;
P1=&H_LC1*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD1;
P2=&H_LC2*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD1;
P3=&H_LC3*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD1;
P4=&H_LC4*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD1;
P5=&H_LC5*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD1;
P6=&H_LC6*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD2;end;
        else if YH=2 then do;
P1=&H_LC1*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD2;
P2=&H_LC2*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD2;
P3=&H_LC3*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD2;
P4=&H_LC4*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD2;
P5=&H_LC5*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD2;
P6=&H_LC6*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD1;end;
        else if YH=3 then do;
P1=&H_LC1*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD1;
P2=&H_LC2*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD1;
P3=&H_LC3*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD1;
P4=&H_LC4*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD1;
P5=&H_LC5*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD1;
P6=&H_LC6*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD2;end;
    else if YH=4 then do;
P1=&H_LC1*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD2;
P2=&H_LC2*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD2;
P3=&H_LC3*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD2;
P4=&H_LC4*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD2;
P5=&H_LC5*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD2;
P6=&H_LC6*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD1;end;
    else if YH=5 then do;
P1=&H_LC1*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD1;
P2=&H_LC2*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD1;
P3=&H_LC3*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD1;
P4=&H_LC4*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD1;
P5=&H_LC5*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD1;
P6=&H_LC6*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD2;end;
```

```
     else if YH=6 then do;
P1=&H_LC1*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD2;
P2=&H_LC2*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD2;
P3=&H_LC3*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD2;
P4=&H_LC4*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD2;
P5=&H_LC5*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD2;
P6=&H_LC6*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD1;end;
     else if YH=7 then do;
P1=&H_LC1*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD1;
P2=&H_LC2*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD1;
P3=&H_LC3*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD1;
P4=&H_LC4*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD1;
P5=&H_LC5*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD1;
P6=&H_LC6*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD2;end;
     else if YH=8 then do;
P1=&H_LC1*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD2;
P2=&H_LC2*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD2;
P3=&H_LC3*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD2;
P4=&H_LC4*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD2;
P5=&H_LC5*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD2;
P6=&H_LC6*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD1;end;
     else if YH=9 then do;
P1=&H_LC1*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD1;
P2=&H_LC2*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD1;
P3=&H_LC3*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD1;
P4=&H_LC4*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD1;
P5=&H_LC5*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD1;
P6=&H_LC6*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD2;end;
     else if YH=10 then do;
P1=&H_LC1*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD2;
P2=&H_LC2*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD2;
P3=&H_LC3*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD2;
P4=&H_LC4*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD2;
P5=&H_LC5*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD2;
P6=&H_LC6*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD1;end;
     else if YH=11 then do;
P1=&H_LC1*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD1;
P2=&H_LC2*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD1;
P3=&H_LC3*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD1;
P4=&H_LC4*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD1;
P5=&H_LC5*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD1;
P6=&H_LC6*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD1;end;
     else if YH=12 then do;
P1=&H_LC1*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD2;
P2=&H_LC2*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD2;
P3=&H_LC3*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD2;
P4=&H_LC4*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD2;
P5=&H_LC5*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD2;
P6=&H_LC6*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD1;end;
     else if YH=13 then do;
P1=&H_LC1*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD1;
P2=&H_LC2*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD1;
```

```
P3=&H_LC3*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD1;
P4=&H_LC4*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD1;
P5=&H_LC5*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD1;
P6=&H_LC6*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD2;end;
     else if YH=14 then do;
P1=&H_LC1*&H_CPA2*&H_CPB2*&H_CPC1*&H_CPD2;
P2=&H_LC2*&H_CPA2*&H_CPB1*&H_CPC1*&H_CPD2;
P3=&H_LC3*&H_CPA1*&H_CPB2*&H_CPC1*&H_CPD2;
P4=&H_LC4*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD2;
P5=&H_LC5*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD2;
P6=&H_LC6*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD1;end;
     else if YH=15 then do;
P1=&H_LC1*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD1;
P2=&H_LC2*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD1;
P3=&H_LC3*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD1;
P4=&H_LC4*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD1;
P5=&H_LC5*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD1;
P6=&H_LC6*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD2;end;
     else if YH=16 then do;
P1=&H_LC1*&H_CPA2*&H_CPB2*&H_CPC2*&H_CPD2;
P2=&H_LC2*&H_CPA2*&H_CPB1*&H_CPC2*&H_CPD2;
P3=&H_LC3*&H_CPA1*&H_CPB2*&H_CPC2*&H_CPD2;
P4=&H_LC4*&H_CPA1*&H_CPB1*&H_CPC2*&H_CPD2;
P5=&H_LC5*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD2;
P6=&H_LC6*&H_CPA1*&H_CPB1*&H_CPC1*&H_CPD1;end;
HTOTP=P1+P2+P3+P4+P5+P6;
retain A1LC1 0 A1LC2 0 A1LC3 0 A1LC4 0 A1LC5 0 A1LC6 0;
L1=(CNTH*P1)/HTOTP;L2=(CNTH*P2)/HTOTP;L3=(CNTH*P3)/HTOTP;
   L4=(CNTH*P4)/HTOTP;L5=(CNTH*P5)/HTOTP;L6=(CNTH*P6)/HTOTP;
if YH ge 9 then do;A1LC1+L1;A1LC2+L2;end;
if YH le 8 then do;A1LC3+L3;A1LC4+L4;A1LC5+L5;A1LC6+L6;end;
if YH IN (5,6,7,8,13,14,15,16)then do; B1LC1+L1;B1LC3+L3;end;
if YH IN (1,2,3,4,9,10,11,12)then do;
B1LC2+L2;B1LC4+L4;B1LC5+L5;B1LC6+L6;end;
if YH IN (3,4,7,8,11,12,15,16)then do;
C1LC1+L1;C1LC2+L2;C1LC3+L3;C1LC4+L4;END;
if YH IN (1,2,5,6,9,10,13,14)then do;C1LC5+L5;C1LC6+L6;END;
if YH IN (2,4,6,8,10,12,14,16)then do;
D1LC1+L1;D1LC2+L2;D1LC3+L3;D1LC4+L4;D1LC5+L5;END;
if YH IN (1,3,5,7,9,11,13,15)then do;D1LC6+L6;END;
bigcnth+cnth;
bigl1+l1;bigl2+l2;bigl3+l3;bigl4+l4;bigl5+l5;bigl6+l6;ALL=1;run;
proc sort data=homo1; by all;run;
data HOMO_4(keep=yh cnt1 cnt2 CNTH grpcnt1 grpcnt2 BIGCNTH Htotp
NOTAGAIN1);
set HOMO1;by all;if last.all;
H_LC1=BIGL1/BIGCNTH;H_LC2=BIGL2/BIGCNTH;H_LC3=BIGL3/BIGCNTH;
H_LC4=BIGL4/BIGCNTH;H_LC5=BIGL5/BIGCNTH;H_LC6=BIGL6/BIGCNTH;
LCA2=A1LC1+A1LC2+A1LC3+A1LC4+A1LC5+A1LC6;
LCB2=B1LC1+B1LC2+B1LC3+B1LC4+B1LC5+B1LC6;
LCC2=C1LC1+C1LC2+C1LC3+C1LC4+C1LC5+C1LC6;
LCD2=D1LC1+D1LC2+D1LC3+D1LC4+D1LC5+D1LC6;
```

```
H_CPA2=LCA2/BIGCNTH;H_CPB2=LCB2/BIGCNTH;
H_CPC2=LCC2/BIGCNTH;H_CPD2=LCD2/BIGCNTH;
H_CPA1=(1-H_CPA2);H_CPB1=(1-H_CPB2);
H_CPC1=(1-H_CPC2);H_CPD1=(1-H_CPD2);
%if &endvar=0 %then %do;
     call symput('H_lc1',H_lc1);call symput('H_lc2',H_lc2);
     call symput('H_lc3',H_lc3);call symput('H_lc4',H_lc4);
     call symput('H_lc5',H_lc5);call symput('H_lc6',H_lc6);
     call symput('H_cpa1',H_cpa1);call symput('H_cpb1',H_cpb1);
     call symput('H_cpc1',H_cpc1);call symput('H_cpd1',H_cpd1);
     call symput('H_cpa2',H_cpa2);call symput('H_cpb2',H_cpb2);
     call symput('H_cpc2',H_cpc2);
     call symput('H_cpd2',H_cpd2);%end;
   ablc1=abs(&oldlc1-H_lc1);ablc2=abs(&oldlc2-H_lc2);
   ablc3=abs(&oldlc3-H_lc3);ablc4=abs(&oldlc4-H_lc4);
   ablc5=abs(&oldlc5-H_lc5);
   aba1=abs(&oldcpa1-H_cpa1);abb1=abs(&oldcpb1-H_cpb1);
   abc1=abs(&oldcpc1-H_cpc1);abd1=abs(&oldcpd1-H_cpd1);
TOTABS=ABLC1+ABLC2+ABLC3+ABLC4+ABLC5+ABA1+ABB1+ABC1+ABD1;
IF TOTABS LE .0001 THEN DO;NOTAGAIN1=1;
     call symput('H_lc1',H_lc1);call symput('H_lc2',H_lc2);
      call symput('H_lc3',H_lc3);call symput('H_lc4',H_lc4);
      call symput('H_lc5',H_lc5);call symput('H_lc6',H_lc6);
      call symput('H_cpa1',H_cpa1);call symput('H_cpb1',H_cpb1);
      call symput('H_cpc1',H_cpc1);call symput('H_cpd1',H_cpd1);
      call symput('HTOTP',HTOTP);end;IF NOTAGAIN1=1 THEN DO;
     CALL SYMPUT ('endvar',1);END;run;
%if (&endvar^=1) %then %let cnt=%eval(&cnt+1);
%if (&cnt>=500) %then %do;%let endvar=1;%end;%end;
%mend cm;%cm

