4-22-2008

"Clustering Categorical Response" Application to Lung Cancer Problems in Living Scales

Ling Guo

“CLUSTERING CATEGORICAL RESPONSE”

APPLICATION TO LUNG CANCER PROBLEMS IN LIVING SCALES

by

Ling Guo

Under the Direction of Yu-Sheng Hsu, Jiawei Liu

ABSTRACT

The study aims to estimate the ability of different grouping techniques on categorical response. We try to find out how well do they work? Do they really find clusters when clusters exist? We use Cancer Problems in Living Scales from the ACS as our categorical data variables and lung cancer survivors as our studying group. Five methods of cluster analysis are examined for their accuracy in clustering on both real CPILS dataset and simulated data. The methods include hierarchical cluster analysis (Ward's method), model-based clustering of raw data, model-based clustering of the factors scores from a maximum likelihood factor analysis, model-based clustering of the predicted scores from independent factor analysis, and the method of latent class clustering. The results from each of the five methods are then compared to actual classifications. The performance of model-based clustering on raw data is poorer than that of the other methods and the latent class clustering method is most appropriate for the specific categorical data examined. These results are discussed and recommendations are made regarding future directions for cluster analysis research.

INDEX WORDS**:** Cluster analysis, Categorical data, CPILS, ACS, Lung Cancer Survivors, Factor analysis, Latent class clustering.

“CLUSTERING CATEGORICAL RESPONSE”

APPLICATION TO LUNG CANCER PROBLEMS IN LIVING SCALES

by

Ling Guo

A Thesis Submitted in Partial Fulfillment of Requirements for the Degree of

Master of Science

in the College of Arts and Sciences

Georgia State University

2008

“CLUSTERING CATEGORICAL RESPONSE”

APPLICATION TO LUNG CANCER PROBLEMS IN LIVING SCALES

by

Ling Guo

Committee Chair: Yu-Sheng Hsu
Jiawei Liu

Committee: Jeff Qin

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
May 2008

DEDICATION

To my parents

ACKNOWLEDGEMENTS

I would like to acknowledge my highly appreciation for the help and support from my committee: Dr. Yu-sheng Hsu, Dr. Jiawei Liu and Dr. Jeff Qin

With special thanks and appreciation to my committee chair Dr. Hsu for his encouragement and advice through my graduate study.

With special thanks and appreciation to my committee chair Dr. Liu who provided valuable suggestion and spent her time making correction through out the process.

This study was supported through a graduate internship from the American Cancer Society, National Home Office, Atlanta, GA

.

The support of Kenneth Portier, PhD, Director of Statistics in the Statistics and Evaluation Center of the ACS NHO is very much appreciated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Background

Clustering techniques have been developed to divide a large group of observations into smaller groups such that the observations within each group are relatively similar to each other and the observations in different groups are relatively dissimilar. Many different approaches to cluster analysis have been developed. Most clustering techniques can handle datasets that contain either numerical or categorical attributes. We are interested in the application of cluster analysis to categorical data, and specifically the data from a health questionnaire sent to cancer survivors.

In this paper, we use data from a survey of lung cancer survivors that measures quality of life, the Cancer Problems in Living Scale (CPILS). CPILS is a set of 31 survey statements designed to identify what cancer survivors are likely to experience following successful treatment for their cancers. Survivors answer these questions with one of three responses: not a problem for me, somewhat a problem for me, or a severe problem for me, which are then coded to 0, 1 and 2, respectively. Using these categorical data the goal is to group or cluster lung cancer survivors into similar response groups.

Several clustering methods are used to achieve this goal. A simulation study is needed to evaluate the performance of each method. Latent Class Analysis (LCA) is a statistical method to find subtypes of related cases (latent classes) from multivariate categorical data. We use a latent class model to create a simulated dataset that can be used to examine effectiveness of the

different clustering methods under differing conditions. Then the results are summarized and compared to the original groups. Finally, all of the cluster procedures are applied to the real quality of life (QOL) dataset.

1.2 Study objectives:

- To identify and describe 5 methods of factoring and clustering categorical data.
- To apply several selected clustering approaches on a real life dataset from American Cancer Society Study of Cancer Survivors
- To perform a simulation study to document the performance of the 5 methods and compare their performance in correctly classifying individuals using simulated test data.
- To learn how to compute latent factors and clusters in the statistical software package R.

1.3 Overview:

The remainder of this thesis is organized as follow: Chapter 2 is a detailed description on the study data. Chapter 3 describes statistical methods for clustering categorical data, and also presents an application to real QOL data and the corresponding implementation in R. Chapter 4 presents the simulation study that examines these methods applied to simulated data and discusses the results.

Chapter 2

The Data Set Overview

2.1 American Cancer Society-Study of Cancer Survivors (ACS-SCS) Study Goals

In this study we use data from an American Cancer Society survivorship survey. The main goal of the Study of Cancer Survivors (SCS) is to describe the needs and quality of life of survivors of the major of cancer types as they change over time in a large national population-based sample of survivors. SCS consists of two surveys, SCS-I and SCS-II. SCS-I was designed as a longitudinal study and SCS-II as a cross sectional study. SCS was designed to examine how behavioral, psychosocial, treatment, and support factors influence the quality of life of survivors. Finally, SCS was designed to help participating states to identify quality of life issues faced by cancer survivors in their state that may be different from those identified for the nation as a whole. Together, the information generated by this study is directed at supporting policy decisions at the American Cancer Society and of health care agencies as they work to improve the quality of life of cancer survivors.

This thesis analyzes data from the first round of SCS – I which targeted survivors of ten cancers from 11 states at approximately 12 months after diagnosis. The main goal of SCS-I is to assess quality of life for survivors of these 10 most commonly occurring cancers within the United States hence the cancer type is used as the primary stratification variable. It is known that the distribution of the ages of cancer survivors is skewed to the left with nearly 77% of all cancers being diagnosed to persons aged 55 or older (Cancer Facts and Figures, 2002).

To ensure a sufficient number of younger cancer survivors, age was also used as a stratification variable (with two levels: 18-54 years and 55+ years of age at diagnosis) with a disproportionately higher sampling of the younger age group. The incidence rates for the 10 target cancers are also known to vary across race and/or ethnicity demographic groups hence race/ethnicity is used as a stratification variable in states that has enough racial diversity to populate substrata formed by this additional stratification variable. Minority strata were over sampled. Finally, SCS-I is designed as a longitudinal study that seeks to follow cancer survivors over a 10 year period, therefore possible losses due to mortality has to be factored into the sample size of the initial survey. The number of survivors sampled for each cancer at the baseline or year 1 study is chosen to take into account the survival rate for each cancer over the expected 10 years of the study.

State cancer registry cases selected in each state for participation in SCS-I are first stratified by cancer type and then by age and race/ethnicity (wherever appropriate). The overall sample sizes are first allocated to each cancer type using fractions that are approximately inversely proportional to the survival rate associated with each cancer type. Next, the cancer specific samples are partitioned equally across the two age substrata. Note that equal allocation on age results in the younger survivors is over-sampled, that is, their sampling fraction is higher than their actual proportion in the population. Finally, for those states that have sufficient identifiable racial diversity among the population of cancer survivors, allocation is performed such that non-white cases are over sampled. This is done to ensure that sufficient non-white cases would be available for analysis. In general, an attempt is made to allocate one-third of the allocated sample size for each age group to non-white cases. It is necessary to modify the target allocation of the sample assigned to non-white cases for some cancer types within some states to

compensate for the variation in cancer case counts for minorities across states. Because of over sampling of younger cancer survivors and non-white cancer survivors, sampling weights are computed and used in the calculation of all statistics reported.

2.2 Sample Design

Survivors are identified for the SCS via cancer registries of states invited to participate in the national study. Selected survivors were given the opportunity to participate via either a mailed questionnaire or a telephone interview. Participants knew that all data collected would be used for research, and responses were strictly confidential. Study protocols for each participating registry were reviewed and approved by a human studies institutional review board, either by the review panel typically used by the state registry for all registry-associated research or by the Emory University Institutional Review Board at the request of researchers at the American Cancer Society National Home Office (ACS-NHO). In addition, the ACS Behavioral Research Center's (BRC) Advisory Committee reviewed and approved these protocols.

The SCS used a sampling design with stratification factors. The analytical sample used had 590 eligible survivors diagnosed with lung cancer. Lung cancer survivors are used since Lung cancer is the second most commonly diagnosed cancer in the United States, and the most common cause of cancer-related deaths for both men and women. Lung cancer survivors typically have significant QOL issues. According to "Cancer Facts & Figures 2007", an estimated 213,380 new cases are expected in 2007, accounting for about 15% of cancer diagnoses; an estimated 160,390 deaths, accounting for about 29% of all cancer deaths, are expected to occur in 2007.

To be eligible for inclusion in this study, individuals were required to meet the following eligibility criteria:

1. Be 18 years of age or older at the time of diagnosis.
2. Participants must have been diagnosed with lung cancer to achieve the desired time-since-diagnosis cohorts.
3. Have been diagnosed with stage I to IV cancer
4. Have been a state resident at the time of diagnosis with cancer.
5. Be alive at the time of initial contact.

2.3 Questionnaire Development

Quality of life is generally seen as a multi-dimensional concept. Among the dimensions typically considered important are the degree of psychological stress/distress, social/interpersonal functioning, physical health status and economic and financial status. The SCS questionnaires contained a selection of instruments and scales widely used in psychosocial research that have been shown to be valid and reliable for use with cancer patients. In addition, the researchers in the study developed other scales and items where no established instruments existed. The development of the SCS questionnaires involved three major activities: 1) consultation with a panel of medical and behavioral cancer researchers; 2) administration of the instrument to a focus group of cancer survivors representing both male and female cancer survivors with a wide range of cancer diagnoses, years since completion of treatment, and ethnic/racial backgrounds; and 3) pilot testing of the instrument with a sample of cancer survivors selected from a limited number of state cancer registries. A Spanish translation of the

questionnaire was prepared by a professional translation company. It was available on request and used by some Spanish speaking respondents. Full details of the questionnaire development process can be found in Smith et al., 2006.

2.4 Cancer Problems In Living Scale (CPILS)

Despite of the progresses in early detection and treatment, a significant portion of cancer survivors continuous to have levels of long term physical, emotional and social problems, sometimes years after treatment ends. Several quality of life (QOL) measurements have been developed and are widely applied. In this paper, findings for one of the more commonly used instruments are reported. The Cancer Problems in Living Scale (CPILS) (Baker et al., 2003) instrument is used to identify the extent of problems typically associated with cancer. CPILS is a set of 31 statements identifying problems that cancer survivors are likely to experience. For each problem, survivors were asked to indicate whether the statement represented 0 = Not a problem for me, 1 = somewhat a problem for me, or 2 = A severe problem for me. In this thesis, we use this measurement to study the important aspects of QOL such as social problems and financial/employment concerns for these lung cancer survivors.

2.5 Analysis Issues

The study design is stratified on cancer type, cohort and in many states on race/ethnicity. Because strata were not sampled in proportion to their population fraction, for example in many cases blacks and minorities were over sampled, responses may need to be weighted to obtain unbiased population estimate. These weights attempt to account for sampling of subgroups of the population in proportions different from their representation in the population and to account

for ineligibility of some individuals initially chosen in the sample. Sampling weights for the studies were computed using AAPOR (AAPOR, 2006) recommended methodology. While every effort was made to remove ineligible individuals prior to select the sample survivors, factors which could make selected survivors ineligible, such as being a non-English and non-Spanish speaker, were not always known by the cancer registry. The full details of the calculation of the sampling weights and associated sampling weight tables are available in Portier, et. al., 2007. Weights may not be important in identifying clusters. Weights may be important in defining latent factors, but their use for this task is beyond the current skills of this research. Hence, for this study sample weight is not used (communication with K Portier,2008).

2.6 Basic SCS –CPILS statistics for 1 year lung cancer survivors.

Table 1 presents top 12 most reported CPILS items and item distribution for weighted and unweighted calculation respectively. Table 2 shows the unweighted percentage of the frequency of the respondents answered each CPILS items.

Table 2.1. Top 12 most reported CPILS items and distributions for 1 year lung cancer survivors.

Cohort (1 Years since diagnosis)		weighted	unweighted
	N	590	
Item Distribution	N	422*	
0 Problem	%	2.96	2.61
1-2 Problems	%	8.72	5.69
3-6 Problems	%	20.89	19.19
7-12 Problems	%	30.22	27.96
>=13 Problems	%	37.21	44.55
1. Fatigue, loss of strength	N	562	
	%	85.0	84.3
2. Feeling fearful illness will return	N	566	
	%	72.5	77.0
3. Concern about relapsing	N	552	
	%	62.2	69.2
4. Fears about the future	N	569	
	%	58.4	65.2
5. Sleep difficulties	N	571	
	%	55.6	60.6
6. Continued major problems with health	N	563	
	%	53.5	56.0
7. Difficulty making long term plans	N	568	
	%	52.5	55.8
8. Less physically able to have sex	N	525	
	%	48.9	49.71
9. Preoccupation with being ill	N	563	
	%	44.7	49.73
10. Feeling dependent	N	564	
	%	43.4	43.6
11. Uncomfortable w/ changes in physical appearance	N	566	
	%	43.0	48.2
12. Diminished ability to concentrate	N	563	
	%	42.7	44.2

* all individuals with the missing values are deleted.

Table 2.2: Description of CPILS items for 1 year cohort lung cancer survivors.
(unweighted percent of respondents)

Item	Description	Missing value (%)	Not a Problem (%)	Somewhat of a Problem (%)	A Severe Problem (%)
CPILS_A	Not being able to change jobs for fear of losing my health insurance coverage.	10	73.9	7.97	8.14
CPILS_B	Job discrimination.	11.02	82.03	4.24	2.71
CPILS_C	Concern about relapsing.	6.44	28.81	50.17	14.58
CPILS_D	Fatigue, loss of strength.	4.75	14.92	60.17	20.17
CPILS_E	Uncomfortable with changes in my physical appearance.	4.07	49.66	38.31	7.97
CPILS_F	Preoccupation with ill.	4.58	47.97	40.17	7.29
CPILS_G	Eating difficulties.	3.90	62.20	27.12	6.78
CPILS_H	Concern about being physically unable to have children.	9.15	87.46	1.53	1.86
CPILS_I	Diminished ability to concentrate.	4.58	53.22	37.63	4.58
CPILS_J	Sleep difficulties.	3.22	38.14	46.27	12.37
CPILS_K	Feeling dependent.	4.41	53.90	33.73	7.97
CPILS_L	Less physically able to have sexual intercourse.	11.02	44.75	26.10	18.14
CPILS_M	Fear about the future.	3.56	33.56	47.80	15.08
CPILS_N	Guilt feelings.	4.41	65.25	24.41	5.93
CPILS_O	Feeling angry.	4.07	58.81	29.66	7.46
CPILS_P	Having difficulties in making long-term plans.	3.73	42.54	40.17	13.56
CPILS_Q	Feeling isolated.	4.24	64.75	24.75	6.27
CPILS_R	Feeling helpless.	3.56	56.78	30.85	8.81
CPILS_S	Feeling vulnerable.	4.92	53.39	34.24	7.46
CPILS_T	Being treated as different from others.	3.39	77.97	16.61	2.03
CPILS_U	Concerned about infection and crowd.	3.56	59.83	29.83	6.78
CPILS_V	Problems with family/children.	3.39	82.54	10.17	3.90
CPILS_W	Difficulty in returning to former roles.	5.43	57.29	27.97	9.32
CPILS_X	Problem communicating with spouse or partner.	10.17	68.98	15.76	5.08
CPILS_Y	Difficulty in meeting my medical expenses.	4.41	62.20	24.58	8.81
CPILS_Z	Feeling fearful that my illness will return.	4.07	22.03	50.85	23.05
CPILS_AA	Being less able to provide for the financial needs of my family	7.97	55.93	23.73	12.37
CPILS_BB	Difficulty in obtaining adequate insurance.	5.25	74.24	12.37	8.14
CPILS_CC	Difficulties in pursuing the career of my choice.	8.47	68.47	11.86	11.19
CPILS_DD	Continued major problems with my health.	4.58	42.03	39.83	13.56
CPILS_EE	Not able to get the information I need about cancer.	4.41	81.69	11.36	2.54

Chapter 3

Clustering Methods

3.1. Distance Clustering (Ward's Method)

3.1.1 General approach to hierarchical clustering

An important component of a clustering algorithm is the distance measure between data points. Hierarchical cluster analysis is one approach to clustering that uses a basic distance measure to systematically group observations. Initially, each individual/object is assigned to its own cluster, so that if we have n objects, we start with n clusters and each cluster contains just one object. We then find the closest (most similar) pair of clusters and merge them into a new group, as a result having one cluster less to begin the next iteration. The program then re-computes distances (similarities) between the new cluster and each of the remaining old clusters. The process is repeated until a certain stopping criterion is met, (typically when all objects are clustered into a single cluster of size *n*).

3.1.2 Distance measures typically used in hierarchical clustering

Inter-object similarity is measured by distance between pairs of objects. The typical ways of computing distances between objects in a multi-dimensional space is via *Euclidean distances*, *Manhattan distance*, or *Mahalanobis Distance*. Let $X_i = (x_{i1}, x_{i2}, \cdots x_{iH})', X_j = (x_{j1}, x_{j2}, \cdots x_{jH})'$ be two *H* dimensional sample points, and let $\hat{\Sigma}^{-1} = \hat{cor}(X)$ denote the inverse of the sample

covariance matrix (assuming it exists).