/*CHI-SQ/AIC FOR HOMO*/
data chi1(keep=all cnt1 Htotp YH); set HOMO1;if cnt1=.
then cnt1=0;all=1;
data chi2(keep=all grpcnt1 BIGCNTH); set HOMO_4;all=1;run;
data HOMchi1;merge chi1 chi2; by all;run;
data HOMchiTS1(KEEP=TCHI2_1 TG2_1 LN1 RC1 GRPCNT1 BIGCNTH CNT1
all); set HOMchi1;
EXPX1=HTOTP*GRPCNT1;CHI21=((EXPX1-CNT1)**2)/EXPX1;TCHI2_1+chi21;
If cnt1 ne 0 then expg1=log(cnt1/expx1);else expg1=.;
retain g21 tg2_1 ln1;
G21=2*CNT1*EXPG1;TG2_1+g21;PROP1=GRPCNT1/BIGCNTH;
TOTPP1=PROP1*HTOTP;
lg1=log(totpP1);LL1=lg1*cnt1;LN1+LL1;I=CNT1/EXPX1;I=(I**.6667)-1;
RC11=CNT1*I;RC1+RC11;RUN;
data chi1(keep=all cnt2 Htotp); set HOMO1;if cnt2=.
then cnt2=0;all=1;
data chi2(keep=all grpcnt2 BIGCNTH); set HOMO_4;all=1;run;
data HOMchi2;merge chi1 chi2; by all;run;
data HOMchiTS2(KEEP=TCHI2_2 TG2_2 LN2 RC2 GRPCNT2 BIGCNTH all
cnt2); set HOMchi2;
EXPX2=HTOTP*GRPCNT2;CHI22=((EXPX2-CNT2)**2)/EXPX2;TCHI2_2+chi22;
```

```
If cnt2 ne 0 then expg2=log(cnt2/expx2);else expg2=.;
retain g22 tg2_2 ln2;
G22=2*CNT2*EXPG2;TG2_2+g22;PROP2=GRPCNT2/BIGCNTH;
TOTPP2=PROP2*HTOTP;
lg2=log(totpP2);LL2=lg2*cnt2;LN2+LL2;I=CNT2/EXPX2;I=(I**.6667)-1;
RC22=CNT2*I;RC2+RC22;RUN;
proc sort data=HOMchiTS1; by all;
proc sort data=HOMchiTS2; by all;
DATA HOMchi; RETAIN TCHI2_1 TCHI2_2 TG2_1 TG2_2 LN1 LN2 RC1 RC2
GRPCNT1 GRPCNT2 all;Merge HOMchiTS1 HOMchiTS2; BY ALL;RUN;
proc sort data=HOMchi; by all;run;
DATA HOMCHI(KEEP=HOMCHI2 HOMG2 HOMLN HOMRC BIGCNTH); SET HOMCHI;
by all;
IF LAST.ALL;RC1=1.7999999*RC1;RC2=1.7999999*RC2;
HOMCHI2=TCHI2_1+TCHI2_2;
HOMG2=TG2_1+TG2_2;HOMLN=LN1+LN2;HOMRC=RC1+RC2;RUN;
DATA HOM_ABR(KEEP=HOMCHI2 HOMG2 HOMRC HOMAIC HOMBIC HOMRIC
CNT_HOM); SET HOMchi;
HOMAIC=(-2*HOMLN)+18;HOMBIC=(-2*HOMLN)+LOG(BIGCNTH)*9;
HOMR=(BIGCNTH+2)/24;
HOMRIC=(-2*HOMLN)+LOG(HOMR)*9;CNT_HOM=&CNT.;run;


/*COUNTING*/
data het_abr;set het_abr;dummy=1;data hom_abr;
set hom_abr;dummy=1;
data par_abr;set par_abr;dummy=1;
Data countnew(DROP= HETCHI2 PARCHI2 HOMCHI2 HETRC PARRC HOMRC
PRC1 PRC2 PARAIC
HETAIC HOMAIC PARBIC HETBIC HOMBIC PARRIC HETRIC HOMRIC HETG2
PARG2 HOMG2 CNT_HET
CNT_PAR CNT_HOM PX1 PX2 PG1 PG2);
        merge het_abr par_abr hom_abr;by dummy;drop dummy;
           IF HETCHI2>21.03 THEN HETFIT05=1;ELSE HETFIT05=0;
           IF PARCHI2>26.30 THEN PARFIT05=1;ELSE PARFIT05=0;
           IF HOMCHI2>32.67 THEN HOMFIT05=1;ELSE HOMFIT05=0;
           IF HETCHI2>26.22 THEN HETFIT01=1;ELSE HETFIT01=0;
           IF PARCHI2>32.00 THEN PARFIT01=1;ELSE PARFIT01=0;
           IF HOMCHI2>38.93 THEN HOMFIT01=1;ELSE HOMFIT01=0;
           IF CNT_HET>=500 THEN HET_NOT=1;ELSE HET_NOT=0;
           IF CNT_PAR>=500 THEN PAR_NOT=1;ELSE PAR_NOT=0;
           IF CNT_HOM>=500 THEN HOM_NOT=1;ELSE HOM_NOT=0;
           PX1=abs(HOMCHI2-PARCHI2);PX2=abs(PARCHI2-HETCHI2);
           IF PX1<11.0705 & PX2<9.4877 THEN CNT_HOMX205=1;
Else cnt_HOMX205=0;
           IF PX1>11.0705 & PX2<9.4877 THEN CNT_PX205=1;
Else cnt_PX205=0;
           IF PX1>11.0705 & PX2>9.4877 THEN CNT_HETX205=1;
Else cnt_HETX205=0;
           IF PX1<11.0705 & PX2>9.4877 THEN CNT_HET2_X205=1;
Else cnt_HET2_X205=0;
PG1=abs(HOMG2-PARG2);PG2=abs(PARG2-HETG2);
```

```
             IF PG1<11.0705 & PG2<9.4877 THEN
CNT_HOMG205=1;Elsecnt_HOMG205=0;
             IF PG1>11.0705 & PG2<9.4877 THEN
CNT_PG205=1;Elsecnt_PG205=0;
             IF PG1>11.0705 & PG2>9.4877 THEN
CNT_HETG205=1;Elsecnt_HETG205=0;
             IF PG1<11.0705 & PG2>9.4877 THEN
CNT_HET2_G205=1;Elsecnt_HET2_G205=0;
             PRC1=abs(HOMRC-PARRC);PRC2=abs(PARRC-HETRC);
             IF PRC1<11.0705 & PRC2<9.4877 THEN
CNT_HOMRC05=1;Elsecnt_HOMRC05=0;
             IF PRC1>11.0705 & PRC2<9.4877 THEN
CNT_PRC05=1;Elsecnt_PRC05=0;
             IF PRC1>11.0705 & PRC2>9.4877 THEN
CNT_HETRC05=1;Elsecnt_HETRC05=0;
             IF PRC1<11.0705 & PRC2>9.4877 THEN
CNT_HET2_RC05=1;Elsecnt_HET2_RC05=0;
             PX1=abs(HOMCHI2-PARCHI2);PX2=abs(PARCHI2-HETCHI2);
             IF PX1<15.09 & PX2<13.28 THEN CNT_HOMX201=1;
Else cnt_HOMX201=0;
             IF PX1>15.09 & PX2<13.28 THEN CNT_PX201=1;
Else cnt_PX201=0;
             IF PX1>15.09 & PX2>13.28 THEN CNT_HETX201=1;
Else cnt_HETX201=0;
             IF PX1<15.09 & PX2>13.28 THEN CNT_HET2_X201=1;
Else cnt_HET2_X201=0;
         PG1=abs(HOMG2-PARG2);PG2=abs(PARG2-HETG2);
             IF PG1<15.09 & PG2<13.28 THEN CNT_HOMG201=1;
Else cnt_HOMG201=0;
             IF PG1>15.09 & PG2<13.28 THEN CNT_PG201=1;
Else cnt_PG201=0;
             IF PG1>15.09 & PG2>13.28 THEN CNT_HETG201=1;
Else cnt_HETG201=0;
             IF PG1<15.09 & PG2>13.28 THEN CNT_HET2_G201=1;
Else cnt_HET2_G201=0;
             PRC1=abs(HOMRC-PARRC);PRC2=abs(PARRC-HETRC);
         IF PRC1<15.09 & PRC2<13.28 THEN CNT_HOMRC01=1;
Else cnt_HOMRC01=0;
         IF PRC1>15.09 & PRC2<13.28 THEN CNT_PRC01=1;
Else cnt_PRC01=0;
         IF PRC1>15.09 & PRC2>13.28 THEN CNT_HETRC01=1;
Else cnt_HETRC01=0;
         IF PRC1<15.09 & PRC2>13.28 THEN CNT_HET2_RC01=1;
Else cnt_HET2_RC01=0;
          IF PARAIC<HETAIC & PARAIC<HOMAIC then PAR_AIC=1;
Else PAR_AIC=0;
        IF PARBIC<HETBIC & PARBIC<HOMBIC then PAR_BIC=1;
Else PAR_bic=0;
          IF PARRIC<HETRIC & PARRIC<HOMRIC then PAR_RIC=1;
Else PAR_ric=0;
        IF HOMAIC<PARAIC & HOMAIC<HETAIC then HOM_AIC=1;
Else HOM_AIC=0;
```

```
        IF HOMBIC<PARBIC & HOMAIC<HETBIC then HOM_BIC=1;
Else HOM_BIC=0;
        IF HOMRIC<PARRIC & HOMAIC<HETRIC then HOM_RIC=1;
Else HOM_RIC=0;
         IF HETAIC<PARAIC & HETAIC<HOMAIC then HET_AIC=1;
Else HET_AIC=0;
         IF HETBIC<PARBIC & HETBIC<HOMBIC then HET_BIC=1;
Else HET_bic=0;
         IF HETRIC<PARRIC & HETRIC<HOMRIC then HET_RIC=1;
Else HET_ric=0;run;%if &k=1 %then
      %do;data count;set countnew;%end;%else
%do;data count;  set count countnew;%end;%END;
%MEND FIRST;%FIRST;
proc means data=count noprint nway;
output out=savelib.nhsda sum=;run;
DATA _NULL_;CALL SYMPUT('END',PUT(TIME(),TIME8.));RUN;%put &END;
```

APPENDIX B-2 WALD SAS PROGRAM