Euclidean distances $D_e(\mathrm{X}_i, X_j) = \sqrt{\sum_{k=1}^{H}(x_{ik} - x_{jk})^2} = \sqrt{(X_i - X_j)'(X_i - X_j)}$

Manhattan distance $D(\mathrm{X}_i, X_j) = \sum_{k=1}^{H} | x_{ik} - x_{jk} |$

Mahalanobis distance $D(X_i, X_j) = \sqrt{(X_i - X_j)' \hat{\Sigma}^{-1} (X_i - X_j)}$

For hierarchical clustering of categorical data we use the *Manhattan distance metric*. The distance between objects whose x vector represents categorical responses is typically measured using the Manhattan metric**.** This distance is simply the sum of the absolute difference across dimensions.

3.1.3 Minimum Variance Clustering (Ward's method)

Ward's method, also known as *Minimum Variance clustering* uses an analysis of variance based metric to evaluate the distances between clusters in a hierarchical clustering algorithm, This method attempts to minimize the ANOVA Sum of Squares (SS) of any two clusters that can be formed at each step. The distance between two elements (individuals or previously defined clusters) is defined by $D(C_{K,} C_L) = \frac{\left\| \overline{X}_K - \overline{X}_L \right\|^2}{\frac{1}{n_K} + \frac{1}{n_L}}$. Where n_K, number of observations in clusters C_K and $\overline{X}_K = \{\overline{x}_{k1,} \overline{x}_{k2,} ... \overline{x}_{kH}\}$, is the mean vector for observations belonging to the cluster C_K. Cluster C_L is defined similarly. A cluster can consist of one point, say point $Y = \{y_1, y_2, ... y_H\}$ and the distance from this point to cluster C_K for example is

$D(Y, C_K) = \dfrac{\left\| Y - \bar{X}_K \right\|^2}{1 + \dfrac{1}{n_K}}$. If we combine two previously defined clusters, say clusters K and L into a new cluster $C_M = C_K \cup C_L$ then the distance $D(C_J, C_M)$ between the cluster C_J and C_M is given by the combinatorial formula defined as the flexible-beta approach proposed by Lance and Williams (1967):

$$D(C_J, C_M) = \frac{(n_J + n_K)}{(n_K + n_M)}\left[D(C_J, C_K) + D(C_J, C_L)\right] - \frac{n_J}{n_J + n_M} D(C_K, C_L).$$

In general, Ward's method merges clusters that maximize the multivariate normal classification likelihood assuming each level of the hierarchy has the same covariance matrices and equal sampling probabilities. It is regarded as very efficient; however, it is very sensitive to outliers.

3.1.4 Minimum Variance Clustering Applied to SCS Data

We use the hclust package in the R language (R-project, see http://www.r-project.org/) and the following commands to perform a hierarchical cluster analysis using Wards method on the CPILS data.

```
D<- dist(cpils.df,method="manhattan")
clust.ward<-hclust(D, method="ward")
```

where cpils.df is a data frame (matrix) of all coded variable responses for all Lung cancer survivor respondents to the SCS I. Please refer to Appendix I for a fuller implementation of R program for this analysis.

When Ward's method is applied to the CPILS data from the SCS, the dendrogram resulting from the hierarchical clustering process is as given in Figure 3.1. Note there are two clusters that are quite far from each other and that one of these clusters can be further divided into two, resulting on three clearly separated groups.

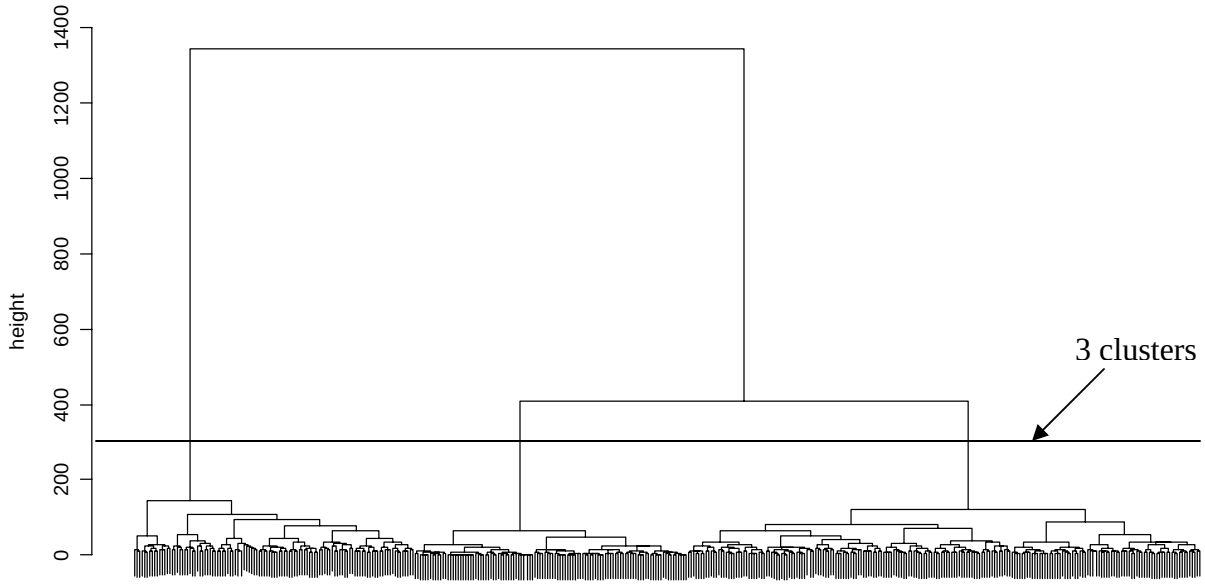


Figure 3.1 Hierarchical Cluster Dendrogrom Applied to CPILS Data.

3.2. Model-based clustering

3.2.1 General approach to Model-based clustering

In model-based clustering, we assume the data come from a mixture of multivariate normal or Gaussian distributions. Each component probability distribution corresponds to one of the clusters. A specific cluster in this model is often referred to as a component distribution. The entire data set is modeled by a mixture of several distributions. Models that differ in members of components or in component distributions can be compared. Outliers are handled by adding one

or more components representing a different distribution for outlying data. Parameters for the component distribution are estimated using likelihood techniques.

Suppose the model is a mixture of *G* components (clusters). Each component is assumed to follow a multivariate Gaussian distribution parameterized by a mean vector μ_k, and covariance matrix Σ_k. Denote the data by $X = (x_1, x_2, \cdots, x_H)'$, X is normal matrix and assume the mixture has *G* components. Although our data is discrete 0,1,2. We show this method just for comparison purpose.

The density of mixture component k is assumed to be multivariate normal:

$$\phi_k(X \mid \mu_k, \Sigma_k) = (2\pi)^{-\frac{H}{2}} |\Sigma_k|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(X_i - \mu_k)'\Sigma_k^{-1}(X_i - \mu_k)\right\}.$$

If τ_k is the probability that an observation belongs to the kth component ($\tau_k \geq 0; \Sigma_{k=1}^{G}\tau_k = 1$), the likelihood of the full data can be written as the follow mixture:

$$L(X) = \prod_{i=1}^{n}\sum_{k=1}^{G}\tau_k\phi_k(X_i \mid \mu_k, \Sigma_k)$$

Maximum likelihood estimators of the parameters can be computed by using the EM algorithm (ref:[8]). In the models considered here, an iteration of EM consists of an E-step followed by an M-step. In the E-step, a matrix ***z*** is computed such that z_{ik} is an estimate of the conditional probability that observation *i* belongs to group k given the current parameter estimates. In the M-step parameter estimates that maximize the expected log-likelihood from given ***z*** are computed. The algorithm proceeds as follows.

1. Initialize parameters. (means, covariances, and mixing proportions)

   Essentially each object is randomly assigned to a group or start with the results of a hierarchical clustering.

$$z_{ik} = \begin{cases} 1 & \text{if } X_i \text{ belongs to group k} \\ 0 & \text{Otherwise} \end{cases}, \quad \text{where } z_i = \left(z_{i1}, z_{i2}, \cdots, z_{iG}\right)$$

Assume that each $\boldsymbol{z}_i$ is iid multinomial having probabilities

$\tau_1, ..., \tau_G$ drawn from *G* groups. The log-likelihood is:

$$\ell(X) = \sum_{i=1}^{n} \log \sum_{k=1}^{G} z_{ik} \left[\tau_k \, f\left(X_i \mid \mu_k, \Sigma_k\right)\right]$$

2. E-step: Estimate the conditional probabilities z_{ik} for all *i* =1, ..., *n*, *k*= 1, ...,*G* assuming the distribution parameters are fixed at their current values.

$$\hat{z}_{ik} = \frac{\hat{\tau}_k \, f_k(X_i \mid \hat{\mu}_{k,} \hat{\Sigma}_k)}{\sum_{j=1}^{G} \hat{\tau}_j \, f_j(X_i \mid \hat{\mu}_j, \hat{\Sigma}_j)}$$

3. M-step: Compute the MLE for each distributional parameters assuming the conditional probabilities are fixed at the values obtained in the previous E-step.

$$\hat{\tau}_k = \frac{\sum_{i=1}^{n} \hat{z}_{ik}}{n}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{n} \hat{z}_{ik} x_i}{\sum_{i=1}^{n} \hat{z}_{ik}}$$

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^{n} \hat{z}_{ik} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)'}{\sum_{i=1}^{n} \hat{z}_{ik}}$$

4. Repeat step 2 and 3 until convergence. The convergence criterion within the MCLUST package in R is specified as a relative convergence tolerance for the log-likelihood (1×10^{-5}) and for parameter estimates convergence, respectively.

The components or clusters in these models have ellipsoidal confidence regions centered at the means μ_k and the covariance matrix Σ_k determines other geometric features. Specification of Σ_k is supported by assuming it can be represented by its eigenvalue decomposition in the form $\Sigma_k = \lambda_k D_k A_k D_k'$, where λ_k is a scalar, D_k is the orthogonal matrix of eigenvectors, and A_k is a diagonal matrix whose elements are proportional to the eigenvalues of Σ_k. The orientation of the density contours of the component distributions and also the principal components of Σ_k is determined by D_k, while A_k determines the shape of the density contours and λ_k specifies the volume of the corresponding density contours, which is proportional to $\lambda_k^d |A_k|$, where d is the data dimension. The covariance structures defining the models are summarized in table 3.1. When the model is multivariate normal with an equal-volume spherical covariance λI, the selection criterion is equivalent the previously discussed Ward's method.

Table 3.1: Parameterizations of Σ_k currently available in MCLUST for multidimensional data.

identifier	Model	Distribution	Volume	Shape	Orientation
EII	λI	Spherical	equal	equal	NA*
VII	$\lambda_k I$	Spherical	variable	equal	NA*
EEI	λA	Diagonal	equal	equal	coordinate axes
VEI	$\lambda_k A$	Diagonal	variable	equal	coordinate axes
EVI	λA_k	Diagonal	equal	variable	coordinate axes
VVI	$\lambda_k A_k$	Diagonal	variable	variable	coordinate axes
EEE	λDAD^T	Ellipsoidal	equal	equal	equal
EEV	$\lambda D_k A D_k^T$	Ellipsoidal	equal	equal	variable
VEV	$\lambda_k D_k A D_k^T$	Ellipsoidal	variable	equal	variable
VVV	$\lambda_k D_k A_k D_k^T$	Ellipsoidal	variable	variable	variable

* Spherical shape has no orientation.

The Bayesian Information Criterion (BIC) is used to decide on the optimal clustering model (parameterization and number of cluster). The BIC has the form BIC=2* $\log L(m)$ -(npar)*log(n), where $\log L(m)$ is the maximized log-likelihood for the model and data, npar represents the number of parameters to be estimated in the model, and n is the number of observations in the data. In general the larger the value of the BIC is, the stronger the evidence for the model and number of clusters.

According to the description above the strategy for clustering based on mixture models is to fit each of the models presented in Table 3.1 for a range of numbers of clusters and compute each model's BIC. Select the best model from those having the largest BIC.

3.2.2 Application of model-based clustering to CPILS data from the SCS

MCLUST is a R package for normal mixture modeling and model-based clustering. It provides functions for parameter estimation via EM for the models given in Table 3.1. The following commands produce the clustering results of the CPILS dataset:

```
clust.raw<-Mclust(cpils.df)
summary<-summary(mclustBIC(cpils.df), data=cpils.df)
plot(clust.raw)
table(clust.raw$classification)
```

The implementation of R program in MCLUST package, please refer to Appendix II.

The general model-based clustering approach is applied to the CPILS data from the SCS. All models showed in Figure 3.2 were fit to the data assuming 1 to 9 clusters (a total of 46 model fits). Using the BIC, the top 3 models were shown in Table 3.2. According to our protocol, we chose as the best model the "VEI" form with 3 components.

Table 3.2 The Results of Mclust Applied to Raw Data

VEI model, 3 clusters			the best BIC values:		
1	2	3	VEI,3	VII,3	VEI,2
88	201	133	-17506.4	-18976	-19209

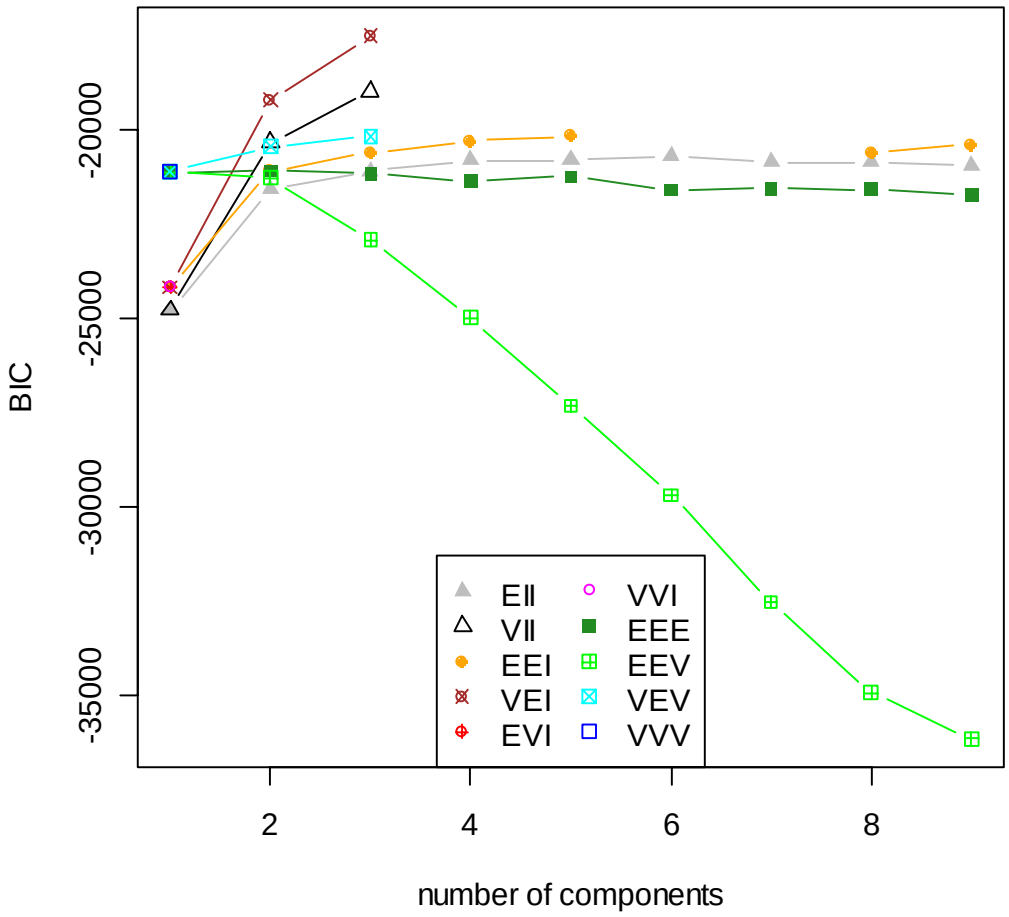


Figure 3.2: BIC plot for raw data of CPILS dataset.

3.3 Clustering on Latent factors

3.3.1 Latent factors approaches/methods

Models in which the orientation is allowed to vary between clusters (EEV, VEV, EVV, VVV), have approximately d^2 parameters per cluster, where d is the number of variables (dimensions). For this reason, the model-based clustering algorithm for large d may not work well or may otherwise be inefficient. It may still be possible to analyze such data with model-based clustering by restricting the models to fewer dimensions resulting in fewer parameters by

first applying a dimension-reduction technique as well as restricting the component shape characteristics (e.g. spherical or diagonal models). Our next approach is to first apply standard factor analysis or perform an independent factor analysis in order to extract latent factors prior to applying the model-based cluster approach.

Factor analysis is widely used as an exploratory tool to reduce the dimensionality of multivariate data. The underlying assumption of factor analysis is that there exist a number of unobserved latent factors that account for the correlations among observed variables. Latent factors typically are computed as linear function or weighted sum of the original variables. Via the Central Limited Theorem, the resulting factor scores should have more normal-like distribution than the original variable values, even in the case where the original variables are categorical. The latent factors as a result should have properties that more closely match the requirements of the clustering methods. For this reason we examine approaches that first find latent factors and then apply clustering to the latent factor scores. We selected two factor analytic techniques: Maximum-likelihood estimation of factor analysis (MLE-FA) and independent factor analysis for use in this study.

3.3.2 Application of MLE-FA to CPILS data from SCS

Maximum likelihood factor analysis is based on a linear combination of variables to form factors, when normality is assumed with large sample sizes. Using the MLE method, the linear combination weights (parameters) are estimated by finding the values those most likely to have resulted in the observed correlation matrix. One nice characteristic of this approach is that MLF generates a chi-square goodness-of-fit test. We can increase the number of factors one at a time until a satisfactory goodness of fit is obtained.

There are many criteria for determining the number of factors. Typically we use one or more of the methods, determine an appropriate range of numbers of factors to investigate, and then select the solution which generates the most comprehensible factor structure. In this study we used the Kiser rule (ref: [10]) and scree plot. The Kiser rule is to drop all components with eigenvalues under 1.0. The scree plot is used to identify where adding further factors results in only marginally additional explanation of total variability (i.e. the point of inflection in the plot). The scree plot (figure 3.3) suggests use of 2 or 3 factors. The "eigenvalues greater than one" rule suggests a maximum of 6 factors (see table 3.3). After comparing fits of different number of factors, we chose 2 factors for use in further analyses since it also with the number of factors chose by the independent factor analysis model which we discuss later.

We used the factanal function in the stats package in R and the following commands to fit a two factor model using maximum likelihood methodology.

```
cpils.fa<-factanal(cpils.df, factors=2,rotation="promax",
scores="regression")
```

In this model, we chose to rotate the factors using a “promax” procedure. This procedure performs an oblique rotation. Factor scores were estimated using the “regression” or Thompson's methodology (ref: [16]). When oblique rotation is used the resulting factors are correlated, so the final factor correlation matrix is not diagonal (see figure 3.4)

Table 3.3: Eigenvalues of the CPILS Correlation Matrix

Number	Eigenvalue	Difference	Proportion	Cumulative
1	14.594	12.276	0.471	0.471
2	2.318	0.655	0.075	0.546
3	1.662	0.340	0.054	0.599
4	1.322	0.169	0.043	0.642
5	1.153	0.100	0.037	0.679
6	1.053	0.150	0.034	0.713
7	0.903	0.058	0.029	0.742
8	0.845	0.079	0.027	0.769
9	0.766	0.112	0.025	0.794
10	0.655	0.042	0.021	0.815
11	0.612	0.022	0.020	0.835
12	0.591	0.018	0.019	0.854
13	0.573	0.083	0.018	0.872
14	0.490	0.014	0.016	0.888
15	0.475	0.041	0.015	0.904
16	0.434	0.050	0.014	0.918
17	0.384	0.014	0.012	0.930
18	0.369	0.063	0.012	0.942
19	0.306	0.020	0.010	0.952
20	0.286	0.038	0.009	0.961
21	0.248	0.027	0.008	0.969
22	0.221	0.026	0.007	0.976
23	0.195	0.038	0.006	0.982
24	0.157	0.023	0.005	0.987
25	0.134	0.009	0.004	0.992
26	0.125	0.006	0.004	0.996
27	0.119	0.037	0.004	1.000
28	0.082	0.033	0.003	1.002
29	0.049	0.026	0.002	1.004
30	0.024	0.169	0.001	1.005
31	-0.145	-	-0.005	1.000

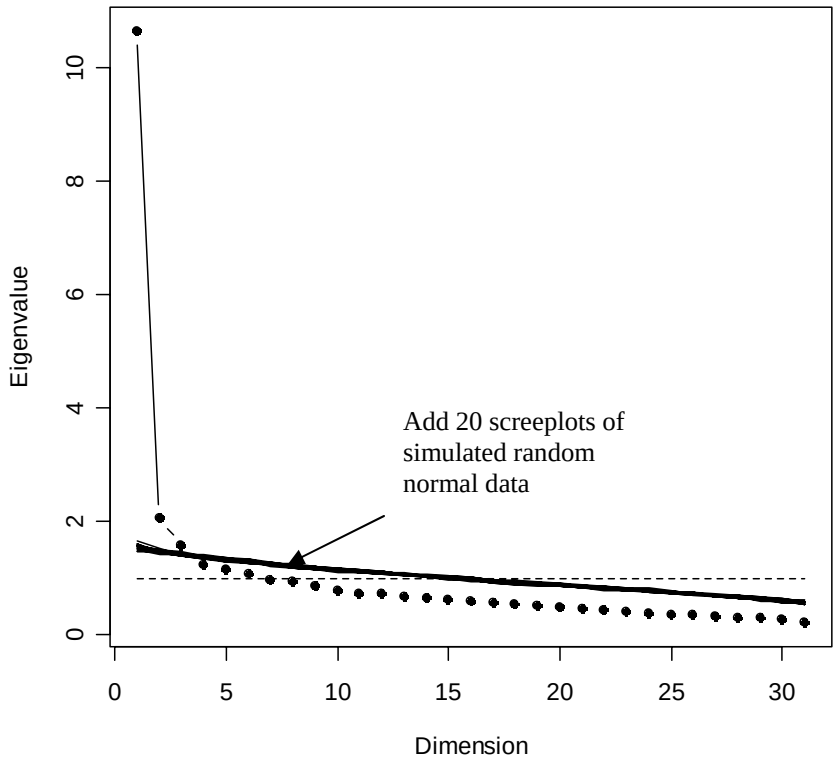


Figure 3.3. Scree Plot

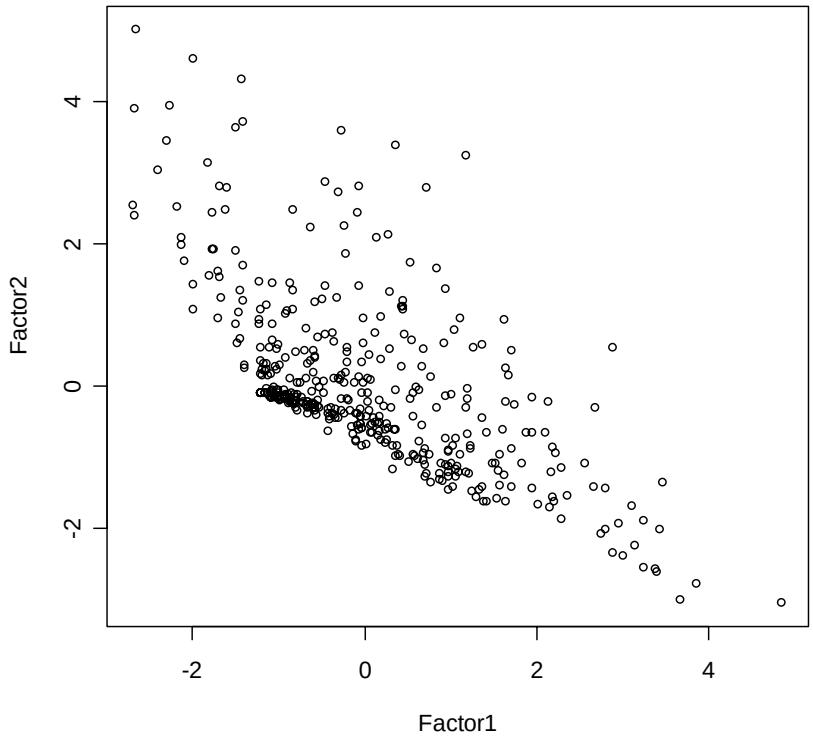


Figure 3.4: plot of factor scores of CPILS data

3.3.3 Model_based clustering of latent factor scores from MLE-FA

The MCLUST method as implemented in R and discussed previously was also applied to the rotated latent factor scores from standard factor analysis. The following commands produce the clustering results using the latent factor scores:

```
clust.fa<-Mclust(cpils.fa$scores, G=3)
plot(clust.fa)
table(clust.fa$classification)
```

The implementation of R program in MCLUST package, please refer to Appendix III.

Figure 3.5 plots BIC of all models that were fit to the data by number of components which ranged from 1 to 9. We can see that a best model best is "VEV" with 4 components and a BIC of -2011.168. In order to have consistency with other methods to be discussed in this study, we chose instead to use the slightly poorer fitting 3 components model "VVV" with BIC of -2045.994.

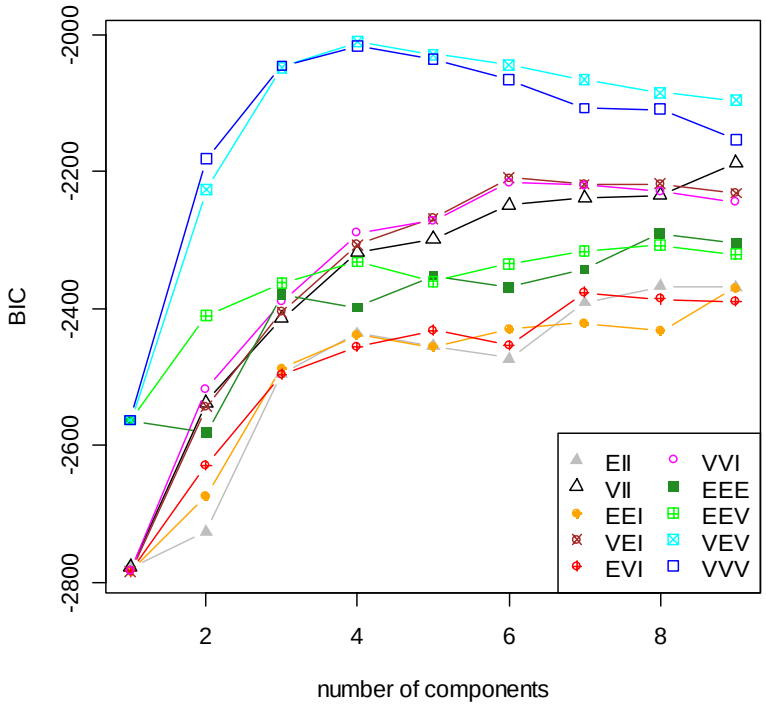


Figure 3.5: BIC plot for CPILS dataset (2 factor scores)

. Figure 3.6 shows the density estimation after applying the clustering functions to fit a model to the CPILS data.

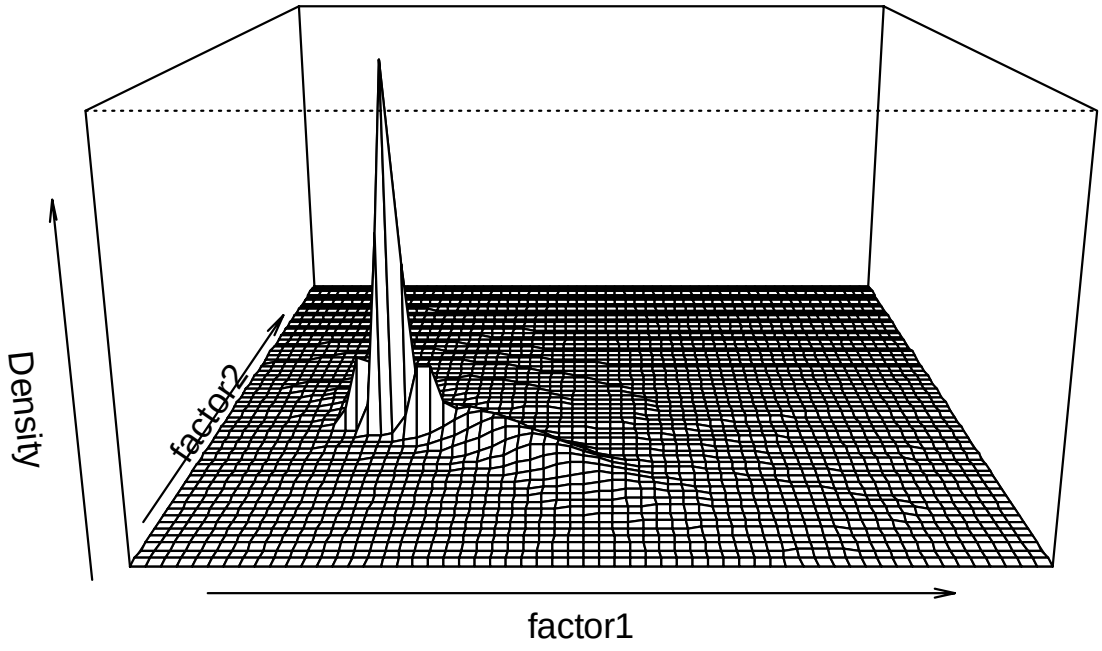


Figure 3.6 Perspective plot of density estimate for CPILS dataset (factor scores)

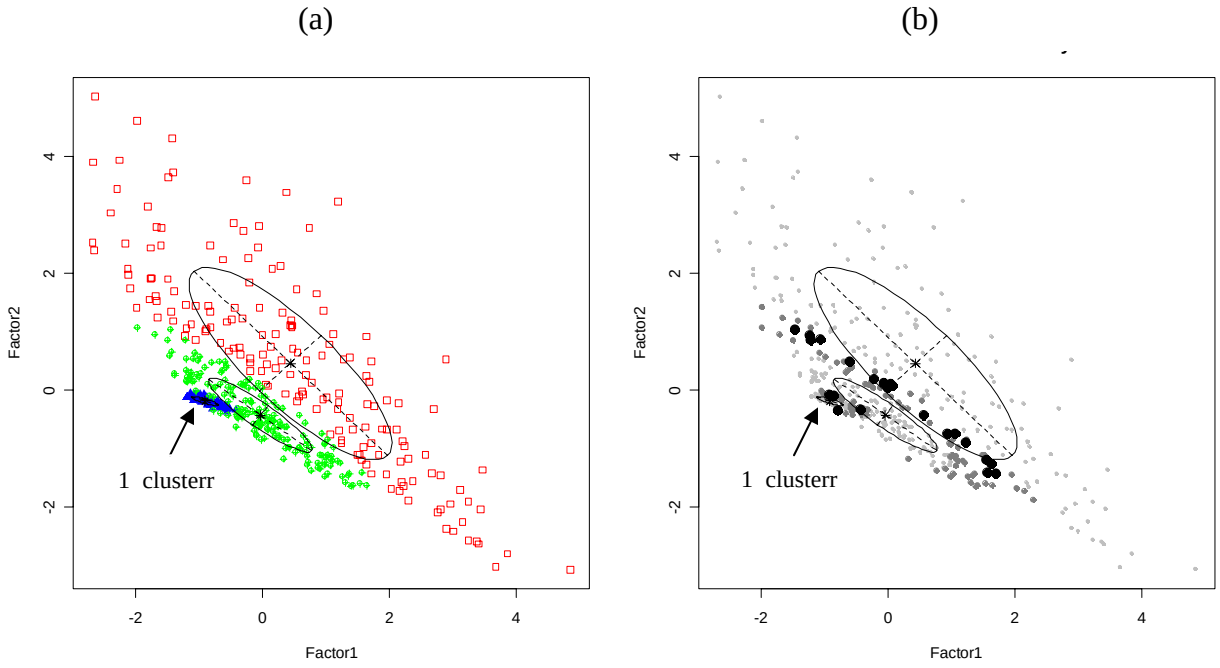


Figure 3.7 Classification (a) and classification uncertainty (b) plots for 2 factor scores of CPILS dataset.

The ellipses shown in Figure 3.7 define equal density regions and illustrate how the component covariance matrices have different size, shape and orientation. In the classification plot, points in different classes are indicated by different symbols. In the uncertainty plot, the symbols have the following meanings: large filled symbols for 95% quantile of uncertainty; smaller open symbols 75-95% quantile; small dots, first three quantiles of uncertainty.

3.3.4. Independent factor analysis applied to SCS data

Independent Factor Analysis (IFA) is a new procedure that does not have a lot of associated research literature or use in practice. IFA assumes a latent factor structure that closely resembles that of the ordinary un-rotated factor analysis model but which assumes that the latent variables are mutually independent and that the density of each latent variable is modeled by a mixture of Gaussians distributions. The model is complex and underlying density parameters are estimated by a EM algorithm. The p-dimensional observed variables X are modeled in terms of a smaller set of k unobserved independent latent variables, Y, and an additive specific term u via a linear function. That is $X=HY+u$, where u is assumed to be normally distributed with diagonal variance matrix Ψ. The factor loading matrix H is also referred to as the mixing matrix.

The general methodology of IFA has been incorporated into the ifa library in R. Use of the methodology requires first that we choose an IFA model form. A function ifa.em fits a specified IFA model by the EM algorithm, and ifa.BIC computes the model-associated Bayesian Information Criterion (BIC) defined previously. According to the formula $BIC=2*\log L(m)$ - $(npar)*log(n)$, each model is run 10 times and model BIC values compared for consistency of fit. We chose a model with two factors, one factor having three mixture components and the other two mixture components (the c(3,2) model) as having the best fit to the CPILS data according

to the average largest BIC value criterion. Table 3.4 shows the BIC and maximum likelihood value (LIK) for each model. After a best fitted model is chosen, the function ifa.predict is used to compute the predicted latent variables also referred to as the "independent factor scores". (Please refer to Appendix III for R implementation)

3.3.5. Model_based clustering of independent factor analysis scores

The clustering results can be displayed as follows:

best BIC values:			classification table:		
VEV,4	VII,3	VEV,3	1	2	3
-1204.733	-1213.640	-1214.534	99	160	163

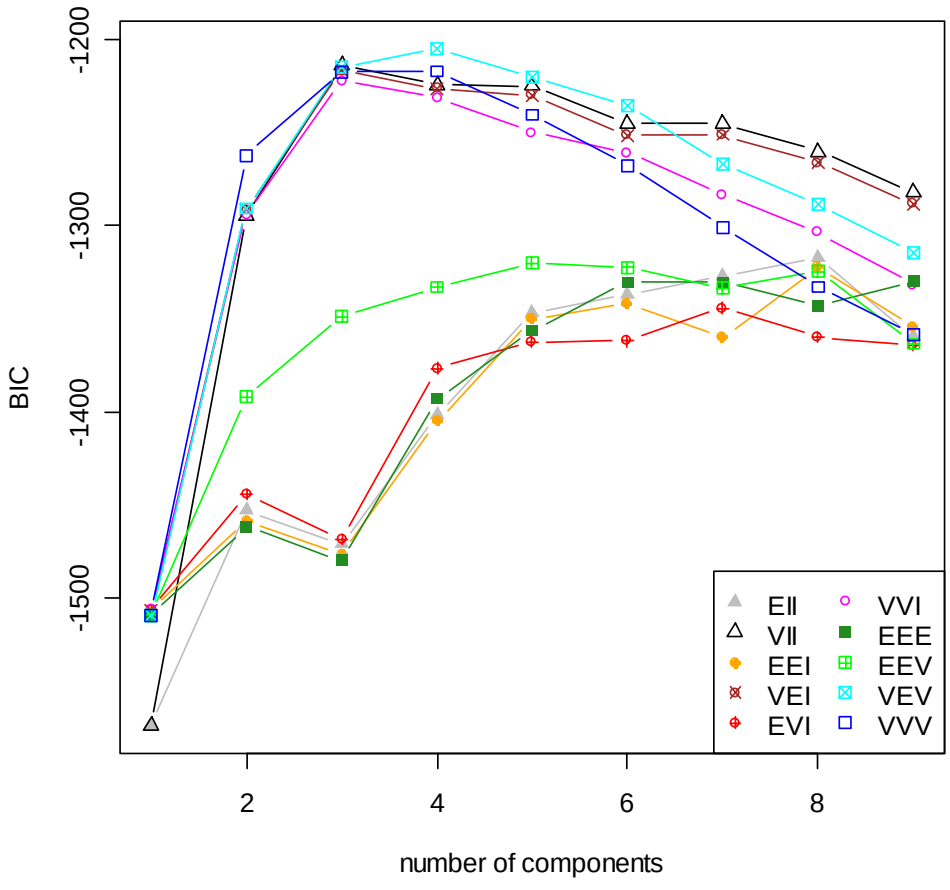


Figure 3.8 BIC plot of CPILS dataset (independent factor scores)

In this case, the best model is “VEV” with 4 components and BIC of -1204.733. Here we picked the second best model 3 components model “VII” with BIC of -1213.640. The spheres shown in Figure 3.10 are the component covariance matrices of different size, same shape and no orientation.