```
LIBNAME SAVELIB "C:\FINAL";
DATA _NULL_;CALL
SYMPUT('START',PUT(TIME(),TIME8.));RUN;
%put &START;%let cp=.75;
data blank16;input y;cards;
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16;run;
data blank16_2;input y2;cards;
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16;run;
%macro JACK;%do k=1 %to 1000;options nosymbolgen nomprint
nomlogic noNOTES;
/*CREATE DATA*/
data cmg1;
    do i = 1 to 20;        /*create Dirichlet values--# OF
CLUSTERS*/
gamma1=rangam(0,8);gamma2=rangam(0,3);gamma3=rangam(0,8);
gamma4=rangam(0,50);gamma5=rangam(0,23);gamma6=rangam(0,8);
gamma7=rangam(0,10);gamma8=rangam(0,5);gamma9=rangam(0,10);
gamma10=rangam(0,40);gamma11=rangam(0,25);gamma12=rangam(0,10);
total1=sum(of gamma1-gamma6);
         d1=gamma1/total1;d2=gamma2/total1;d3=gamma3/total1;
         d4=gamma4/total1;d5=gamma5/total1;d6=gamma6/total1;
total2=sum(of gamma7-gamma12);
         d7=gamma7/total2;d8=gamma8/total2;d9=gamma9/total2;
      d10=gamma10/total2;d11=gamma11/total2;d12=gamma12/total2;
              do j=1 to 25;    /* create clusters-- CLUSTER
SIZE*/
              x=ranuni(0);
         if x le D1 then c=1;else if D1 lt x le D1+D2 then c=2;
         else if D1+D2 lt x le D1+D2+D3 then c=3;
         else if D1+D2+D3 lt x le D1+D2+D3+D4 then c=4;
         else if (D1+D2+D3+D4) lt x le D1+D2+D3+D4+D5 then c=5;
         else if x gt D1+D2+D3+D4+D5 then c=6;
              x6=ranuni(0);
         if x6 le D7 then c2=1;else if D7 lt x6 le D7+D8 then
c2=2;
         else if D7+D8 lt x6 le D7+D8+D9 then c2=3;
         else if D7+D8+D9 lt x6 le D7+D8+D9+D10 then c2=4;
         else if (D7+D8+D9+D10) lt x6 le D7+D8+D9+D10+D11 then
c2=5;              else if x6 gt D7+D8+D9+D10+D11 then
c2=6;output;end;end;
data cmg1(keep=V1 V2 V3 V4 V12 V22 V32 V42);set cmg1(keep=c c2);
x2=ranuni(0);x3=ranuni(0);x4=ranuni(0);x5=ranuni(0);cp=&cp;
if c=1 then do;
if x2 le cp then v1=0; else v1=1;if x3 le cp then v2=0; else
v2=1;
if x4 le cp then v3=0; else v3=1;if x5 le cp then v4=0;
else v4=1;end;if c=2 then do;
if x2 le cp then v1=0; else v1=1;if x3 le cp then v2=1; else
v2=0;
```

140

```
if x4 le cp then v3=0; else v3=1;if x5 le cp then v4=0;
else v4=1;end;if c=3 then do;
if x2 le cp then v1=1; else v1=0;if x3 le cp then v2=0; else
v2=1;
if x4 le cp then v3=0; else v3=1;if x5 le cp then v4=0;
else v4=1;end;if c=4 then do;
if x2 le cp then v1=1; else v1=0;if x3 le cp then v2=1; else
v2=0;
if x4 le cp then v3=0; else v3=1;if x5 le cp then v4=0;
else v4=1;end;
if c=5 then do;
if x2 le cp then v1=1; else v1=0;if x3 le cp then v2=1; else
v2=0;
if x4 le cp then v3=1; else v3=0;if x5 le cp then v4=0;
else v4=1;end;if c=6 then do;
if x2 le cp then v1=1; else v1=0;if x3 le cp then v2=1; else
v2=0;
if x4 le cp then v3=1; else v3=0;if x5 le cp then v4=1;
else v4=0;end;
x7=ranuni(0);x8=ranuni(0);x9=ranuni(0);x10=ranuni(0);
if c2=1 then do;
if x7 le cp then v12=0; else v12=1;if x8 le cp then v22=0;
else v22=1;
if x9 le cp then v32=0; else v32=1;if x10 le cp then v42=0;
else v42=1;end;if c2=2 then do;
if x7 le cp then v12=0; else v12=1;if x8 le cp then v22=1;
else v22=0;
if x9 le cp then v32=0; else v32=1;if x10 le cp then v42=0;
else v42=1;end;if c2=3 then do;
if x7 le cp then v12=1; else v12=0;if x8 le cp then v22=0;
else v22=1;
if x9 le cp then v32=0; else v32=1;if x10 le cp then v42=0;
else v42=1;end;if c2=4 then do;
if x7 le cp then v12=1; else v12=0;if x8 le cp then v22=1;
else v22=0;
if x9 le cp then v32=0; else v32=1;if x10 le cp then v42=0;
else v42=1;end;
if c2=5 then do;if x7 le cp then v12=1; else v12=0;
if x8 le cp then v22=1; else v22=0;
if x9 le cp then v32=1; else v32=0;if x10 le cp then v42=0;
else v42=1;end;if c2=6 then do;
if x7 le cp then v12=1; else v12=0;if x8 le cp then v22=1;
else v22=0;
if x9 le cp then v32=1; else v32=0;if x10 le cp then v42=1;
else v42=0;end;run;
%let LC1=.08;
%let LC2=.03;%let LC3=.08;%let LC4=.50;%let LC5=.23;%let LC6=.08;
%let LC12=.10;%let LC22=.05;%let LC32=.10;%let LC42=.40;
%let LC52=.25;%let LC62=.10;
%let CPA1=.75;%let CPB1=.75;%let CPC1=.75;%let CPD1=.75;
%let CPA2=.25;%let CPB2=.25;%let CPC2=.25;%let CPD2=.25;
%let CPA1_2=.75;%let CPB1_2=.75;%let CPC1_2=.75;%let CPD1_2=.75;
```

```
%let CPA2_2=.25;%let CPB2_2=.25;%let CPC2_2=.25;%let CPD2_2=.25;
data cmg1(keep=y y2); set cmg1;
%let cnt=0;%let endvar=0;%let j=25;
IF V1=0 AND V2=0 AND V3=0 AND V4=0 THEN Y=1; ELSE
IF V1=0 AND V2=0 AND V3=0 AND V4=1 THEN Y=2; ELSE
IF V1=0 AND V2=0 AND V3=1 AND V4=0 THEN y=3; ELSE
IF V1=0 AND V2=0 AND V3=1 AND V4=1 THEN y=4; ELSE
IF V1=0 AND V2=1 AND V3=0 AND V4=0 THEN y=5; ELSE
IF V1=0 AND V2=1 AND V3=0 AND V4=1 THEN y=6; ELSE
IF V1=0 AND V2=1 AND V3=1 AND V4=0 THEN y=7; ELSE
IF V1=0 AND V2=1 AND V3=1 AND V4=1 THEN y=8; ELSE
IF V1=1 AND V2=0 AND V3=0 AND V4=0 THEN y=9; ELSE
IF V1=1 AND V2=0 AND V3=0 AND V4=1 THEN y=10; ELSE
IF V1=1 AND V2=0 AND V3=1 AND V4=0 THEN y=11; ELSE
IF V1=1 AND V2=0 AND V3=1 AND V4=1 THEN y=12; ELSE
IF V1=1 AND V2=1 AND V3=0 AND V4=0 THEN y=13; ELSE
IF V1=1 AND V2=1 AND V3=0 AND V4=1 THEN y=14; ELSE
IF V1=1 AND V2=1 AND V3=1 AND V4=0 THEN y=15; ELSE
IF V1=1 AND V2=1 AND V3=1 AND V4=1 THEN y=16;
IF V12=0 AND V22=0 AND V32=0 AND V42=0 THEN Y2=1; ELSE
IF V12=0 AND V22=0 AND V32=0 AND V42=1 THEN Y2=2; ELSE
IF V12=0 AND V22=0 AND V32=1 AND V42=0 THEN Y2=3; ELSE
IF V12=0 AND V22=0 AND V32=1 AND V42=1 THEN Y2=4; ELSE
IF V12=0 AND V22=1 AND V32=0 AND V42=0 THEN Y2=5; ELSE
IF V12=0 AND V22=1 AND V32=0 AND V42=1 THEN Y2=6; ELSE
IF V12=0 AND V22=1 AND V32=1 AND V42=0 THEN Y2=7; ELSE
IF V12=0 AND V22=1 AND V32=1 AND V42=1 THEN Y2=8; ELSE
IF V12=1 AND V22=0 AND V32=0 AND V42=0 THEN Y2=9; ELSE
IF V12=1 AND V22=0 AND V32=0 AND V42=1 THEN Y2=10; ELSE
IF V12=1 AND V22=0 AND V32=1 AND V42=0 THEN Y2=11; ELSE
IF V12=1 AND V22=0 AND V32=1 AND V42=1 THEN Y2=12; ELSE
IF V12=1 AND V22=1 AND V32=0 AND V42=0 THEN Y2=13; ELSE
IF V12=1 AND V22=1 AND V32=0 AND V42=1 THEN Y2=14; ELSE
IF V12=1 AND V22=1 AND V32=1 AND V42=0 THEN Y2=15; ELSE
IF V12=1 AND V22=1 AND V32=1 AND V42=1 THEN Y2=16;RUN;
data cmg2;set cmg1;notagain1=0;notagain2=0;
data blank16;set blank16;notagain1=0;notagain2=0;
data blank16_2;set blank16_2;notagain1=0;notagain2=0;run;
PROC FREQ NOPRINT data=cmg2; TABLES Y /
OUT=YFREQ(DROP=PERCENT);RUN;
PROC SORT DATA=CMG2; BY Y; run;
DATA CMG3;MERGE YFREQ(RENAME=(COUNT=CNT1)) CMG2;BY Y;if
first.Y;run;
data cmg3;MERGE CMG3 blank16; BY Y;RUN;
data cmg3 (drop=y2);SET CMG3;if cnt1=. then cnt1=0;RUN;
PROC SORT DATA=CMG2; BY Y2; run;PROC FREQ  NOPRINT data=cmg2;
TABLES Y2 / OUT=Y2FREQ(DROP=PERCENT);RUN;
DATA CMG32(keep=y2 cnt2);MERGE Y2FREQ(RENAME=(COUNT=cnt2)) CMG2;
BY Y2;if first.Y2;run;
proc sort data=cmg32; by y2;proc sort data=blank16_2; by y2; run;
data cmg32;MERGE CMG32 blank16_2; BY Y2;RUN;
```