Table 3.4 .Bayesian Information Criterion (BIC) and log-likelihood (LIK) for ifa models

model		1	2	3	4	5	6	7	8	9	10
c(1,1)	BIC	-37148	-37132	-37140	-37138	-37132	-37146	-37149	-37132	-37133	-37140
	LIK	-18293	-18285	-18289	-18288	-18285	-18292	-18293	-18285	-18285	-18289
c(1,2)	BIC	-37140	-37149	-37158	-37157	-37158	-37141	-37151	-37157	-37146	-37165
	LIK	-18280	-18284	-18289	-18288	-18289	-18280	-18285	-18288	-18283	-18292
c(2,2)	BIC	-36808	-36886	-36744	-36842	-37186	-36786	-36751	-36979	-37183	-37163
	LIK	-18105	-18144	-18073	-18122	-18294	-18094	-18076	-18190	-18292	-18283
c(3.1)	BIC	-37185	-36899	-37154	-37177	-37168	-37178	-37164	-37015	-36881	-37175
	LIK	-18293	-18150	-18278	-18289	-18285	-18290	-18283	-18209	-18141	-18288
c(3,2)	BIC	-36825	-36853	-36807	-37192	-36783	-36806	-36824	-36152	-36719	-35402
	LIK	-18104	-18118	-18095	-18288	-18083	-18095	-18104	-17768	-18051	-17393
c(3,3)	BIC	-37221	-36917	-37216	-37217	-36834	-37220	-36891	-37214	-37215	-37219
	LIK	-18293	-18141	-18291	-18291	-18099	-18293	-18128	-18290	-18290	-18292
c(1,4)	BIC	-37201	-36870	-36899	-36909	-37202	-37187	-37196	-37190	-37180	-37179
	LIK	-18292	-18127	-18141	-18146	-18293	-18285	-18290	-18287	-18282	-18281
c(2,4)	BIC	-37219	-37191	-37206	-36820	-37211	-37201	-36880	-37000	-37204	-37201
	LIK	-18292	-18278	-18285	-18092	-18288	-18283	-18123	-18183	-18285	-18283
c(3,4)	BIC	-36877	-37232	-37230	-37227	-37231	-37225	-36853	-37227	-36801	-37236
	LIK	-18112	-18290	-18289	-18287	-18289	-18286	-18100	-18287	-18074	-18292
c(1,1,1)	BIC	-37332	-37333	-37320	-37336	-37336	-37335	-37335	-37333	-37335	-37336
	LIK	-18291	-18292	-18285	-18293	-18293	-18293	-18293	-18292	-18293	-18293
c(1,1,2)	BIC	-37353	-37349	-37185	-37062	-37338	-37345	-37335	-37335	-37336	-37353
	LIK	-18293	-18291	-18209	-18147	-18285	-18288	-18284	-18284	-18284	-18293
c(2,2,2)	BIC	-37386	-37393	-36910	-36798	-37384	-37385	-36775	-37385	-37389	-37383
	LIK	-18291	-18294	-18053	-17997	-18290	-18291	-17986	-18291	-18293	-18290
c(3,2,2)	BIC	-37402	-37404	-37402	-37402	-37036	-37297	-37033	-36818	-37402	-37408
	LIK	-18290	-18291	-18290	-18290	-18107	-18238	-18106	-17998	-18290	-18293
c(1,1,1,1)	BIC	-37520	-37519	-37522	-37523	-37522	-37512	-37522	-37522	-37520	-37522
	LIK	-18291	-18291	-18293	-18293	-18292	-18287	-18292	-18292	-18292	-18292

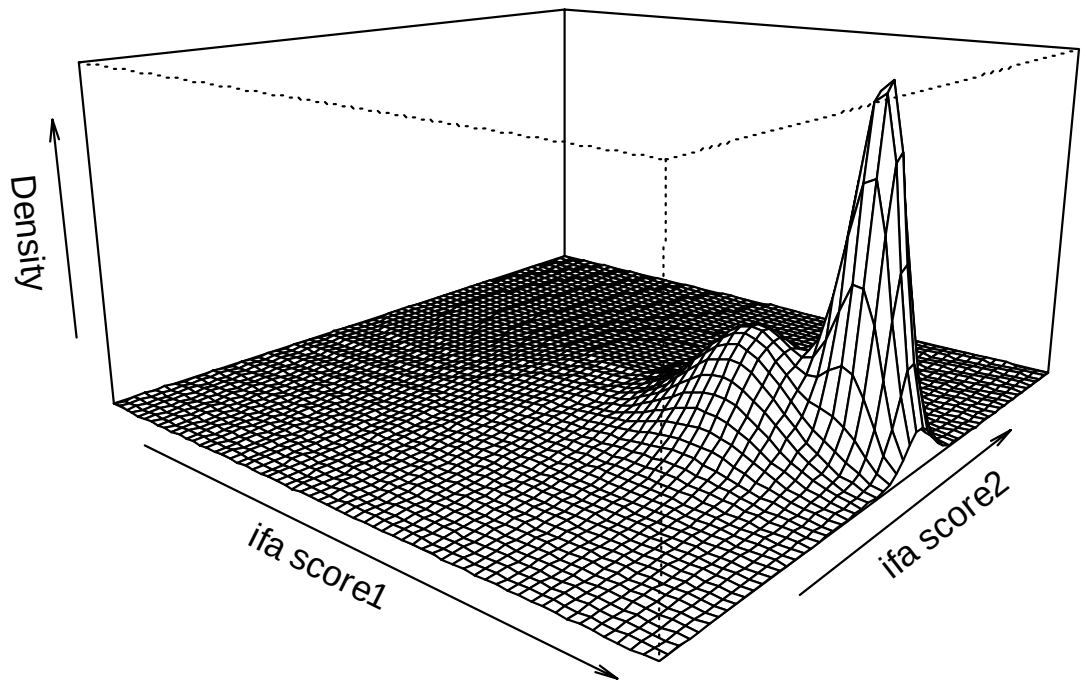


Figure 3.9 Perspective plot of density estimate for CPILS dataset (independent factor scores)

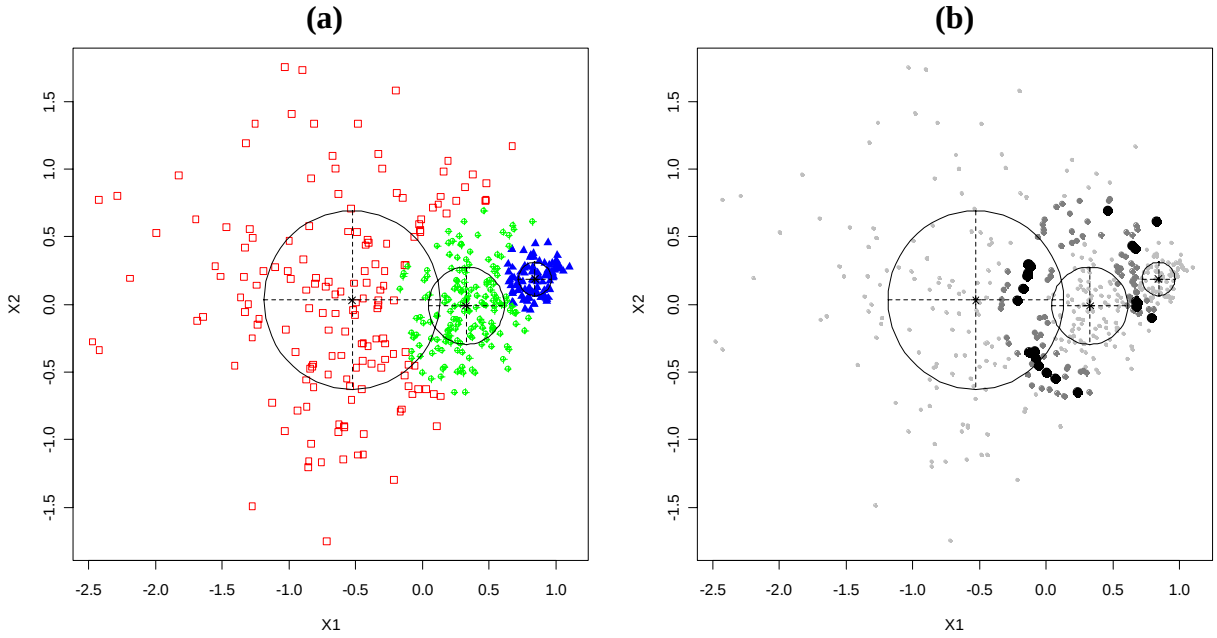


Figure 3.10 (a) Classification and (b) Uncertainty plots for independent factor scores of CPILS dataset.

3.4 Latent Class Cluster analysis

3.4.1 General approaching

Latent Class Analysis (LCA) is another relatively new statistical approach for identifying unmeasured class membership from multivariate categorical response. The model identified classes are called “latent” because a case's underlying class membership is not directly observed. LCA does not assume linearity of latent structure, normality of response data, or homogeneity of variances. The basic LCA model assumes that the distribution of responses for each observed variable can be represented by a finite mixture of the mutually independent latent component class response distributions.

Suppose we have *J* categorical variables (the “manifest” variables), each of which contain K_j possible outcomes, for $i = 1...n$ individuals. The manifest variables may have different numbers of outcomes. But in the case of CPILS, all have the same number (3) possible outcomes. Let Y_{ijk} represent the observed values of the *J* manifest variables such that Y_{ijk} = 1 if individual *i* give the *k*th response to the *j*th variable, arnd Y_{ijk} = 0 otherwise, where *j* = 1...J and *k* = 1… K_j The latent class model approximates the observed joint distribution of the manifest variables as the weighted sum of a finite number, *R*, of constituent cross-classification tables. Let π_{jrk} denote the class-conditional probability that an observation in class *r* = 1…*R* produces the *k*th outcome on the *j*th variable. Within each class, for each manifest variable, therefore, $\sum_{k=1}^{K_j} \pi_{jrk} = 1$.

Further denote as p_r the *R* mixing proportions that provide the weights in

the weighted sum of the component tables, with $\sum_{r=1}^{R} p_r = 1$. The probability that an individual *i* in class *r* produce a particular set of *J* outcomes on the manifest variables, assuming local independence, is the product

$$f(Y_i) = \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}}$$

The probability density function across all classes is the weighted sum

$$\Pr(Y_i \mid \pi, p) = \sum_{r=1}^{R} p_r \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}}$$

The parameters estimated by the latent class model are p_r and π_{jrk}. Given estimates $\hat{p}_r$ and $\hat{\pi}_{jrk}$ of p_r and π_{jrk}, respectively, the posterior probability that each individual belongs to each class, conditional on the observed values of the manifest variables, can be calculated using Bayes' formula:

$$\hat{\Pr}(r \mid Y_i) = \frac{\hat{p}_r f(Y_i; \hat{\pi}_r)}{\sum_{q=1}^{R} \hat{p}_q f(Y_i; \hat{\pi}_q)}$$

$\hat{\pi}_r$ are estimates of outcome probabilities *conditional on* class *r*.

The latent class model is estimated by maximizing the log-likelihood function

$$\ln L = \sum_{i=1}^{N} \ln \sum_{r=1}^{R} p_r \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}}$$

with respect to p_r and π_{jrk}, using the expectation-maximization (EM) algorithm

This log-likelihood function is identical with the standard finite mixture model log-likelihood. (ref:[12])

The EM algorithm proceeds iteratively:

1. Begin with arbitrary initial values of $\hat{p}_r$ and $\hat{\pi}_{jrk}$

2. In the E step, calculate the unknown class membership probabilities using

$$\hat{\Pr}(r \mid Y_i) = \frac{\hat{p}_r f(Y_i; \hat{\pi}_r)}{\sum_{q=1}^{R} \hat{p}_q f(Y_i; \hat{\pi}_q)}$$

And substituting in old $\hat{p}_r$ and old $\hat{\pi}_{jrk}$

3. In the M step, update $\hat{\Pr} = \frac{1}{N}\sum_{i=1}^{N} \hat{\Pr}(r \mid Y_i)$ as the new prior probabilities and

$\hat{\pi}_{jk} = \frac{\sum_{i=1}^{N} Y_{ij} \hat{\Pr}(r \mid Y_i)}{\sum_{i=1}^{N} \hat{\Pr}(r \mid Y_i)}$ as the new class-conditional outcome probabilities, $\hat{\pi}_{jk}$ is the vector of length K_j of class-*r* conditional outcome probabilities for the *j*th manifest variable.

4. Repeat step 2 and step 3, until the overall log-likelihood reaches a maximum and stop ‘ iterating when the overall likelihood or all parameter estimate changes are less than some arbitrarily small value. (In poCLA function, the default value is 1×10^{-10} when convergence has been reached.)

Once the latent class model is estimated, each case is assigned to the latent class for which it has the highest posterior probability of membership.

When we determine the number of the classes, The primary method is to iteratively test the goodness of fit of models with 2, 3, ..., up to the maximum plausible number of latent classes using the likelihood ratio chi-square test or Bayesian Information Criterion. Preferred models are those that maximum values of BIC or AIC.

*BIC=2*log L(m)-(npar)*log(n)*

$$AIC=2*\log L(m)-2*(npar)$$

We used BIC in this study.

3.4.2 Applied to SCS data.

We used the poLCA library in R. The response categories must be denoted by the integers 1, 2, ... since the algorithms do not allow 0 to denote a response level. This required recoding of the original CPILS categorical variables from a 0 to 2 scale to a 1 to 3 scale.

The following R commands perform latent class clustering analysis:

```
f  <- cbind(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p,
          q, r, s, t, u, v, w, x, y, z, aa, bb, cc, dd, ee) ~ 1
M2<poLCA(f,cpils2.df,nclass=3,na.rm=TRUE,maxiter=1000,graphs=TRUE)
```

Please refer to appendix IV for R program implementation of latent class clustering analysis.

(a) (b)

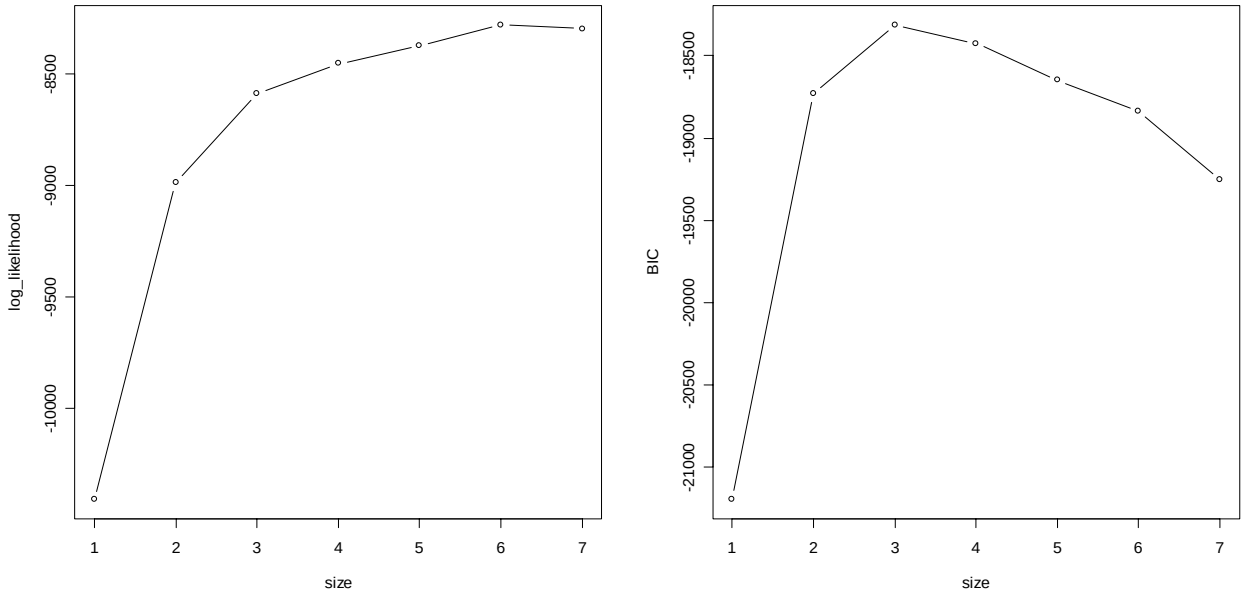


Figure 3.11 plots for assessing the best fitted model by log-likelihood (a) and BIC(b).

From figure 3.11, we determined that the model with 3 latent classes is the best fitted model since it has largest BIC value.

Table 3.5. : Classification of individuals based on the their most likely latent class membership

Latent Classes*	Class Counts	Proportion
1	181	0.42891
2	137	0.32464
3	104	0.24645

*the membership of class is unordered.

For each observation, the LCA model allows estimation of what class a person belongs to by referring to its posterior class membership probabilities. Table 3.5 shows how the original 422 observations in the CPILS dataset are allocated to the three latent classes with 181 (42.9%) categorized as Class 1, 137 (32.5%) as Class 2, and 104 (24.6%) as Class 3. Table 3.6 shows the average latent class probabilities for most likely latent class membership.

Table 3.6: Average Latent Class Probabilities for Most Likely Latent Class Membership (Row) by Latent Class (Column)

	1	2	3
1	0.971	0.014	0.015
2	0.035	0.965	0.000
3	0.020	0.000	0.980

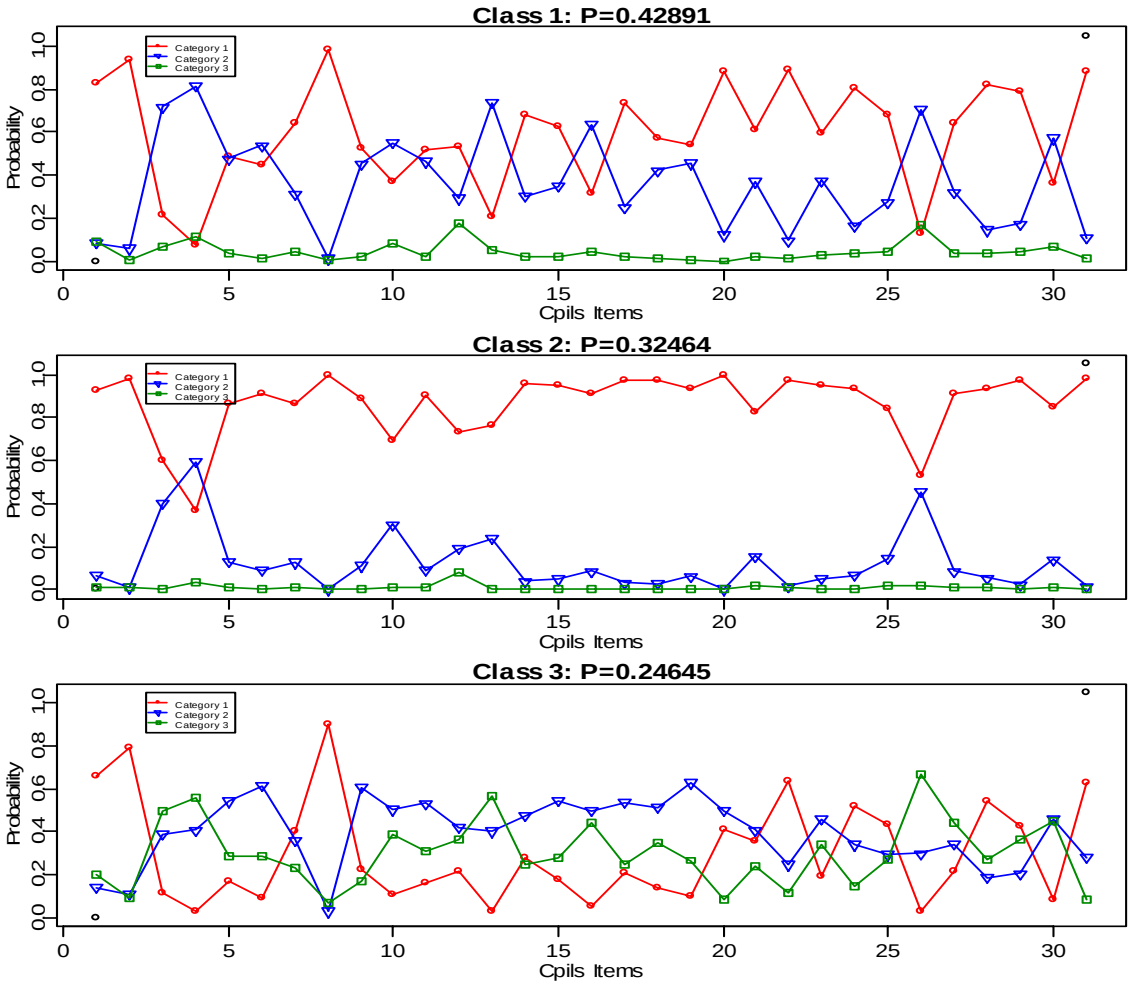


Figure 3.12 CPILS items conditional probabilities for 3 classes model

Figure 3.12 presents the estimated class-conditional response probabilities by original CPILS question. These are one plot for each latent class or type of individual. For each different CPILS item the expected probability of responding 1, 2 or 3 are plotted above the item number. Probabilities of responding 1 are linked via a red line across all CPILS items, responding 2 with a blue line and responding 3 with a green line. So, for example for CIPLS item one" Not being able to change jobs for fear of losing my health insurance coverage", if one belongs to Class 1, one has a 82.9% probability of saying "Not a problem for me" (response 1) , 8.2% chance to say "Somewhat a problem to me" (response 2) and 8.9% chance to say "A severe problem for me"

(response 3). If one belongs to Class 2, one has a 92.5% probability of saying "not a problem for me", 6.7% chance to say “Somewhat a problem to me” and 0.9% chance to say “A severe problem for me”. If one belongs to Class 3, one has a 65.9% chance of saying "not a problem for me", 14.3% chance to say “Somewhat a problem to me” and 19.8% chance to say “A severe problem for me”.

A slightly different view of these data is given in Figure 3.13. Since the sum of the probabilities of response to a question for any given latent class must sum to 1, these probabilities can be displayed on a barycentric graph. The three probabilities for any one question in a given class is plotted as one point on this graph. Here the points for all 31 questions and three classes are displayed. We see from this plot that individuals in latent class 3 typically have higher probabilities for response 3 and near equal probabilities for response 1 and 2. Class 2 individuals have lower probabilities for response 3 but a mix of questions that have high probabilities for response 2. Finally, latent class 1 individuals typically have high probability of response 1 for almost all questions.

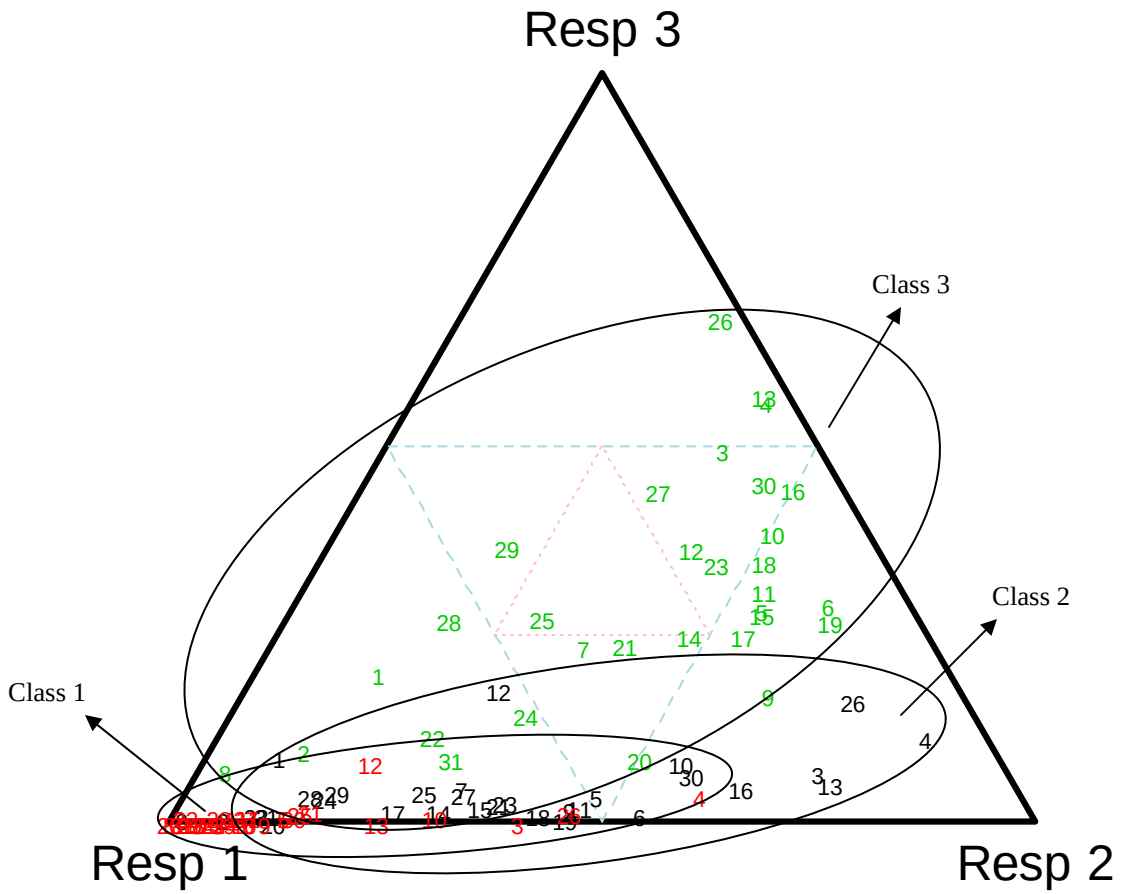


Figure 3.13 Barycentric coordinate display for 3 classes model

One other way of visualizing the results of the LCA results is to assign individuals to the latent class having the highest posterior probability and then coloring points to identify latent class membership in a bivariate scatter plot of the regular factor scores or the independent factor scores. This is given in figure 3.14 (a) and (b) respectively. This graph clearly demonstrates that the LCA does separate individuals into distinct clusters with little overlap.

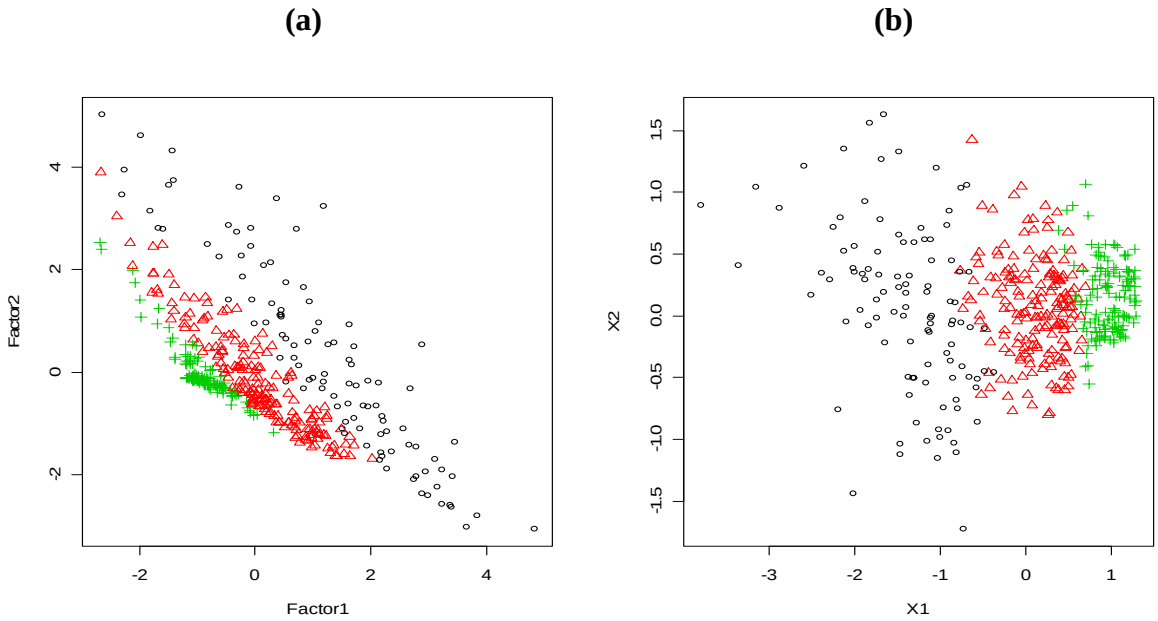


Figure 3.14 Plot for latent class classification on (a)fa score and (b) ifa score

Table 3.7: Comparison classifications for 5 methods

		LCA			raw			fa			ifa			ward		
	class	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
LCA	1	181	0	0	0	153	28	0	57	124	0	52	129	2	166	13
	2		137	0	88	48	1	82	6	49	99	6	32	106	31	0
	3			104	0	0	104	0	104	0	0	102	2	0	6	98
raw	1				88	0	0	71	0	17	83	0	5	86	2	0
	2					201	0	11	36	154	16	38	147	22	175	4
	3						133	0	131	2	0	122	11	0	26	107
fa	1							82	0	0	73	0	9	78	4	0
	2								167	0	0	146	21	0	57	110
	3									173	26	14	133	30	142	1
ifa	1										99	0	0	89	10	0
	2											160	0	0	54	106
	3												163	19	139	5
ward	1													108	0	0
	2														203	0
	3															111

3. 5. General application of these methods to categorical data

From figure 3.15, we can see that all five methods find clusters underlying the CPILS data but that the clusters found do not always have the same membership. This is quantified more extensively in table 3.7 where we look at how the clusters of each method overlap with those of the other methods. There does seem to exist at least 3 latent clusters among the lung cancer survivors. Latent class analysis (LCA) seems to have done a nice job on clustering these categorical data.

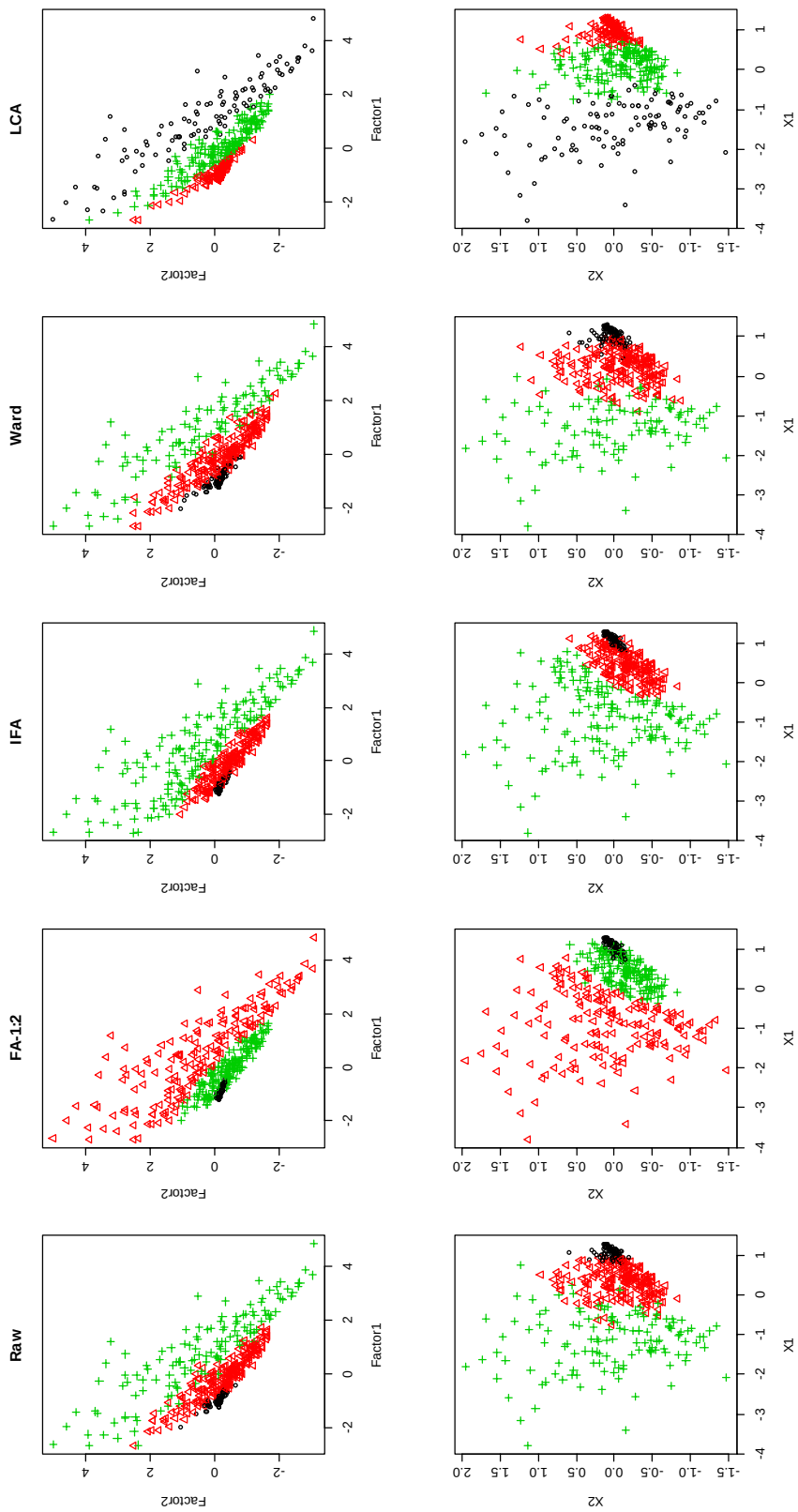


Figure 3.15 Display of the results of 5 methods classification on CPILS data

Chapter 4

Simulation Study

4.1 Introduction

In the previous chapter, we examined the ability of several techniques to cluster the SCS categorical data. In this chapter, we present some simulations in order to study the validity of the methods. As was seen in the previous chapter, the LCCA model did a good job in clustering observations into relevant groups. Since it can be used for identifying groups of observations in multivariate categorical data, estimating the characteristics of these groups, and predicting the probability that each observation belongs to each group, we use the latent class model's posterior probabilities to create a simulated dataset that can be used to test not only the properties of the latent class estimator, but also the properties of the other clustering methods which were used in the previous chapter, and then compare and contrast simulation results for each method to see what is actually happening.