```
DATA CMG3; SET CMG3;YH=Y; all=1;DATA CMG32; SET
CMG32;YH=Y2;all=1;
proc SORT data=CMG3; BY YH;proc SORT data=CMG32; BY YH;run;
DATA CMG3; MERGE CMG3 CMG32; BY YH;run;
%macro cm;
%do %until (&endvar=1);
%let OLDLC1 = &LC1;%let OLDLC2 = &LC2;%let OLDLC3 = &LC3;
%let OLDLC4 = &LC4;%let OLDLC5 = &LC5;%let OLDLC6 = &LC6;
%let OLDCPA1 = &CPA1;%let OLDCPB1 = &CPB1;
%let OLDCPC1 = &CPC1;%let OLDCPD1 = &CPD1;
data cmg3; set cmg3(keep=cnt1 y y2 yh all cnt2 notagain1
notagain2);
retain grpcnt1 0 bigl1 0 bigl2 0 bigl3 0 bigl4 0 bigl5 0 bigl6 0;
    if Y=1 then do;
P1=&LC1*&CPA1*&CPB1*&CPC1*&CPD1;P2=&LC2*&CPA1*&CPB2*&CPC1*&CPD1;
P3=&LC3*&CPA2*&CPB1*&CPC1*&CPD1;P4=&LC4*&CPA2*&CPB2*&CPC1*&CPD1;
P5=&LC5*&CPA2*&CPB2*&CPC2*&CPD1;P6=&LC6*&CPA2*&CPB2*&CPC2*&CPD2;e
nd;
    else if Y=2 then do;
P1=&LC1*&CPA1*&CPB1*&CPC1*&CPD2;P2=&LC2*&CPA1*&CPB2*&CPC1*&CPD2;
P3=&LC3*&CPA2*&CPB1*&CPC1*&CPD2;P4=&LC4*&CPA2*&CPB2*&CPC1*&CPD2;
P5=&LC5*&CPA2*&CPB2*&CPC2*&CPD2;P6=&LC6*&CPA2*&CPB2*&CPC2*&CPD1;e
nd;
    else if Y=3 then do;
P1=&LC1*&CPA1*&CPB1*&CPC2*&CPD1;P2=&LC2*&CPA1*&CPB2*&CPC2*&CPD1;
P3=&LC3*&CPA2*&CPB1*&CPC2*&CPD1;P4=&LC4*&CPA2*&CPB2*&CPC2*&CPD1;
P5=&LC5*&CPA2*&CPB2*&CPC1*&CPD1;P6=&LC6*&CPA2*&CPB2*&CPC1*&CPD2;e
nd;
    else if Y=4 then do;
P1=&LC1*&CPA1*&CPB1*&CPC2*&CPD2;P2=&LC2*&CPA1*&CPB2*&CPC2*&CPD2;
P3=&LC3*&CPA2*&CPB1*&CPC2*&CPD2;P4=&LC4*&CPA2*&CPB2*&CPC2*&CPD2;
P5=&LC5*&CPA2*&CPB2*&CPC1*&CPD2;P6=&LC6*&CPA2*&CPB2*&CPC1*&CPD1;e
nd;
    else if Y=5 then do;
P1=&LC1*&CPA1*&CPB2*&CPC1*&CPD1;P2=&LC2*&CPA1*&CPB1*&CPC1*&CPD1;
P3=&LC3*&CPA2*&CPB2*&CPC1*&CPD1;P4=&LC4*&CPA2*&CPB1*&CPC1*&CPD1;
P5=&LC5*&CPA2*&CPB1*&CPC2*&CPD1;P6=&LC6*&CPA2*&CPB1*&CPC2*&CPD2;e
nd;
    else if Y=6 then do;
P1=&LC1*&CPA1*&CPB2*&CPC1*&CPD2;P2=&LC2*&CPA1*&CPB1*&CPC1*&CPD2;
P3=&LC3*&CPA2*&CPB2*&CPC1*&CPD2;P4=&LC4*&CPA2*&CPB1*&CPC1*&CPD2;
P5=&LC5*&CPA2*&CPB1*&CPC2*&CPD2;P6=&LC6*&CPA2*&CPB1*&CPC2*&CPD1;e
nd;
    else if Y=7 then do;
P1=&LC1*&CPA1*&CPB2*&CPC2*&CPD1;P2=&LC2*&CPA1*&CPB1*&CPC2*&CPD1;
P3=&LC3*&CPA2*&CPB2*&CPC2*&CPD1;P4=&LC4*&CPA2*&CPB1*&CPC2*&CPD1;
P5=&LC5*&CPA2*&CPB1*&CPC1*&CPD1;P6=&LC6*&CPA2*&CPB1*&CPC1*&CPD2;e
nd;
    else if Y=8 then do;
P1=&LC1*&CPA1*&CPB2*&CPC2*&CPD2;P2=&LC2*&CPA1*&CPB1*&CPC2*&CPD2;
P3=&LC3*&CPA2*&CPB2*&CPC2*&CPD2;P4=&LC4*&CPA2*&CPB1*&CPC2*&CPD2;
```

```
P5=&LC5*&CPA2*&CPB1*&CPC1*&CPD2;P6=&LC6*&CPA2*&CPB1*&CPC1*&CPD1;e
nd;
     else if Y=9 then do;
P1=&LC1*&CPA2*&CPB1*&CPC1*&CPD1;P2=&LC2*&CPA2*&CPB2*&CPC1*&CPD1;
P3=&LC3*&CPA1*&CPB1*&CPC1*&CPD1;P4=&LC4*&CPA1*&CPB2*&CPC1*&CPD1;
P5=&LC5*&CPA1*&CPB2*&CPC2*&CPD1;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD2;e
nd;
     else if Y=10 then do;
P1=&LC1*&CPA2*&CPB1*&CPC1*&CPD2;P2=&LC2*&CPA2*&CPB2*&CPC1*&CPD2;
P3=&LC3*&CPA1*&CPB1*&CPC1*&CPD2;P4=&LC4*&CPA1*&CPB2*&CPC1*&CPD2;
P5=&LC5*&CPA1*&CPB2*&CPC2*&CPD2;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD1;e
nd;
     else if Y=11 then do;
P1=&LC1*&CPA2*&CPB1*&CPC2*&CPD1;P2=&LC2*&CPA2*&CPB2*&CPC2*&CPD1;
P3=&LC3*&CPA1*&CPB1*&CPC2*&CPD1;P4=&LC4*&CPA1*&CPB2*&CPC2*&CPD1;
P5=&LC5*&CPA1*&CPB2*&CPC1*&CPD1;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD1;e
nd;
     else if Y=12 then do;
P1=&LC1*&CPA2*&CPB1*&CPC2*&CPD2;P2=&LC2*&CPA2*&CPB2*&CPC2*&CPD2;
P3=&LC3*&CPA1*&CPB1*&CPC2*&CPD2;P4=&LC4*&CPA1*&CPB2*&CPC2*&CPD2;
P5=&LC5*&CPA1*&CPB2*&CPC1*&CPD2;P6=&LC6*&CPA1*&CPB2*&CPC1*&CPD1;e
nd;
     else if Y=13 then do;
P1=&LC1*&CPA2*&CPB2*&CPC1*&CPD1;P2=&LC2*&CPA2*&CPB1*&CPC1*&CPD1;
P3=&LC3*&CPA1*&CPB2*&CPC1*&CPD1;P4=&LC4*&CPA1*&CPB1*&CPC1*&CPD1;
P5=&LC5*&CPA1*&CPB1*&CPC2*&CPD1;P6=&LC6*&CPA1*&CPB1*&CPC2*&CPD2;e
nd;
     else if Y=14 then do;
P1=&LC1*&CPA2*&CPB2*&CPC1*&CPD2;P2=&LC2*&CPA2*&CPB1*&CPC1*&CPD2;
P3=&LC3*&CPA1*&CPB2*&CPC1*&CPD2;P4=&LC4*&CPA1*&CPB1*&CPC1*&CPD2;
P5=&LC5*&CPA1*&CPB1*&CPC2*&CPD2;P6=&LC6*&CPA1*&CPB1*&CPC2*&CPD1;e
nd;
     else if Y=15 then do;
P1=&LC1*&CPA2*&CPB2*&CPC2*&CPD1;P2=&LC2*&CPA2*&CPB1*&CPC2*&CPD1;
P3=&LC3*&CPA1*&CPB2*&CPC2*&CPD1;P4=&LC4*&CPA1*&CPB1*&CPC2*&CPD1;
P5=&LC5*&CPA1*&CPB1*&CPC1*&CPD1;P6=&LC6*&CPA1*&CPB1*&CPC1*&CPD2;e
nd;
     else if Y=16 then do;
P1=&LC1*&CPA2*&CPB2*&CPC2*&CPD2;P2=&LC2*&CPA2*&CPB1*&CPC2*&CPD2;
P3=&LC3*&CPA1*&CPB2*&CPC2*&CPD2;P4=&LC4*&CPA1*&CPB1*&CPC2*&CPD2;
P5=&LC5*&CPA1*&CPB1*&CPC1*&CPD2;P6=&LC6*&CPA1*&CPB1*&CPC1*&CPD1;e
nd;
TOTP1=P1+P2+P3+P4+P5+P6;
retain A1LC1 0 A1LC2 0 A1LC3 0 A1LC4 0 A1LC5 0 A1LC6 0;
L1=(CNT1*P1)/TOTP1;L2=(CNT1*P2)/TOTP1;L3=(CNT1*P3)/TOTP1;
L4=(CNT1*P4)/TOTP1;L5=(CNT1*P5)/TOTP1;L6=(CNT1*P6)/TOTP1;
if Y ge 9 then do;A1LC1+L1;A1LC2+L2;end;
if Y le 8 then do;A1LC3+L3;A1LC4+L4;A1LC5+L5;A1LC6+L6;end;
if Y IN (5,6,7,8,13,14,15,16)then do;B1LC1+L1;B1LC3+L3;end;
if Y IN (1,2,3,4,9,10,11,12)then
do;B1LC2+L2;B1LC4+L4;B1LC5+L5;B1LC6+L6;end;
```

```
if Y IN (3,4,7,8,11,12,15,16)then
do;C1LC1+L1;C1LC2+L2;C1LC3+L3;C1LC4+L4;END;
if Y IN (1,2,5,6,9,10,13,14)then do;C1LC5+L5;C1LC6+L6;END;
if Y IN (2,4,6,8,10,12,14,16)then
do;D1LC1+L1;D1LC2+L2;D1LC3+L3;D1LC4+L4;D1LC5+L5;END;
if Y IN (1,3,5,7,9,11,13,15)then do;D1LC6+L6;END;
grpcnt1+cnt1;bigl1+l1;bigl2+l2;bigl3+l3;bigl4+l4;bigl5+l5;bigl6+l
6;all=1;yh=y;run;
%let OLDLC12=&LC12;%let OLDLC22=&LC22;%let OLDLC32=&LC32;
%let OLDLC42=&LC42;%let OLDLC52=&LC52;%let OLDLC62=&LC62;
%let OLDCPA1_2=&CPA1_2;%let OLDCPB1_2=&CPB1_2;
%let OLDCPC1_2=&CPC1_2;%let OLDCPD1_2=&CPD1_2;
data cmg32; set CMG32(keep = y2 cnt2 notagain1 notagain2 yh all);
retain grpcnt2 0 bigl12 0 bigl22 0 bigl32 0 bigl42 0 bigl52 0
bigl62 0;
    if Y2=1 then do;  /*calculate P values FOR GRP2*/
P12=&LC12*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;end;
    else if Y2=2 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
    else if Y2=3 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;end;
    else if Y2=4 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;end;
    else if Y2=5 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;end;
    else if Y2=6 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
```