4.2 Simulation Methods

4.2.1. Generating categorical responses – use of LCCA method

We use the poLCA package in R to first generate a random sample using appropriate normal probability distributions, and then create simulated categorical data by applying the class-conditional outcome probabilities (please refer to Figure 3.12) and class mixing proportions (class population shares, please refer to table 3.5) from the

output of the application of the three latent class model of the real CPILS dataset. The simulated data allows us to know the "true" class membership for each observation. So we use this known “true” classification to compare with the classification produced by the application of different methods to the simulated data, and as a result can evaluate their accuracy of clustering.

4.2.2 Clustering Methods applied to simulated data.

In this section, we applied the five clustering approaches which were used in the previous chapter to the simulation data. Methods examined include the distance clustering (Ward’s method), Model-based clustering on raw data, Model-based clustering on factor scores, Model-based clustering on independent factor scores and latent class (LCCA) clustering methods. (Please refer to appendix VI for program implementation in R)

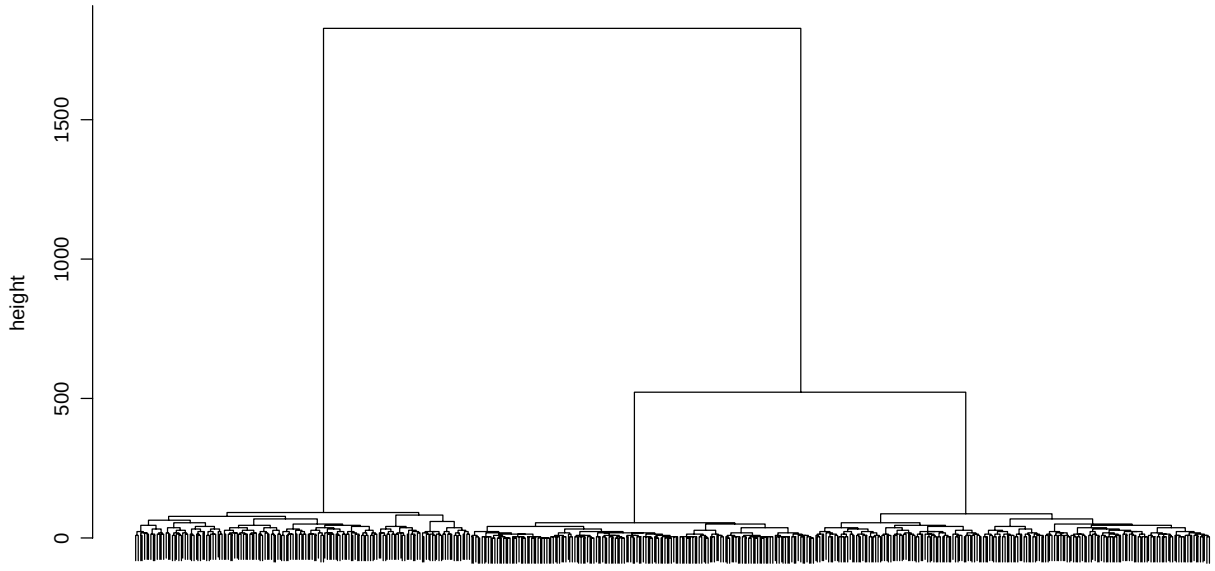
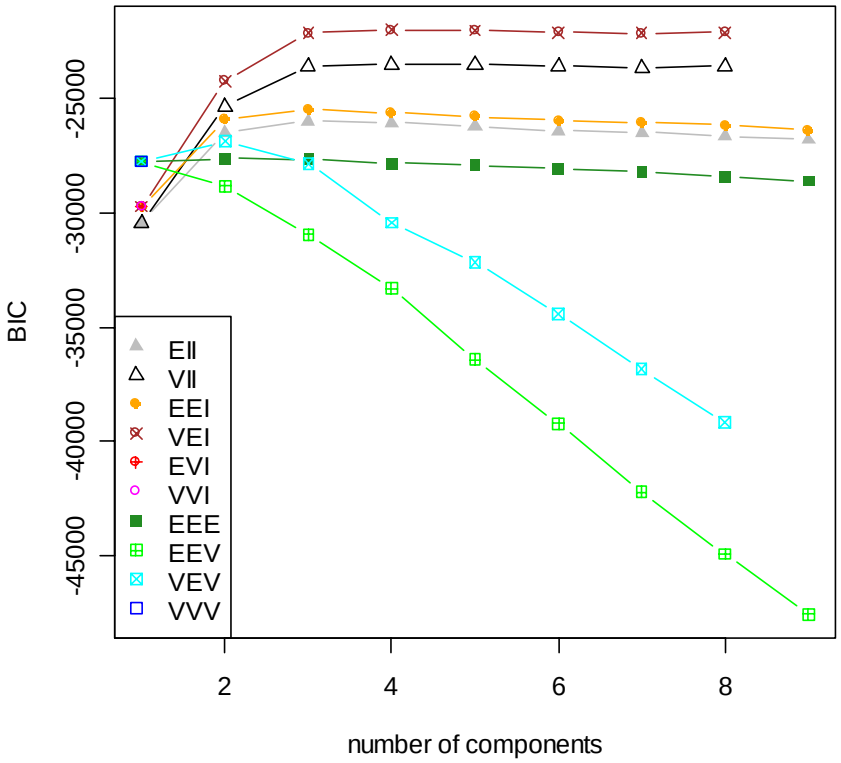


Figure 4.1 Hierarchical cluster dendrogrom applied to one simulated set of data

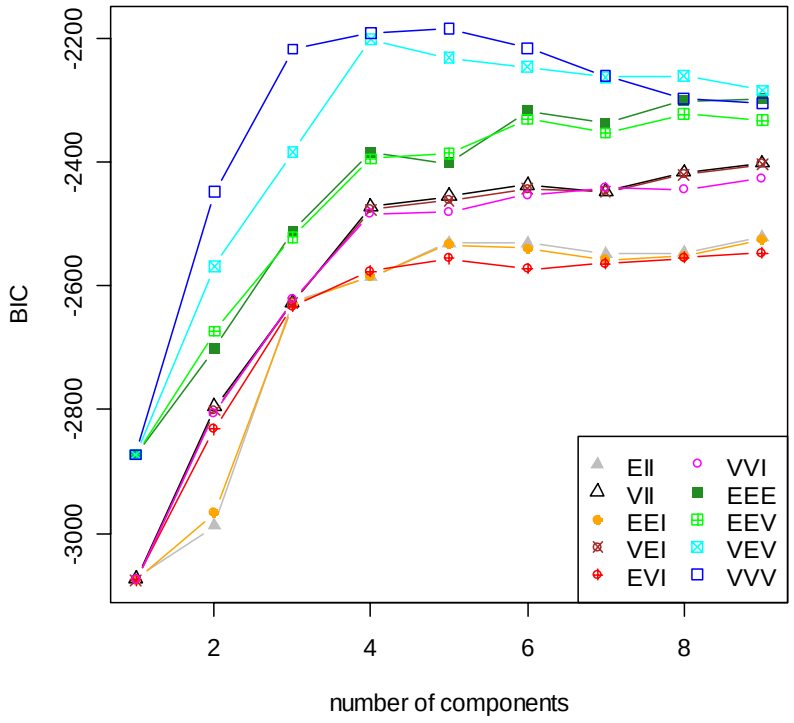
Figure 4.1 Applys hierarchical cluster analysis to the simulation data. It indicates three clusters clearly.

Figure 4.2 shows BIC values for model-based clustering on 3 sets of input data, (a) the best model is “VEI” on raw data. (b) The best model is “VVV” on factor score data. (c) the best model is “VEI” on independent factor score data.

(a)



(b)



(c)

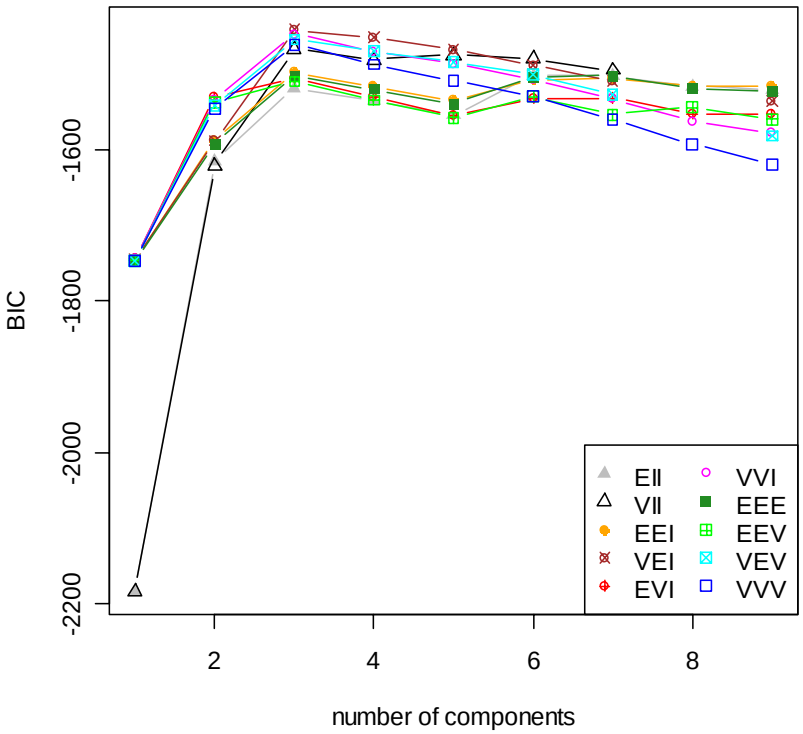


Figure 4.2 BIC for a simulated dataset, (a). Raw data. (b). Factor scores. (c). Independent factor scores.

Table 4.1 Classification compare with actual class

true class	LCA			raw			fa			ifa			ward			total
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
1	1	157	0	0	0	158	155	0	3	2	0	156	4	0	154	158
2	1	0	149	19	131	0	0	142	8	3	147	0	1	149	0	150
3	187	0	5	187	3	2	0	3	189	187	5	0	179	11	2	192
	189	157	154	206	134	160	155	145	200	192	152	156	184	160	156	500

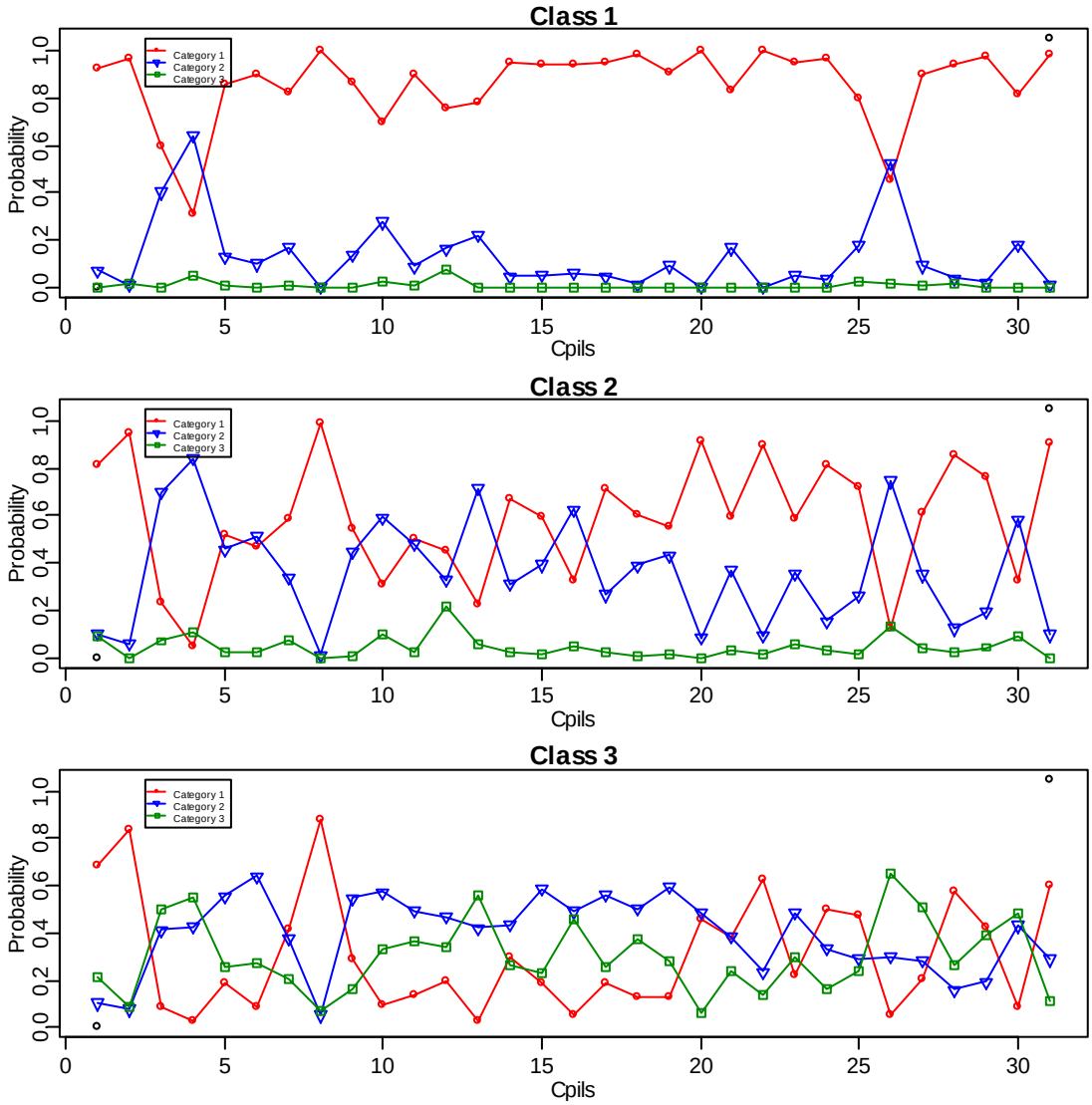


Figure 4.3 Simulated Data conditional Probabilities for 3 Classes Model

Table 4.1 shows the classification results of the 5 methods applying for one simulated dataset of 500 observations, and comparison with the true class.