```
P22=&LC22*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;end;
      else if Y2=7 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;end;
      else if Y2=8 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;end;
      else if Y2=9 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;end;
      else if Y2=10 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
      else if Y2=11 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
      else if Y2=12 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;end;
      else if Y2=13 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
```

```
P52=&LC52*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;end;
     else if Y2=14 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;end;
     else if Y2=15 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;end;
     else if Y2=16 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;end;
TOTP2=P12+P22+P32+P42+P52+P62;IF CNT2=. THEN CNT2=0;
retain A1LC12 0 A1LC22 0 A1LC32 0 A1LC42 0 A1LC52 0 A1LC62 0;
L12=(CNT2*P12)/TOTP2;L22=(CNT2*P22)/TOTP2;L32=(CNT2*P32)/TOTP2;
L42=(CNT2*P42)/TOTP2;L52=(CNT2*P52)/TOTP2;L62=(CNT2*P62)/TOTP2;
if Y2 ge 9 then do;A1LC12+L12;A1LC22+L22;end;
if Y2 le 8 then
do;A1LC32+L32;A1LC42+L42;A1LC52+L52;A1LC62+L62;end;
if Y2 IN (5,6,7,8,13,14,15,16)then do;B1LC12+L12;B1LC32+L32;end;
if Y2 IN (1,2,3,4,9,10,11,12)then do;
B1LC22+L22;B1LC42+L42;B1LC52+L52;B1LC62+L62;end;
if Y2 IN (3,4,7,8,11,12,15,16)then do;
C1LC12+L12;C1LC22+L22;C1LC32+L32;C1LC42+L42;END;
if Y2 IN (1,2,5,6,9,10,13,14)then do;C1LC52+L52;C1LC62+L62;END;
if Y2 IN (2,4,6,8,10,12,14,16)then do;
D1LC12+L12;D1LC22+L22;D1LC32+L32;D1LC42+L42;D1LC52+L52;END;
if Y2 IN (1,3,5,7,9,11,13,15)then do;D1LC62+L62;END;grpcnt2+cnt2;
bigl12+l12;bigl22+l22;bigl32+l32;bigl42+l42;bigl52+l52;bigl62+l62
;all=1;yh=y2;run;
proc SORT data=CMG3; BY YH; run;proc SORT data=CMG32; BY YH; run;
DATA CMG3; MERGE CMG3 CMG32; BY YH; run;
data cm4(keep=y y2 lc1-lc6 cpa1 cpb1 cpc1 cpd1 cpa2 cpb2 cpc2
cpd2
lc12 lc22 lc32 lc42 lc52 lc62 cpa1_2 cpb1_2 cpc1_2 cpd1_2 cpa2_2
cpb2_2 cpc2_2 cpd2_2 cnt1 cnt2 grpcnt1 grpcnt2 totp1 totp2 yh);
set cmg3; by all;if last.all;
LC1=BIGL1/GRPCNT1;LC2=BIGL2/GRPCNT1;LC3=BIGL3/GRPCNT1;
LC4=BIGL4/GRPCNT1;LC5=BIGL5/GRPCNT1;LC6=BIGL6/GRPCNT1;
LCA2=A1LC1+A1LC2+A1LC3+A1LC4+A1LC5+A1LC6;
LCB2=B1LC1+B1LC2+B1LC3+B1LC4+B1LC5+B1LC6;
```

```
LCC2=C1LC1+C1LC2+C1LC3+C1LC4+C1LC5+C1LC6;
LCD2=D1LC1+D1LC2+D1LC3+D1LC4+D1LC5+D1LC6;
CPA2=LCA2/GRPCNT1;CPB2=LCB2/GRPCNT1;
CPC2=LCC2/GRPCNT1;CPD2=LCD2/GRPCNT1;
CPA1=(1-CPA2);CPB1=(1-CPB2);
CPC1=(1-CPC2);CPD1=(1-CPD2);
%if &endvar=0 %then %do;
call symput('lc1',lc1);call symput('lc2',lc2);
call symput('lc3',lc3);call symput('lc4',lc4);
call symput('lc5',lc5);call symput('lc6',lc6);
call symput('cpa1',cpa1);call symput('cpb1',cpb1);
call symput('cpc1',cpc1);call symput('cpd1',cpd1);
call symput('cpa2',cpa2);call symput('cpb2',cpb2);
call symput('cpc2',cpc2);call symput('cpd2',cpd2);%end;
ablc1=abs(&oldlc1-lc1);ablc2=abs(&oldlc2-lc2);
ablc3=abs(&oldlc3-lc3);ablc4=abs(&oldlc4-lc4);
ablc5=abs(&oldlc5-lc5);
aba1=abs(&oldcpa1-cpa1);abb1=abs(&oldcpb1-cpb1);
abc1=abs(&oldcpc1-cpc1);abd1=abs(&oldcpd1-cpd1);
TOTABS=ABLC1+ABLC2+ABLC3+ABLC4+ABLC5+ABA1+ABB1+ABC1+ABD1;
IF TOTABS LE .0001 AND NOTAGAIN1 NE 1 THEN DO;NOTAGAIN1=1;
call symput('lc1',lc1);call symput('lc2',lc2);
call symput('lc3',lc3);call symput('lc4',lc4);
call symput('lc5',lc5);call symput('lc6',lc6);
call symput('cpa1',cpa1);call symput('cpb1',cpb1);
call symput('cpc1',cpc1);call symput('cpd1',cpd1);
call symput('TOTP1',TOTP1);call symput('CNT1',CNT1);
call symput('GRPCNT1',GRPCNT1);END;
LC12=BIGL12/GRPCNT2;LC22=BIGL22/GRPCNT2;LC32=BIGL32/GRPCNT2;
LC42=BIGL42/GRPCNT2;LC52=BIGL52/GRPCNT2;LC62=BIGL62/GRPCNT2;
LCA22=A1LC12+A1LC22+A1LC32+A1LC42+A1LC52+A1LC62;
LCB22=B1LC12+B1LC22+B1LC32+B1LC42+B1LC52+B1LC62;
LCC22=C1LC12+C1LC22+C1LC32+C1LC42+C1LC52+C1LC62;
LCD22=D1LC12+D1LC22+D1LC32+D1LC42+D1LC52+D1LC62;
CPA2_2=LCA22/GRPCNT2;CPB2_2=LCB22/GRPCNT2;
CPC2_2=LCC22/GRPCNT2;CPD2_2=LCD22/GRPCNT2;
CPA1_2=(1-CPA2_2);CPB1_2=(1-CPB2_2);
CPC1_2=(1-CPC2_2);CPD1_2=(1-CPD2_2);
%if (&endvar=0) %then %do;
call symput('lc12',lc12);call symput('lc22',lc22);
call symput('lc32',lc32);call symput('lc42',lc42);
call symput('lc52',lc52);call symput('lc62',lc62);
call symput('cpa1_2',cpa1_2);call symput('cpb1_2',cpb1_2);
call symput('cpc1_2',cpc1_2);call symput('cpd1_2',cpd1_2);
call symput('cpa2_2',cpa2_2);call symput('cpb2_2',cpb2_2);
call symput('cpc2_2',cpc2_2);call symput('cpd2_2',cpd2_2);%end;
ablc12=abs(&oldlc12-lc12);ablc22=abs(&oldlc22-lc22);
ablc32=abs(&oldlc32-lc32);ablc42=abs(&oldlc42-lc42);
ablc52=abs(&oldlc52-lc52);ablc62=abs(&oldlc62-lc62);
         aba12=abs(&oldcpa1_2-cpa1_2);abb12=abs(&oldcpb1_2-
cpb1_2);
```

```
        abc12=abs(&oldcpc1_2-cpc1_2);abd12=abs(&oldcpd1_2-
cpd1_2);
TOTABS2=sum(ABLC12,ABLC22,ABLC32,ABLC42,ABLC52,
ABA12,ABB12,ABC12,ABD12);
IF TOTABS2 LE .0001 AND NOTAGAIN2 NE 1 THEN DO;NOTAGAIN2=1;
call symput('lc12',lc12);call symput('lc22',lc22);
call symput('lc32',lc32);call symput('lc42',lc42);
call symput('lc52',lc52);call symput('lc62',lc62);
call symput('cpa1_2',cpa1_2);call symput('cpb1_2',cpb1_2);
call symput('cpc1_2',cpc1_2);call symput('cpd1_2',cpd1_2);
call symput('TOTP2',TOTP2);call symput('CNT2',CNT2);
call symput('GRPCNT2',GRPCNT2);end;
IF NOTAGAIN1=1 AND NOTAGAIN2=1 THEN DO;
CALL SYMPUT ('endvar',1);END;run;
%if (&endvar^=1) %then %let cnt=%eval(&cnt+1);%if (&cnt>=500)
%then %do;%let endvar=1;%end;%end;
%mend cm;%cm
data WHET1 (keep=lc1-lc5 cpa1 cpb1 cpc1 cpd1)
     WHET2 (keep=lc12 LC22 LC32 LC42 LC52 cpa1_2 cpb1_2 cpc1_2
cpd1_2);set cm4;run;


/*Start Wald*/
%let LC1=.08;%let LC2=.03;%let LC3=.08;%let LC4=.50;
%let LC5=.23;%let LC6=.08;
%let LC12=.10;%let LC22=.05;%let LC32=.10;%let LC42=.40;
%let LC52=.25;%let LC62=.10;
%let CPA1=.75;%let CPB1=.75;%let CPC1=.75;%let CPD1=.75;
%let CPA2=.25;%let CPB2=.25;%let CPC2=.25;%let CPD2=.25;
%let CPA1_2=.75;%let CPB1_2=.75;%let CPC1_2=.75;%let CPD1_2=.75;
%let CPA2_2=.25;%let CPB2_2=.25;%let CPC2_2=.25;%let CPD2_2=.25;
%let cnt=0;%let endvar=0;
%macro wald;%do cell=1 %to 16;%let cnt=0;%let endvar=0;
data Wald&cell.;set cmg3;IF Y=&CELL THEN PSCNT1=(CNT1-1);
ELSE PSCNT1=CNT1;run;
%do %until (&endvar=1);%if &cnt>0 %then %do;
%let LC1=&LC1_n;%let LC2=&LC2_n;%let LC3=&LC3_n;
%let LC4=&LC4_n;%let LC5=&LC5_n;%let LC6=&LC6_n;
%let CPA1=&CPA1_n;%let CPB1=&CPB1_n;
%let CPC1=&CPC1_n;%let CPD1=&CPD1_n;
%let CPA2=&CPA2_n;%let CPB2=&CPB2_n;
%let CPC2=&CPC2_n;%let CPD2=&CPD2_n;%end;
     DATA Wald&cell.;set Wald&cell.(keep=y pscnt1);
retain grpcnt1 0 bigl1 0 bigl2 0 bigl3 0 bigl4 0 bigl5 0 bigl6 0;
if Y=1 then do;
P1=&LC1*&CPA1*&CPB1*&CPC1*&CPD1;P2=&LC2*&CPA1*&CPB2*&CPC1*&CPD1;
P3=&LC3*&CPA2*&CPB1*&CPC1*&CPD1;P4=&LC4*&CPA2*&CPB2*&CPC1*&CPD1;
P5=&LC5*&CPA2*&CPB2*&CPC2*&CPD1;P6=&LC6*&CPA2*&CPB2*&CPC2*&CPD2;e
nd;
        else if Y=2 then do;
P1=&LC1*&CPA1*&CPB1*&CPC1*&CPD2;P2=&LC2*&CPA1*&CPB2*&CPC1*&CPD2;
P3=&LC3*&CPA2*&CPB1*&CPC1*&CPD2;P4=&LC4*&CPA2*&CPB2*&CPC1*&CPD2;
```

```
P5=&LC5*&CPA2*&CPB2*&CPC2*&CPD2;P6=&LC6*&CPA2*&CPB2*&CPC2*&CPD1;e
nd;
        else if Y=3 then do;
P1=&LC1*&CPA1*&CPB1*&CPC2*&CPD1;P2=&LC2*&CPA1*&CPB2*&CPC2*&CPD1;
P3=&LC3*&CPA2*&CPB1*&CPC2*&CPD1;P4=&LC4*&CPA2*&CPB2*&CPC2*&CPD1;
P5=&LC5*&CPA2*&CPB2*&CPC1*&CPD1;P6=&LC6*&CPA2*&CPB2*&CPC1*&CPD2;e
nd;
        else if Y=4 then do;
P1=&LC1*&CPA1*&CPB1*&CPC2*&CPD2;P2=&LC2*&CPA1*&CPB2*&CPC2*&CPD2;
P3=&LC3*&CPA2*&CPB1*&CPC2*&CPD2;P4=&LC4*&CPA2*&CPB2*&CPC2*&CPD2;
P5=&LC5*&CPA2*&CPB2*&CPC1*&CPD2;P6=&LC6*&CPA2*&CPB2*&CPC1*&CPD1;e
nd;
        else if Y=5 then do;
P1=&LC1*&CPA1*&CPB2*&CPC1*&CPD1;P2=&LC2*&CPA1*&CPB1*&CPC1*&CPD1;
P3=&LC3*&CPA2*&CPB2*&CPC1*&CPD1;P4=&LC4*&CPA2*&CPB1*&CPC1*&CPD1;
P5=&LC5*&CPA2*&CPB1*&CPC2*&CPD1;P6=&LC6*&CPA2*&CPB1*&CPC2*&CPD2;e
nd;
        else if Y=6 then do;
P1=&LC1*&CPA1*&CPB2*&CPC1*&CPD2;P2=&LC2*&CPA1*&CPB1*&CPC1*&CPD2;
P3=&LC3*&CPA2*&CPB2*&CPC1*&CPD2;P4=&LC4*&CPA2*&CPB1*&CPC1*&CPD2;
P5=&LC5*&CPA2*&CPB1*&CPC2*&CPD2;P6=&LC6*&CPA2*&CPB1*&CPC2*&CPD1;e
nd;
        else if Y=7 then do;
P1=&LC1*&CPA1*&CPB2*&CPC2*&CPD1;P2=&LC2*&CPA1*&CPB1*&CPC2*&CPD1;
P3=&LC3*&CPA2*&CPB2*&CPC2*&CPD1;P4=&LC4*&CPA2*&CPB1*&CPC2*&CPD1;
P5=&LC5*&CPA2*&CPB1*&CPC1*&CPD1;P6=&LC6*&CPA2*&CPB1*&CPC1*&CPD2;e
nd;
        else if Y=8 then do;
P1=&LC1*&CPA1*&CPB2*&CPC2*&CPD2;P2=&LC2*&CPA1*&CPB1*&CPC2*&CPD2;
P3=&LC3*&CPA2*&CPB2*&CPC2*&CPD2;P4=&LC4*&CPA2*&CPB1*&CPC2*&CPD2;
P5=&LC5*&CPA2*&CPB1*&CPC1*&CPD2;P6=&LC6*&CPA2*&CPB1*&CPC1*&CPD1;e
nd;
        else if Y=9 then do;
P1=&LC1*&CPA2*&CPB1*&CPC1*&CPD1;P2=&LC2*&CPA2*&CPB2*&CPC1*&CPD1;
P3=&LC3*&CPA1*&CPB1*&CPC1*&CPD1;P4=&LC4*&CPA1*&CPB2*&CPC1*&CPD1;
P5=&LC5*&CPA1*&CPB2*&CPC2*&CPD1;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD2;e
nd;
        else if Y=10 then do;
P1=&LC1*&CPA2*&CPB1*&CPC1*&CPD2;P2=&LC2*&CPA2*&CPB2*&CPC1*&CPD2;
P3=&LC3*&CPA1*&CPB1*&CPC1*&CPD2;P4=&LC4*&CPA1*&CPB2*&CPC1*&CPD2;
P5=&LC5*&CPA1*&CPB2*&CPC2*&CPD2;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD1;e
nd;
        else if Y=11 then do;
P1=&LC1*&CPA2*&CPB1*&CPC2*&CPD1;P2=&LC2*&CPA2*&CPB2*&CPC2*&CPD1;
P3=&LC3*&CPA1*&CPB1*&CPC2*&CPD1;P4=&LC4*&CPA1*&CPB2*&CPC2*&CPD1;
P5=&LC5*&CPA1*&CPB2*&CPC1*&CPD1;P6=&LC6*&CPA1*&CPB2*&CPC2*&CPD1;e
nd;
        else if Y=12 then do;
P1=&LC1*&CPA2*&CPB1*&CPC2*&CPD2;P2=&LC2*&CPA2*&CPB2*&CPC2*&CPD2;
P3=&LC3*&CPA1*&CPB1*&CPC2*&CPD2;P4=&LC4*&CPA1*&CPB2*&CPC2*&CPD2;
P5=&LC5*&CPA1*&CPB2*&CPC1*&CPD2;P6=&LC6*&CPA1*&CPB2*&CPC1*&CPD1;e
nd;
```

```
        else if Y=13 then do;
P1=&LC1*&CPA2*&CPB2*&CPC1*&CPD1;P2=&LC2*&CPA2*&CPB1*&CPC1*&CPD1;
P3=&LC3*&CPA1*&CPB2*&CPC1*&CPD1;P4=&LC4*&CPA1*&CPB1*&CPC1*&CPD1;
P5=&LC5*&CPA1*&CPB1*&CPC2*&CPD1;P6=&LC6*&CPA1*&CPB1*&CPC2*&CPD2;e
nd;
        else if Y=14 then do;
P1=&LC1*&CPA2*&CPB2*&CPC1*&CPD2;P2=&LC2*&CPA2*&CPB1*&CPC1*&CPD2;
P3=&LC3*&CPA1*&CPB2*&CPC1*&CPD2;P4=&LC4*&CPA1*&CPB1*&CPC1*&CPD2;
P5=&LC5*&CPA1*&CPB1*&CPC2*&CPD2;P6=&LC6*&CPA1*&CPB1*&CPC2*&CPD1;e
nd;
        else if Y=15 then do;
P1=&LC1*&CPA2*&CPB2*&CPC2*&CPD1;P2=&LC2*&CPA2*&CPB1*&CPC2*&CPD1;
P3=&LC3*&CPA1*&CPB2*&CPC2*&CPD1;P4=&LC4*&CPA1*&CPB1*&CPC2*&CPD1;
P5=&LC5*&CPA1*&CPB1*&CPC1*&CPD1;P6=&LC6*&CPA1*&CPB1*&CPC1*&CPD2;e
nd;
        else if Y=16 then do;
P1=&LC1*&CPA2*&CPB2*&CPC2*&CPD2;P2=&LC2*&CPA2*&CPB1*&CPC2*&CPD2;
P3=&LC3*&CPA1*&CPB2*&CPC2*&CPD2;P4=&LC4*&CPA1*&CPB1*&CPC2*&CPD2;
P5=&LC5*&CPA1*&CPB1*&CPC1*&CPD2;P6=&LC6*&CPA1*&CPB1*&CPC1*&CPD1;e
nd;
            TOTP=P1+P2+P3+P4+P5+P6;
            retain A1LC1 0 A1LC2 0 A1LC3 0 A1LC4 0 A1LC5 0 A1LC6
0;
     L1=(pscnt1*P1)/TOTP;L2=(pscnt1*P2)/TOTP;L3=(pscnt1*P3)/TOTP
;
     L4=(pscnt1*P4)/TOTP;L5=(pscnt1*P5)/TOTP;L6=(pscnt1*P6)/TOTP
;
if Y ge 9 then do;A1LC1+L1;A1LC2+L2;end;
if Y le 8 then do;
A1LC3+L3;A1LC4+L4;A1LC5+L5;A1LC6+L6;end;
if Y IN (5,6,7,8,13,14,15,16)then do;B1LC1+L1;B1LC3+L3;end;
if Y IN (1,2,3,4,9,10,11,12)then do;
B1LC2+L2;B1LC4+L4;B1LC5+L5;B1LC6+L6;end;
if Y IN (3,4,7,8,11,12,15,16)then do;
C1LC1+L1;C1LC2+L2;C1LC3+L3;C1LC4+L4;END;
if Y IN (1,2,5,6,9,10,13,14)then do;C1LC5+L5;C1LC6+L6;END;
if Y IN (2,4,6,8,10,12,14,16)then do;
D1LC1+L1;D1LC2+L2;D1LC3+L3;D1LC4+L4;D1LC5+L5;END;
if Y IN (1,3,5,7,9,11,13,15)then do;D1LC6+L6;END;grpcnt1+pscnt1;
bigl1+l1;bigl2+l2;bigl3+l3;bigl4+l4;bigl5+l5;bigl6+l6;all=1;run;
%let OLDLC1=&LC1;%let OLDLC2=&LC2;%let OLDLC3=&LC3;
%let OLDLC4=&LC4;%let OLDLC5=&LC5;%let OLDLC6=&LC6;
%let OLDCPA1=&CPA1;%let OLDCPB1=&CPB1;
%let OLDCPC1=&CPC1;%let OLDCPD1=&CPD1;
     data wald&cell.last;set wald&cell.;by all;
if last.all;cell=&cell;
LC1=BIGL1/GRPCNT1;LC2=BIGL2/GRPCNT1;LC3=BIGL3/GRPCNT1;
LC4=BIGL4/GRPCNT1;LC5=BIGL5/GRPCNT1;LC6=BIGL6/GRPCNT1;
LCA2=A1LC1+A1LC2+A1LC3+A1LC4+A1LC5+A1LC6;
LCB2=B1LC1+B1LC2+B1LC3+B1LC4+B1LC5+B1LC6;
LCC2=C1LC1+C1LC2+C1LC3+C1LC4+C1LC5+C1LC6;
LCD2=D1LC1+D1LC2+D1LC3+D1LC4+D1LC5+D1LC6;
```