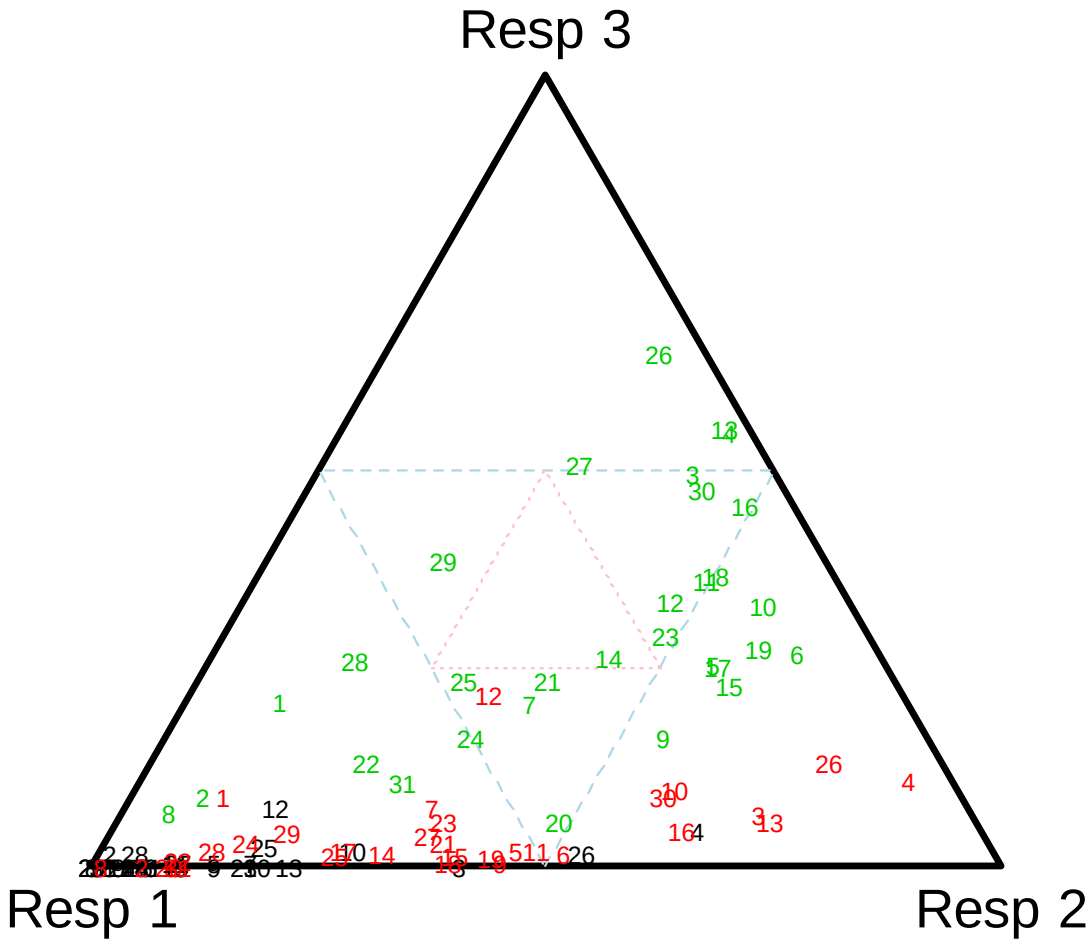


Figure 4.4 Barycentric coordinate display for 3 classed model

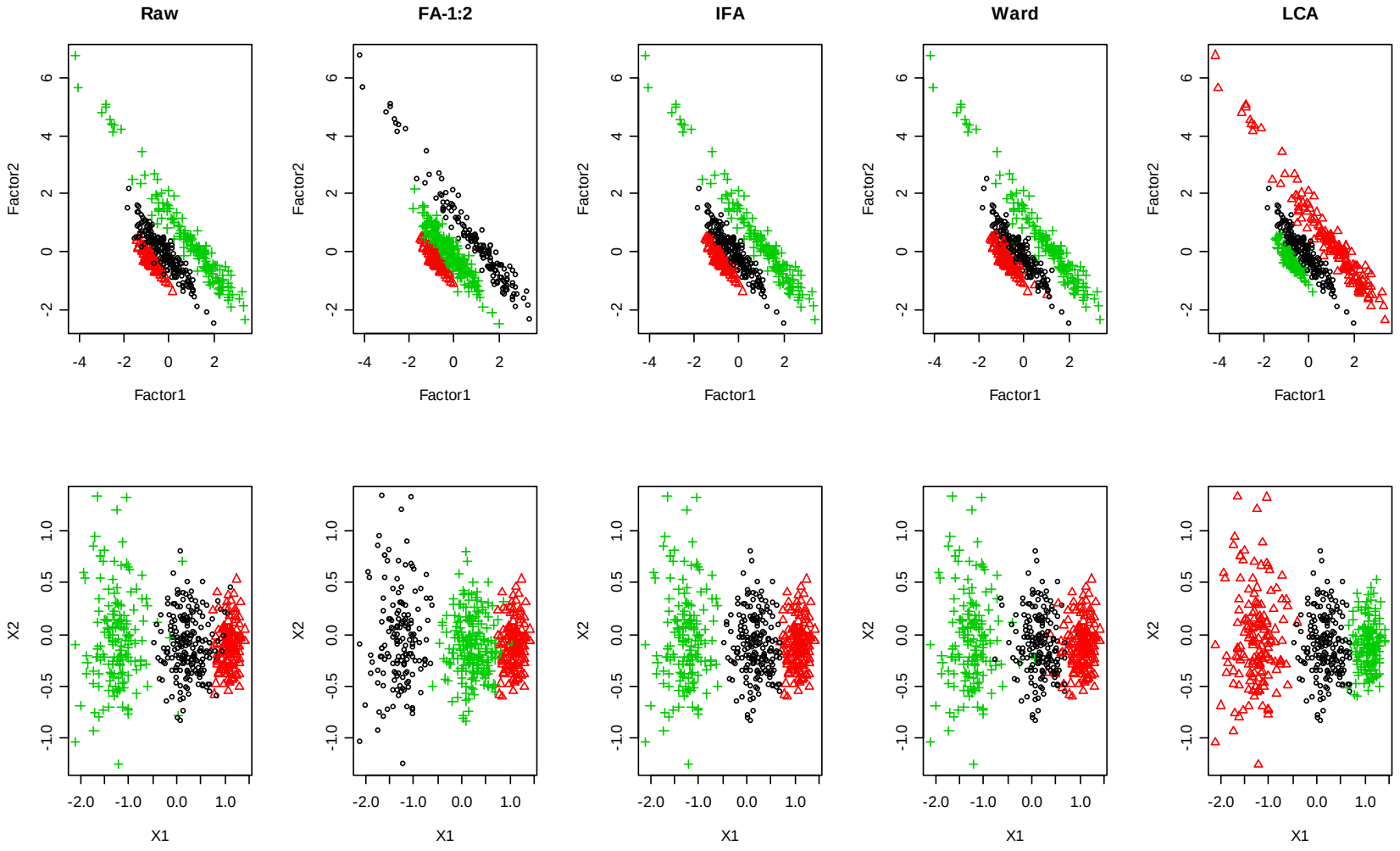


Figure 4.5 Comparison of 5 methods classification

4.3 Results

The different clustering models were applied to 20 simulated dataset and we recorded the proportions of the misclassification for each method. Table 4.3 summarizes the misclassification for each method and presents means, standard deviation, and 95% confidence interval for the whole experiment. Given that proportions are typically not normally distributed, the arcsine square root transformation is used to normalize the percentages for analysis purpose. In table 4.3 the arcsine of the square root of the raw data: $p_i' = \sin^{-1}(\sqrt{p_i})$ is first computed, then means and standard deviation are computed on the transformed data, and finally these means and 95% CI are converted it back into units of the raw data by taking the sine of the squared results (in this case, the 95% confident interval is estimated by $\sin(\overline{p}' \pm 1.96 * SD')^2$).

One-Way Analysis of Variance is one way to test the equality of three or more means at one time. We use the GLM procedure in SAS to test for mean equality on the transformed data (please refer to appendix VII for SAS program). From the SAS code output we compute a P-value that is <0.001 suggesting that at least one mean is significant different from other means. Using Tukey HSD multiple comparison procedures we analyze pairs of means to determine where the significant differences lie. The results are shown in Table 4.3.

Table 4.2: Summary of Misclassification

	Latent class analysis	Model-based cluster on factor scores	Model-based cluster on independent factor scores	Ward's method	Model-based cluster on raw data
1	1.00%	1.60%	2.00%	2.40%	5.60%
2	1.00%	1.40%	3.60%	3.40%	4.00%
3	2.20%	6.00%	9.00%	7.40%	12.20%
4	1.20%	3.20%	5.00%	3.00%	9.80%
5	1.40%	4.40%	5.80%	4.20%	9.40%
6	1.00%	2.00%	2.00%	4.40%	4.60%
7	0.80%	0.80%	1.20%	1.80%	1.80%
8	2.00%	1.80%	2.00%	3.60%	3.80%
9	1.00%	1.80%	3.00%	4.20%	3.00%
10	1.00%	1.40%	3.20%	3.40%	4.00%
11	2.20%	6.60%	8.40%	7.40%	12.20%
12	0.80%	1.80%	1.80%	3.40%	7.20%
13	2.00%	3.00%	2.60%	4.20%	11.80%
14	1.00%	1.00%	1.20%	5.00%	2.80%
15	2.00%	1.80%	2.20%	6.00%	3.40%
16	0.80%	1.60%	1.40%	3.00%	4.40%
17	1.80%	2.00%	1.80%	4.00%	2.40%
18	1.00%	1.40%	3.20%	3.40%	4.00%
19	2.20%	6.60%	9.00%	7.40%	12.20%
20	1.00%	1.80%	3.00%	4.20%	3.00%
Sample size=500					
Average	1.33%	2.38%	3.27%	4.22%	5.71%
Standard deviation	0.54%	1.83%	2.54%	1.61%	3.72%
95% confidence interval*	(0.50%,2.55%)	(0.29%,6.49%)	(0.34%,9.16%)	(1.69%,7.89%)	(0.83%,14.92%)

*Based on Arcsine square root Transformation predictors

4.4 Discussion

From Table 4.2, it is clear that the latent class (LCCA) model yields the smallest misclassification when clustering these simulated categorical data, followed by model-based clustering on factor scores data and then model-based clustering on independent factor score data. Although Ward's method has higher misclassification, it also has a smaller standard deviation and hence smaller confidence interval, so it's maybe more reliable. Model-based clustering on raw categorical data seems to have the highest misclassification rate, suggesting that it may not be advisable to use the raw data directly for clustering categorical data. From table 4.3, we can

see that the performance of model-based clustering on raw data is significantly poorer than that of the other three methods except for Ward's method. There are few differences between model-based clusters on factor scores and clusters based on independent factor scores. These findings also suggest that the latent class clustering method may be the most appropriate method categorical data. But, given that the simulation model data are originally generated according to an LCCA model structure it is not surprising that the LCCA clustering produced the best fit. What is surprising is that model-based clustering on factor analysis scores produced clusters that were very similar to that produced by the LCCA fit. Given that LCCA requires more complex fitting methodology and specified software whereas software for factor analysis and model-based clustering are much more readily available, it would seem useful to use this second method as a starting point for clustering categorical data.

Table 4.3 Result of Tukey's HSD comparisons

Means with the same letter are not significantly different.				
Tukey Grouping		**Mean***	**N**	**Method**
	A	0.23908	20	raw
	A			
B	A	0.20545	20	ward
B				
B	C	0.18075	20	ifa
	C			
D	C	0.15442	20	fa
D				
D		0.11529	20	lca

* Based on Arcsine square root Transformation data

4. 5 Conclusion and further consideration.

From the above analysis and discussion, we can conclude that all these five methods will cluster categorical data, but their accuracy is different.

Among them, the latent class clustering method seems the most appropriate method for clustering categorical data. Model-based clustering on raw categorical data with the assumption of multivariate normal clusters produces results that are clearly poorer than the other methods. The similarity of the model-based clustering of the common factor scores suggests a starting point for applications, especially with high dimensions.

The small sample size of this simulation study places limitations on the conclusions that can be made. In the future this study should be replicated with hundreds of simulated samples and from more than one underlying latent class structure. Other clustering technique, such as the use of Ward's clustering or a K-means clustering on the common factor scores should be examined as well.

References

[1]. American Cancer Society, Cancer Facts & Figures 2007, Atlanta: American Cancer Society; 2007.

[2]. American Cancer Society, National Quality of life Survey.

[3]. American Cancer Society, 2007. Quality of Life of Cancer Survivors, A Report From the American Cancer Society Study of Cancer Survivors, Atlanta: American Cancer Society; 2007, (unpublished manuscript).

[4]. McLachlan, G. and David Peel, 2000. Finite Mixture Models, John Wiley & Sons, New York .

[5]. Lattin, James, J. Douglas Carroll, Paul E. Green, 2002. Analyzing Multivariate Data. Brooks/Cole-Thompson Learning Inc., Pacific Grove, CA.

[6]. Portier, Kenneth M,. 2005. STA 4702/5701 Multivariate Statistical Methods class notes. Including ClusterSeminar.ppt, (unpublished notes)

[7]. SAS Institute Inc., SAS OnlineDoc 9.1.3 ®, SAS/STAT User's Guide, 2002-2007, SAS Institute Inc. Cary, NC, USA.

[8]. Fraley, C. and A. E. Raftery, 2002. Model-based clustering, discriminant analysis and density estimation. **Journal of the American Statistical Association**. Vol.97, No.458. pages 611-631

[9]. Fraley, C. and A. E. Raftery, 2006. MCLUST Version 3 for R: Normal Mixture Modeling and Model-Based Clustering, Technical Report no. 504. (Web document) http://www.stat.washington.edu/fraley/tr504.pdf

[10]. Garson, G. David, 2008. Factor Analysis: Statnotes from North Carolina State

University, Raleigh, NC. Public Administration Program. (Web document) http://www2.chass.ncsu.edu/garson/pa765/factor.htm,

[11]. Attias, H,1999. Independent Factor Analysis, **Neural Computation**, 11, 803–851.

[12]. Linzer, Drew A. and Jeffrey Lewis, 2007. poLCA: Polytomous Variable Latent Class Analysis, Version 1.1 (Web document) http://userwww.service.emory.edu/»dlinzer/poLCA

[13]. Magidson, Jay, and Jeroen K. Vermunt, 2001,Latent Class Factor and Cluster Models, Bi-Plots, and Related Graphical Displays, **Sociological Methodology**, Vol.31.(2001), pp.223-264.

[14]. Garson, G. David, 2008. Latent class Analysis: Statnotes from North Carolina State University, Public Administration Program. (Web document) http://www2.chass.ncsu.edu/garson/PA765/latclass.htm , North Carolina State University, Raleigh, NC

[15]. Muthén, L.K. and Muthén, B.O. 1998-2007. Mplus User's Guide. Fifth Edition. Los Angeles, CA: Muthén & Muthén.

[16]. R Documentation, Package *stats* version 2.5.1, (Web document) http://stat.ethz.ch/R-manual/R-patched/library/stats/html/00Index.html

[17]. R Documentation, Package *mclust* version 3.1-1, (Web document) http://cran.r-project.org/web/packages/mclust/mclust.pdf

[18]. R Documentation, Package *ifa* version 5.0, (Web document) http://cran.miscellaneousmirror.org/doc/packages/ifa.pdf

[19]. R Documentation, Package *poLCA* version 1.1, (Web document) http://cran.r-project.org/web/packages/poLCA/poLCA.pdf

APPENDICES

Appendix I: Implementation in R for Section 3.1

```
#load all the required packages for this thesis
library(ifa)
library(Rcmdr)
library(ade4)
library(mclust)
library(psy)
library(poLCA)
library(TeachingDemos)

#write CPILS dataset from my local drive
cpils<- read.table("C:/Documents and Settings/lguo/My
Documents/thesis/cpils.csv", header=TRUE, sep=",", na.strings="NA",
dec=".", strip.white=TRUE)
attach(cpils)

#delete the missing values
cpils<-as.matrix(cpils)
cpils.data <- na.omit(cpils)
cpils.df<-data.frame(cpils.data)
names(cpils.df)
dim(cpils.df)


#================================================================
#                         Hierarchical clustering
#================================================================

# Hierarchical clustering  using manhattan metric and Ward's
# method
D<-  dist(cpils.df,method="manhattan")
clust.ward<-hclust(D, method="ward")

# plot hierarchical Dendrogram
plot(clust.ward, labels=FALSE,main=NULL, hang=0.05,axes=TRUE,
frame.plot=FALSE, ann=TRUE, sub=NULL, ylab="height")

#cut the tree into three clusters
clust.ward3 <- cutree(clust.ward,k=3)

#show the size of each cluster
table(clust.ward3)
```

Appendix II: Implementation in R for Section 3.2

```
#================================================================
#                    Model-based clustering on raw data
#================================================================

# Cluster the raw data
clust.raw<-Mclust(cpils.df) #return a best model
names(clust.raw)
summary<-summary(mclustBIC(cpils.df), data=cpils.df)
#specify the range of models and numbers of clusters to be #considered
summary

#plot BIC
plot(clust.raw)

#show the size of classification
table(clust.raw$classification)
```

Appendix III: Implementation in R for Section 3.3

```
#================================================================
#                 Model-based clustering on factor scores
#================================================================

#****Factor analysis: this is the maximum likelihood method  ****

# First perform a principal components analysis and compute SCREE
#plot
cpils.pca <- princomp(~.,data=cpils.df,cor=TRUE)
cpils.pca

cpils.fa<-factanal(cpils.df, factors=2,rotation="promax",
scores="regression") # varimax is the default

names(cpils.fa)
cpils.fa$loadings
cpils.fa$uniquenesses
cpils.fa$scores

#produce the SCREE plot.
par(mfrow=c(1,2))
scree.plot(cpils.df,use="P")

# add p screeplots of simulated random normal data
scree.plot(cpils.df,use="P",simu=20)
#plot factor scores
plot(cpils.fa$scores, main="factor scores")
fa.scores <- data.frame(cpils.fa$scores)

#plot 3D plot if factors=3
with(fa.scores,scatter3d(Factor1,Factor2,Factor3,surface=FALSE,
point.col="red",sphere.size=1.5) )
x <- with(fa.scores,identify3d(Factor1,Factor2,Factor3,col="blue"))

# Cluster the 2 factor scores
clust.fa<-Mclust(cpils.fa$scores, G=3)
names(clust.fa)
summary<-summary(mclustBIC(cpils.fa$scores),G=3, data=cpils.fa$scores)
summary

#plot BIC
plot(clust.fa)

#display perspective plot of density estimate
apply(cpils.fa$scores,2,range)
x<-grid1(60,range=range(cpils.fa$scores[,1]))
y<-grid1(60,range=range(cpils.fa$scores[,2]))
xy<-grid2(x,y)
```

```
xyDens<-
dens(modelName=clust.fa$modelName,data=xy,parameters=clust.fa$paramete
rs)
xyDens<-matrix(xyDens,nrow=length(x), ncol=length(y))
par(pty="m")
Z<-xyDens
persp(x=x,y=y,z=Z,main="Perspective Plot", expand=0.5,
box=TRUE,xlab="factor1",ylab="factor2", zlab="Density")
par(mfrow=c(1,2))

#display the classification, uncertainty
par(mfrow=c(1,2))
mclust2Dplot(data=cpils.fa$scores,what="classification",identify=TRUE,
parameters=clust.fa$parameters,z=clust.fa$z)
mclust2Dplot(data=cpils.fa$scores,what="uncertainty",identify=TRUE,par
ameters=clust.fa$parameters,z=clust.fa$z)



#=================================================================
#          Model-based clustering on independent factor scores
#=================================================================

#************   independent factor analysis model  **************

# the function to choose the optimal model
MYFIT<-function(size)
{
ndf<-length(size)
#init.values<-ifa.init.random(cpils.df,ndf)
fit<-ifa.em(cpils.df,size,it=200,eps=0.0001,scaling=TRUE)
#list(H=fit$H,L=fit$L,NumVar=fit$numvar,NumObs=fit$numobs,W=fit$w,
#MU=fit$mu,VU=fit$vu,BIC=ifa.bic(fit),LIK=fit$lik,
#PSI=diag(fit$psi))
list(BIC=ifa.bic(fit),LIK=max(fit$lik))}

#select the best model by BIC
MYFIT(c(1,1))
MYFIT(c(2,1))
MYFIT(c(2,2))
MYFIT(c(3,1))
MYFIT(c(3,2))
MYFIT(c(3,3))
MYFIT(c(1,4))
MYFIT(c(2,4))
MYFIT(c(3,4))
MYFIT(c(4,4))
MYFIT(c(1,1,1))
MYFIT(c(2,1,1))
MYFIT(c(2,2,1))
MYFIT(c(2,2,2))
```

```
MYFIT(c(3,1,1))
MYFIT(c(3,2,1))
MYFIT(c(3,2,2))
MYFIT(c(3,3,1))
MYFIT(c(3,3,2))
MYFIT(c(3,3,3))
MYFIT(c(2,2,2,2))

# Cluster the ifa scores
size <- c(3,2)
ndf <- 2
fit2<-ifa.em(cpils.df,size,it=400,eps=0.0001,scaling=TRUE)
y_hat.df<-data.frame(ifa.predict(cpils.df, fit2))
clust.ifa<-Mclust(y_hat.df)


#display perspective plot of density estimate of CPILS data
apply(y_hat.df,2,range)
x<-grid1(60,range=range(y_hat.df[,1]))
y<-grid1(60,range=range(y_hat.df[,2]))
xy<-grid2(x,y)
xyDens<-
dens(modelName=clust.ifa$modelName,data=xy,parameters=clust.ifa$parame
ters)
xyDens<-matrix(xyDens,nrow=length(x), ncol=length(y))
par(pty="m")
Z<-xyDens
persp(x=x,y=y,z=Z,main="Perspective Plot",theta=30,phi=15,expand=0.5,
box=TRUE,xlab="ifa score1",ylab="ifa score2", zlab="Density")
par(mfrow=c(1,2))

#plot BIC
plot(clust.ifa)

#show the size of classification
table(clust.ifa$classification)

#display the classification, uncertainty
par(mfrow=c(1,2))
mclust2Dplot(data=y_hat.df,what="classification",identify=TRUE,paramet
ers=clust.ifa$parameters,z=clust.ifa$z)
mclust2Dplot(data=y_hat.df,what="uncertainty",identify=TRUE,parameters
=clust.ifa$parameters,z=clust.ifa$z)
```