```
CPA2=LCA2/GRPCNT1;CPB2=LCB2/GRPCNT1;
CPC2=LCC2/GRPCNT1;CPD2=LCD2/GRPCNT1;
CPA1=(1-CPA2);CPB1=(1-CPB2);CPC1=(1-CPC2);CPD1=(1-CPD2);
          %if &endvar=0 %then %do;
call symput('lc1_n',lc1);call symput('lc2_n',lc2);
call symput('lc3_n',lc3);call symput('lc4_n',lc4);
call symput('lc5_n',lc5);call symput('lc6_n',lc6);
call symput('cpa1_n',cpa1);call symput('cpb1_n',cpb1);
call symput('cpc1_n',cpc1);call symput('cpd1_n',cpd1);
call symput('cpa2_n',cpa2);call symput('cpb2_n',cpb2);
call symput('cpc2_n',cpc2);call symput('cpd2_n',cpd2);%end;
ablc1=abs(&oldlc1-lc1);ablc2=abs(&oldlc2-lc2);
ablc3=abs(&oldlc3-lc3);ablc4=abs(&oldlc4-lc4);
ablc5=abs(&oldlc5-lc5);ablc6=abs(&oldlc6-lc6);
aba1=abs(&oldcpa1-cpa1);abb1=abs(&oldcpb1-cpb1);
abc1=abs(&oldcpc1-cpc1);abd1=abs(&oldcpd1-cpd1);
TOTABS=ABLC1+ABLC2+ABLC3+ABLC4+ABLC5+ABA1+ABB1+ABC1+ABD1;
if totabs le .0001 then call symput('endvar',1);run;
          %if (&endvar^=1) %then %let cnt=%eval(&cnt+1);
        %if &cnt>= 500 %then %do; %let endvar=1;%end;%end;
            data wald&cell.last(keep=lc1-lc5 cpa1 cpb1 cpc1 cpd1);
            set wald&cell.last;run;%end;
%mend wald;%wald;
data waldall_1;
set wald1last wald2last wald3last wald4last wald5last wald6last
wald7last wald8last wald9last wald10last wald11last wald12last
wald13last wald14last wald15last wald16last;
%let cnt=0;%let endvar=0;
%macro wald2;
%do cell=1 %to 16;%let cnt=0;%let endvar=0;
data Wald2_&cell.; set cmg32;IF Y2=&CELL THEN PSCNT2=(CNT2-1);
ELSE PSCNT2=CNT2;run;
      %do %until (%eval(&endvar)=1);%if &cnt>0 %then %do;
%let LC12=&LC12_n;%let LC22=&LC22_n;%let LC32=&LC32_n;
%let LC42=&LC42_n;%let LC52=&LC52_n;%let LC62=&LC62_n;
%let CPA1_2=&CPA12_n;%let CPB1_2=&CPB12_n;
%let CPC1_2=&CPC12_n;%let CPD1_2=&CPD12_n;
%let CPA2_2=&CPA22_n;%let CPB2_2=&CPB22_n;
%let CPC2_2=&CPC22_n;%let CPD2_2=&CPD22_n;%end;
      DATA Wald2_&cell.;set Wald2_&cell.(keep=y2 pscnt2);
retain grpcnt2 0 bigl12 0 bigl22 0 bigl32 0 bigl42 0 bigl52 0
bigl62 0;
if Y2=1 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;end;
                else if Y2=2 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
```

```
P32=&LC32*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
                else if Y2=3 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;end;
            else if Y2=4 then do;
P12=&LC12*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;end;
            else if Y2=5 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;end;
            else if Y2=6 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;end;
            else if Y2=7 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;end;
            else if Y2=8 then do;
P12=&LC12*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;end;
            else if Y2=9 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
```

```
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;end;
                else if Y2=10 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
                else if Y2=11 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;end;
                else if Y2=12 then do;
P12=&LC12*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;end;
                else if Y2=13 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC1_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC1_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC1_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;end;
                else if Y2=14 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC1_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC1_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC1_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;end;
                else if Y2=15 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC2_2*&CPD1_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC2_2*&CPD1_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC2_2*&CPD1_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC2_2*&CPD1_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;end;
                else if Y2=16 then do;
P12=&LC12*&CPA2_2*&CPB2_2*&CPC2_2*&CPD2_2;
P22=&LC22*&CPA2_2*&CPB1_2*&CPC2_2*&CPD2_2;
P32=&LC32*&CPA1_2*&CPB2_2*&CPC2_2*&CPD2_2;
P42=&LC42*&CPA1_2*&CPB1_2*&CPC2_2*&CPD2_2;
P52=&LC52*&CPA1_2*&CPB1_2*&CPC1_2*&CPD2_2;
P62=&LC62*&CPA1_2*&CPB1_2*&CPC1_2*&CPD1_2;end;
            TOTP2=P12+P22+P32+P42+P52+P62;
retain A1LC12 0 A1LC22 0 A1LC32 0 A1LC42 0 A1LC52 0 A1LC62 0;
```

```
L12=(PSCNT2*P12)/TOTP2;L22=(PSCNT2*P22)/TOTP2;
L32=(PSCNT2*P32)/TOTP2;L42=(PSCNT2*P42)/TOTP2;
L52=(PSCNT2*P52)/TOTP2;L62=(PSCNT2*P62)/TOTP2;
if Y2 ge 9 then do;A1LC12+L12;A1LC22+L22;end;
if Y2 le 8 then
do;A1LC32+L32;A1LC42+L42;A1LC52+L52;A1LC62+L62;end;
if Y2 IN (5,6,7,8,13,14,15,16)then do;B1LC12+L12;B1LC32+L32;end;
if Y2 IN (1,2,3,4,9,10,11,12)then do;
B1LC22+L22;B1LC42+L42;B1LC52+L52;B1LC62+L62;end;
if Y2 IN (3,4,7,8,11,12,15,16)then do;
C1LC12+L12;C1LC22+L22;C1LC32+L32;C1LC42+L42;END;
if Y2 IN (1,2,5,6,9,10,13,14)then do;C1LC52+L52;C1LC62+L62;END;
if Y2 IN (2,4,6,8,10,12,14,16)then do;
D1LC12+L12;D1LC22+L22;D1LC32+L32;D1LC42+L42;D1LC52+L52;END;
if Y2 IN (1,3,5,7,9,11,13,15)then
do;D1LC62+L62;END;grpcnt2+pscnt2;
bigl12+l12;bigl22+l22;bigl32+l32;
bigl42+l42;bigl52+l52;bigl62+l62;all=1;run;
      %let OLDLC12=LC12;%let OLDLC22=LC22;%let OLDLC32=LC32;
     %let OLDLC42=LC42;%let OLDLC52=LC52;%let OLDLC62=LC62;
      %let OLDCPA1_2=CPA1_2;%let OLDCPB1_2=CPB1_2;
      %let OLDCPC1_2=CPC1_2;%let OLDCPD1_2=CPD1_2;
      data wald2_&cell.last;set wald2_&cell.;by all;if
last.all;cell=&cell

     LC12=BIGL12/GRPCNT2;LC22=BIGL22/GRPCNT2;LC32=BIGL32/GRPCNT2
;
     LC42=BIGL42/GRPCNT2;LC52=BIGL52/GRPCNT2;LC62=BIGL62/GRPCNT2
;
            LCA22=A1LC12+A1LC22+A1LC32+A1LC42+A1LC52+A1LC62;
            LCB22=B1LC12+B1LC22+B1LC32+B1LC42+B1LC52+B1LC62;
            LCC22=C1LC12+C1LC22+C1LC32+C1LC42+C1LC52+C1LC62;
            LCD22=D1LC12+D1LC22+D1LC32+D1LC42+D1LC52+D1LC62;
            CPA2_2=LCA22/GRPCNT2;CPB2_2=LCB22/GRPCNT2;
        CPC2_2=LCC22/GRPCNT2;CPD2_2=LCD22/GRPCNT2;
            CPA1_2=(1-CPA2_2);CPB1_2=(1-CPB2_2);
            CPC1_2=(1-CPC2_2);CPD1_2=(1-CPD2_2);
            %if &endvar=0 %then %do;
call symput('lc12_n',lc12);call symput('lc22_n',lc22);
call symput('lc32_n',lc32);call symput('lc42_n',lc42);
call symput('lc52_n',lc52);call symput('lc62_n',lc62);
call symput('cpa12_n',cpa1_2);call symput('cpb12_n',cpb1_2);
call symput('cpc12_n',cpc1_2);call symput('cpd12_n',cpd1_2);
call symput('cpa22_n',cpa2_2);call symput('cpb22_n',cpb2_2);
call symput('cpc22_n',cpc2_2);call symput('cpd22_n',cpd2_2);%end;
ablc12=abs(&oldlc12-lc12);
ablc22=abs(&oldlc22-lc22);ablc32=abs(&oldlc32-lc32);
ablc42=abs(&oldlc42-lc42);
ablc52=abs(&oldlc52-lc52);ablc62=abs(&oldlc62-lc62);
aba12=abs(&oldcpa1_2-cpa1_2);abb12=abs(&oldcpb1_2-cpb1_2);
abc12=abs(&oldcpc1_2-cpc1_2);abd12=abs(&oldcpd1_2-cpd1_2);
TOTABS2=sum(ABLC12,ABLC22,ABLC32,ABLC42,ABLC52,
```

```
ABA12,ABB12,ABC12,ABD12);
      if totabs2 le .0001 then call symput('endvar',1);run;
     %if (&endvar^=1) %then %let cnt=%eval(&cnt+1);%if &cnt>= 500
%then %do;%let endvar = 1;%end;%end;
data wald2_&cell.last(keep=LC12 LC22 LC32 LC42 LC52
cpa1_2 cpb1_2 cpc1_2 cpd1_2);
set wald2_&cell.last;run;%end;
%mend wald2;%wald2;
data waldall_2;
set wald2_1last wald2_2last wald2_3last wald2_4last wald2_5last
    wald2_6last wald2_7last wald2_8last wald2_9last wald2_10last
    wald2_11last wald2_12last wald2_13last wald2_14last
wald2_15last    wald2_16last;
proc iml;
use waldall_1; read all into W1;/*16X9*/
use waldall_2; read all into W2;/*16X9*/
reset log NOprint;tmp=(W1-
repeat(W1[:,],nrow(W1)));c1=(tmp`*tmp)/(nrow(W1)-1);
   create ZZ1 from c1;append from c1;
reset log NOprint;tmp=(W2-
repeat(W2[:,],nrow(W2)));c2=(tmp`*tmp)/(nrow(W2)-1);
create ZZ2 from c2;append from c2;
X=j(9,9,0);X2=j(9,9,0);WF1=c1//X;WF2=X2//c2;
VAR=WF1||WF2; VAR=VAR # 500;
Use WHET1; read all into H1;Use WHET2; read all into H2; /*9X1*/
H1=T(H1);H2=T(H2);THETA=H1//H2;
RHOM={ 1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0,
       0 1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0,
       0 0 1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0,
       0 0 0 1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0,
       0 0 0 0 1 0 0 0 0 0 0 0 0 -1 0 0 0 0 0,
       0 0 0 0 0 1 0 0 0 0 0 0 0 0 -1 0 0 0 0,
       0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -1 0 0 0,
       0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -1 0 0,
       0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -1 0,
       0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -1};
RPAR={ 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 -1 0 0 0,
       0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 -1 0 0,
       0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 -1 0,
       0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 -1};
HOMF=RHOM*THETA;T_HOMF=T(HOMF);T_RHOM=T(RHOM);
VHOM=RHOM*VAR*T_RHOM;IVHOM=INV(VHOM);
WALDH=T_HOMF*IVHOM*HOMF;PARF=RPAR*THETA;
T_PARF=T(PARF);T_RPAR=T(RPAR);
VPAR=RPAR*VAR*T_RPAR;IVPAR=INV(VPAR);
WALDP=T_PARF*IVPAR*PARF;WALD=WALDH||WALDP; cname={"WALDH"
"WALDP"};
   create COUNTW from wald[COLNAME=CNAME];append from WALD;Quit;
   /*COUNTING*/
data COUNTNEW_W(KEEP=CNT_W HET_NOT); SET COUNTW;
          IF Waldp<9.4877 & Waldh>16.91898 THEN
CNT_W=1;Elsecnt_W=0;cnt_het=&cnt;
          IF CNT_HET>=500 THEN HET_NOT=1;ELSE HET_NOT=0;RUN;
```

```
%if &k=1 %then %do;data finalW;set countnew_W;run;%end;%else %do;
proc append base=finalW data=countnew_W;run;%end;%end;
%mend jack;%Jack;
proc means data=finalW noprint NWAY;
output out=SAVELIB.countsumW2 sum=;run;
DATA _NULL_;CALL SYMPUT('END',PUT(TIME(),TIME8.));RUN;%put &END;
```

REFERENCES

Akaike, H. (1973). "Information theory and an extension of the maximum likelihood principle," in *2nd International Symposium on Information Theory*. eds. B. N. Petrov and F. Csaki, Budapest: Akademiai Kiado, pp. 267-281.

Akaike, H. (1987). "Factor analysis and AIC," *Psychometrika*, 52, 317-332.

Blaze-Temple, D., and Lo, S. K. (1992). "Stages of drug use: A community survey of Perth teenagers," *British Journal of Addiction*, 87, 215-25.

Bock, R. D., and Lieberman, M. (1970). "Fitting a response model for n dichotomously scored items," *Psychometrika*, 35, 179-197.

Clogg, C. C., and Goodman, L. A. (1984). "Latent structure analysis of a set of multidimensional tables," *Journal of the American Statistical Association*, 79, 762-771.

Clogg, C. C., and Goodman, L. A. (1985). "Simultaneous latent structure analysis in several groups," in *Sociological Methodology 1985*, ed. N. B. Tuma, San Francisco: Jossey-Bass.

Clogg, C. C., and Goodman, L. A. (1986). "On scaling models applied to data from several groups," *Psychometrika*, 51, 123-135.

Clogg, C. C., and Sawyer, D. O. (1981). "A comparison of alternative models for analyzing the scalability of response patterns," in *Sociological Methodology 1981*. ed. S. Leinhardt, San Francisco: Josey-Bass.

Collins, L. M., et al (1994). "Crossvalidation of Latent Class Models of early substance use onset," *Multivariate Behavioral Research*, 29, 165-183.

Dayton, C. M. (1999). *Latent Class Scaling Analysis*. Sage University Papers Series on Quantitative Applications in the Social Sciences, 07-126. Thousand Oaks, CA: Sage Publications.

Dayton, C. M., and Macready, G. B. (1976). "A probabilistic model for the validation of behavioral hierarchies," *Psychometrika*, 41, 189-204.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). "Maximum Likelihood estimation from incomplete data via the EM Algorithm," *Journal of the Royal Statistical Society*, 39, 1-38.

Evans, M., Hastings, N., and Peacock, B. (1993). *Statistical Distributions (2nd ed.)*. New York: John Wiley.

Golub, A., and Johnson, B. D. (2001). "Variation in youthful risks of progression from alcohol and tobacco to marijuana and to hard drugs across generations," *American Journal of Public Health*, 91, 225-232.

Guttman, L., (1947). "On Festinger's evaluation of scale analysis," *Psychological Bulletin*, 44, 451-465.

Haberman, S. J. (1979). *Analysis of Qualitative Data. Vol 2. New Developments*, New York: Academic Press.

Hambleton, R. K., and Swaminathan, H. (1985). *Item Response Theory: Principles and Application*. Boston, MA: Kluwer Nijhoff.

Johnson, P. B., Boles, S. M., Vaughan, R., and Kleber, H. D. (2000). "The co-occurrence of smoking and binge drinking in adolescence," *Addictive Behaviors*, 25, 779-83.

Kalton G. (1983). *Introduction to Survey Sampling*. Newbury Park, CA: Sage Publications.

Kandel, D. B., and Faust, R. (1975). "Sequences and stages in patterns of adolescent drug use," *Archives of General Psychiatry*, 32, 923–932.

Kandel, D. B., Yamaguchi, K., and Chen, K. (1992). "Stages of progression in drug involvement from adolescence to adulthood: Further evidence for the gateway theory," *Journal on Studies on Alcohol*, 53, 447–457.

Kish, L. (1965). *Survey Sampling*. New York: John Wiley and Sons.

Korn, E. L., and Graubard, B. I. (1999). *Analysis of Health Surveys*. New York: John Wiley and Sons.

Lai, S., et al. (2000). "The Association between cigarette smoking and drug abuse in the United States," *Journal of Addictive Diseases*, 19, 11-23.

Lee, E. S., Forthofer, R. N., and Lorimor, R. J. (1986). "Analysis of complex sample survey data: Problems and strategies," *Sociological Methods & Research*, 15, 69-100.

Lehtonen, R., and Pahkinen, E. J. (1994). *Practical Methods for Design and Analysis of Complex Surveys*. New York: John Wiley and Sons.

Lohr, S. L. (1999). *Sampling: Design and Analysis*. Duxbury Press.

McCutcheon, A. L. (1987). *Latent Class Analysis*. Sage University Papers Series on Quantitative Applications in the Social Sciences, 07-64. Thousand Oaks, CA: Sage Publications.

Mooney, C. Z., and Duval, R. D. (1993). *Bootstrapping: A Non-Parametric Approach to Statistical Inference*. Newbury Park, CA: Sage Publications.

National Institute on Drug Abuse. (1990). *National household survey on drug abuse: Main findings 1988.* Rockville, MD: DHHS Publication No. 90-1682.

Patterson, B. H., Dayton, C. M., and Graubard, B. I. (2002). "Latent Class Analysis of complex sample survey data: Application to Dietary Data," *Journal of the American Statistical Association*, 97, 1-9.

Pfeffermann, D. (1993). "The role of sampling weights when modeling survey data," *International Statistical Review*, 61, 317-337.

Proctor, C. H. (1970). "A probabilistic formulation and statistical analysis of Guttman scaling," *Psychometrika*, 35, 73-78.

Rao, C. R. (1965). *Linear statistical inference and its applications*. New York: Wiley.

Read, T. R. C., and Cressie N. A. C. (1988) *Goodness-of-Fit Statistics for Discrete Multivariate Data*. New York: Springer-Verlag.

Rissanen, J. (1978). "Modeling by shortest data description," *Automatica*, 14, 465-471.

SAS User's Guide: Statistics. (1995) Cary, North Carolina: SAS Institute, Inc.

Sclove, S. (1987). "Application of model-selection criteria to some problems in multivariate analysis," *Psychometrika*, 52, 333-343.

Vermunt, J. K. (1997). "The LEM User Manual," WORC paper, Tilburg University, The Netherlands.

Vermunt, J. K., and Magidson, J. (2000). "Latent GOLD 2.0 User's Guide," Belmont, MA: Statistical Innovations, Inc.

Wolter, K. M. (1985). *Introduction to Variance Estimation.* New York: Springer-Verlag.

Yamaguchi, K., and Kandel D. B., (1984). "Patterns of drug use from adolescence to young adulthood: II. Consequences of progression," *American Journal of Public Health,* 74, 668-672.