Appendix IV: Implementation in R for Section 3.4

```
#=============================================================#
#                latent class clustering analysis             #
#=============================================================#
## models without covariates:
## M0: Loglinear independence model.
## M1: Two-class latent class model.
## M2: Three-class latent class model.

#recode cpils categorial variables 0=1, 1=2, 2=3
cpils2<- read.table("C:/Documents and Settings/lguo/My
Documents/thesis/cpils2.csv", header=TRUE, sep=",", na.strings="NA",
dec=".", strip.white=TRUE)
attach(cpils2)

#delete the missing values
cpils2<-as.matrix(cpils2)
cpils2.data <- na.omit(cpils2)
cpils2.df<-data.frame(cpils2.data)
names(cpils2.df)
dim(cpils2.df)

library(poLCA)
#define model formula f:
f  <- cbind(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,
u, v, w, x, y, z, aa, bb, cc, dd, ee) ~ 1

M0<-poLCA(f,cpils2.df,nclass=1,na.rm=TRUE,maxiter=8000)
M1<-poLCA(f,cpils2.df,nclass=2,na.rm=TRUE,maxiter=8000)
M2<poLCA(f,cpils2.df,nclass=3,na.rm=TRUE,maxiter=1000,graphs=TRUE)
M3<-poLCA(f,cpils2.df,nclass=4,na.rm=TRUE,maxiter=8000)
M4<-poLCA(f,cpils2.df,nclass=5,na.rm=TRUE,maxiter=8000)
M5<-poLCA(f,cpils2.df,nclass=6,na.rm=TRUE,maxiter=8000)
M6<-poLCA(f,cpils2.df,nclass=7,na.rm=TRUE,maxiter=8000)

# What is best fitting model
llik_L<-c(M0$llik,M1$llik,M2$llik,M3$llik, M4$llik,M5$llik, M6$llik)
bic_L<-c(-M0$bic,-M1$bic,-M2$bic,-M3$bic,-M4$bic,-M5$bic,- M6$bic)
par(mfrow=c(1,2))
plot(1:7, llik_L,type="b",xlab="size", ylab="log_likelihood")
plot(1:7, bic_L,type="b",xlab="size", ylab="BIC")

#estimated class-conditional response probabilities
M2$probs

# plot conditional probabilities for 3 classes model
cl<-data.frame(M2$probs)
dim(cl)

plotlca<-function(class, title)
```

```
{
plot(c(1,31), c(0,1.05), xlab="Cpils Items", ylab="Probability",
main=title)
lines(c(1:31),c(cl[class,seq(1,91,3)]), col="red",type="o",pch=1)
lines(c(1:31),c(cl[class,seq(2,92,3)]), col="blue",type="o",pch=6)
lines(c(1:31),c(cl[class,seq(3,93,3)]), col="green4",type="o", pch=22)
legend(2.5,1.05, c("Category 1", "Category 2", "Category 3"),cex=0.5,
lty=c(1,1,1),pch=c(1,6,22),col=c("red", "blue", "green4"),   merge=TRUE)
}

par(mfrow=c(3,1))
plotlca(1, "Class 1: P=0.42891")
plotlca(2, "Class 2: P=0.32464")
plotlca(3, "Class 3: P=0.24645")

#plot triplot
c1 <- matrix(1,nrow=31, ncol=4)  # class 1 probabilities
c2 <- matrix(2,nrow=31, ncol=4)  # class 2 probabilities
c3 <- matrix(3,nrow=31, ncol=4)  # class 3 probabilities

for (ques in (1:31))
{
   class <-M2$probs[[ques]]
   c1[ques,1:3] <- class[1,]
   c2[ques,1:3] <- class[2,]
   c3[ques,1:3] <- class[3,]
}
qs <- rep(rep(1:31,1),3)
clc <-rbind(c1, c2,c3)

triplot(clc[,1:3],txt=qs,col=clc[,4],pch=clc[,4],labels=c("Resp
1","Resp 2","Resp 3"),cex=.7)

#comparision classifications for consistency
table(M2$predclass,clust.raw$classification)
table(M2$predclass,clust.fa$classification)
table(M2$predclass,clust.ifa$classification)
table(clust.raw$classification,clust.fa$classification)
table(clust.raw$classification,clust.ifa$classification)
table(clust.fa$classification,clust.ifa$classification)
table(clust.ward3,clust.M2$predclass)
table(clust.ward3,clust.raw$classification)
table(clust.ward3,clust.fa$classification)
table(clust.ward3,clust.ifa$classification)
```

Appendix V: Implementation in R for Section 3.5

```
#*****  Comparison of 5 methods Applied on CPILS dataset *******

DoAn <- function(newc.df,pclass)
{
  # Cluster the raw data
  clust.raw<-Mclust(newc.df,G=3)

  # Cluster the 2 factor scores
  cpils.fa<-factanal(newc.df, factors=2,rotation="promax",
  scores="regression")
  clust.fa<-Mclust(cpils.fa$scores,G=3)

  # Cluster the ifa scores
  size <- c(3,2)
  ndf <- 2
  fit2<-ifa.em(newc.df,size,it=400,eps=0.0001,scaling=TRUE)
  y_hat.df<-data.frame(ifa.predict(newc.df, fit2))
  clust.ifa<-Mclust(y_hat.df,G=3)

  # Hierarchical clustering  using manhattan metric and Ward's method
  D<-dist(newc.df,method="manhattan")
  clust.ward<-hclust(D, method="ward")

  #cut the tree into three clusters and reconstruct the upper part of
  #the tree from the cluster centers.
  clust.ward3 <- cutree(clust.ward,k=3)

  # Plot cluster membership on different orientations
  attach(newc.df)
  f1 <- cbind(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s,
    t, u, v, w, x, y, z, aa, bb, cc, dd, ee) ~ 1
  clust.lca<-
  poLCA(f1,newc.df,nclass=3,na.rm=TRUE,maxiter=8000,graphs=FALSE)

  #  Plot using factor orientations.

  par(mfrow=c(2,5))
  plot(cpils.fa$scores[,1:2],pch=clust.raw$classification,
  col=clust.raw$classification,main="Raw")
  plot(cpils.fa$scores[,1:2],pch=clust.fa$classification,
  col=clust.fa$classification,main="FA-1:2")
  plot(cpils.fa$scores[,1:2],pch=clust.ifa$classification,
  col=clust.ifa$classification,main="IFA")
  plot(cpils.fa$scores[,1:2],
  pch=clust.ward3,col=clust.ward3,main="Ward")
  plot(cpils.fa$scores[,1:2],pch=clust.lca$predclass,
  col=clust.lca$predclass,main="LCA")

  # Plot using ifa factor orientations.
  plot(y_hat.df[,1:2],pch=clust.raw$classification,
  col=clust.raw$classification)
  plot(y_hat.df[,1:2],pch=clust.fa$classification,col=clust.fa$classification)
  plot(y_hat.df[,1:2],pch=clust.ifa$classification,
  col=clust.ifa$classification)
```

```
  plot(y_hat.df[,1:2],pch=clust.ward3,col=clust.ward3)
  plot(y_hat.df[,1:2],pch=clust.lca$predclass, col=clust.lca$predclass)
  # Compare classifications for consistency
  t.raw <-table(clust.raw$classification,pclass)
  t.ifa <-table(clust.ifa$classification,pclass)
  t.fa <-table(clust.fa$classification,pclass)
  t.ward<-table(clust.ward3,pclass)
  t.lca <- table(clust.lca$predclass,pclass)
  t.fa_ifa<-table(clust.fa$classification,clust.ifa$classification)

  list(t.raw=t.raw,t.ifa=t.ifa,t.fa=t.fa,t.ward=t.ward,t.lca=t.lca,
      t.fa_ifa=t.fa_ifa)
 }

DA <-DoAn(cpils2.df,pclass=M2$predclass)
DA
```

Appendix VI: Implementation in R for section 4.2

```
#*****************************************************************
#                          Simulation study
#*****************************************************************

#recode cpils categorial variables 0=1, 1=2, 2=3
cpils2<- read.table("C:/Documents and Settings/lguo/My
Documents/thesis/cpils2.csv", header=TRUE, sep=",", na.strings="NA",
dec=".", strip.white=TRUE)
attach(cpils2)

#delete the missing values
cpils2<-as.matrix(cpils2)
cpils2.data <- na.omit(cpils2)
cpils2.df<-data.frame(cpils2.data)
names(cpils2.df)
dim(cpils2.df)

# create simulated categorical data by applying the class-conditional
outcome #probabilities and class mixing proportions from the output of
the #application of the three latent class model to the real CPILS
dataset.
f  <- cbind(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s,
    t, u, v, w, x, y, z, aa, bb, cc, dd, ee) ~ 1
M2<-poLCA(f,cpils2.df,nclass=3,na.rm=TRUE,maxiter=1000,graphs=TRUE)

# The function check model-based clustering for three sets of data:
raw data, # factor scores and independent factor scores
CheckBest <- function(newc.df)
{
  # Cluster the raw data
  clust.raw<-Mclust(newc.df)

  # Cluster the 2 factor scores
  cpils.fa<-factanal(newc.df, factors=2,rotation="promax",
  scores="regression") # varimax is the default
  clust.fa<-Mclust(cpils.fa$scores)

  # Cluster the ifa scores
  size <- c(3,2)
  ndf <- 2
  fit2<-ifa.em(newc.df,size,it=400,eps=0.0001,scaling=TRUE)
  y_hat.df<-data.frame(ifa.predict(newc.df, fit2))
  clust.ifa<-Mclust(y_hat.df)

  #display BIC plots
  plot(clust.raw)
  dev.next(2)
  plot(clust.fa)
  dev.next(2)
```

```
  plot(clust.ifa)
  dev.next(2)
 }

# Fit best cluster model with three clusters to all three sets of
#data.
DoAn <- function(newc.df,pclass)
{
  # Cluster the raw data
  clust.raw<-Mclust(newc.df,G=3)

  # Cluster the 2 factor scores
  cpils.fa<-factanal(newc.df, factors=2,rotation="promax",
  scores="regression")
  clust.fa<-Mclust(cpils.fa$scores,G=3)

  # Cluster the ifa scores
  size <- c(3,2)
  ndf <- 2
  fit2<-ifa.em(newc.df,size,it=400,eps=0.0001,scaling=TRUE)
  y_hat.df<-data.frame(ifa.predict(newc.df, fit2))
  clust.ifa<-Mclust(y_hat.df,G=3)

  # Hierarchical clustering  using manhattan metric and Ward's
  # method
  D<-dist(newc.df,method="manhattan")
  clust.ward<-hclust(D, method="ward")

  #cut the tree into three clusters and reconstruct the upper
  # part of the
  #tree from the cluster centers.
  clust.ward3 <- cutree(clust.ward,k=3)

  # Plot cluster membership on different orientations
  attach(newc.df)
  f1 <- cbind(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q,
        r, s,
    t, u, v, w, x, y, z, aa, bb, cc, dd, ee) ~ 1
  clust.lca<-
  poLCA(f1,newc.df,nclass=3,na.rm=TRUE,maxiter=8000,graphs=FALSE)

  #  Plot using factor orientations.
  par(mfrow=c(2,5))
  plot(cpils.fa$scores[,1:2],pch=clust.raw$classification,
       col=clust.raw$classification,main="Raw")
  plot(cpils.fa$scores[,1:2],pch=clust.fa$classification,
       col=clust.fa$classification,main="FA-1:2")
  plot(cpils.fa$scores[,1:2],pch=clust.ifa$classification,
  col=clust.ifa$classification,main="IFA")
  plot(cpils.fa$scores[,1:2],
  pch=clust.ward3,col=clust.ward3,main="Ward")
  plot(cpils.fa$scores[,1:2],pch=clust.lca$predclass,
```

```
      col=clust.lca$predclass,main="LCA")

  # Plot using ifa factor orientations.
  plot(y_hat.df[,1:2],pch=clust.raw$classification,
       col=clust.raw$classification)

plot(y_hat.df[,1:2],pch=clust.fa$classification,col=clust.fa$classific
ation)
  plot(y_hat.df[,1:2],pch=clust.ifa$classification,
  col=clust.ifa$classification)
  plot(y_hat.df[,1:2],pch=clust.ward3,col=clust.ward3)
  plot(y_hat.df[,1:2],pch=clust.lca$predclass,  col=clust.lca$predclass)

  # Compare classifications for consistency
  t.raw <-table(pclass,clust.raw$classification)
  t.fa <-table(pclass,clust.fa$classification)
  t.ifa <-table(pclass,clust.ifa$classification)
  t.ward<-table(pclass,clust.ward3)
  t.lca <- table(pclass,clust.lca$predclass)
  list(t.raw=t.raw,t.fa=t.fa,t.ifa=t.ifa,t.ward=t.ward,t.lca=t.lca)
}

#generate simulated dataset
xk <- matrix(rnorm(500,mean=0,sd=2.0),nrow=500,ncol=1)
newcp <-
poLCA.simdata(N=500,probs=M2$probs,nclass=3,x=xk,classdist=M2$P,missva
l=FALSE)
newc.df <- newcp$dat[,1:31]
names(newc.df) <- names(cpils.df)

CheckBest(newc.df)
DA <-DoAn(newc.df,pclass=newcp$trueclass)
DA

# plot Hierarchical dendrogram
plot(clust.ward, labels=FALSE,main=NULL, hang=0.05,axes=TRUE,
frame.plot=FALSE, ann=TRUE, sub=NULL, ylab="height")

# plot conditional probabilities for 3 classes model
cl<-data.frame(clust.lca$probs)
cl

dim(cl)

plotlca<-function(class, title)
{
plot(c(1,31), c(0,1.05), xlab="Cpils", ylab="Probability", main=title)
lines(c(1:31),c(cl[class,seq(1,91,3)]), col="red",type="o",pch=1)
lines(c(1:31),c(cl[class,seq(2,92,3)]), col="blue",type="o",pch=6)
lines(c(1:31),c(cl[class,seq(3,93,3)]), col="green4",type="o", pch=22)
legend(2.5,1.05, c("Category 1", "Category 2", "Category 3"),cex=0.5,
```

```
       lty=c(1,1,1),pch=c(1,6,22),col=c("red", "blue", "green4"),
merge=TRUE)
}
par(mfrow=c(3,1))
plotlca(1, "Class 1")
plotlca(2, "Class 2")
plotlca(3, "Class 3")

c1 <- matrix(1,nrow=31, ncol=4)  # class 1 probabilities
c2 <- matrix(2,nrow=31, ncol=4)  # class 2 probabilities
c3 <- matrix(3,nrow=31, ncol=4)  # class 3 probabilities

#plot triplot
for (ques in (1:31))
{
   class <-lca$probs[[ques]]
   c1[ques,1:3] <- class[1,]
   c2[ques,1:3] <- class[2,]
   c3[ques,1:3] <- class[3,]
}
qs <- rep(rep(1:31,1),3)
clc <-rbind(c1, c2,c3)

triplot(clc[,1:3],txt=qs,col=clc[,4],pch=clc[,4],labels=c("Resp
1","Resp 2","Resp 3"),cex=.7)
```

Appendix VII: Implementation in SAS for section 4.3

```
*****************************************************************
*                SAS code for one way ANOVA test                *
*****************************************************************;

data test;
input grp $ ca @@;
cards;
lca   0.1002 lca  0.1002 lca  0.1489 lca  0.1098 lca  0.1186
lca   0.1002 lca  0.0896 lca  0.1419 lca  0.1002 lca  0.1002
lca   0.1489 lca  0.0896 lca  0.1419 lca  0.1002 lca  0.1419
lca   0.0896 lca  0.1346 lca  0.1002 lca  0.1489 lca  0.1002
fa    0.1268 fa   0.1186 fa   0.2475 fa   0.1799 fa   0.2113
fa    0.1419 fa   0.0896 fa   0.1346 fa   0.1346 fa   0.1186
fa    0.2598 fa   0.1346 fa   0.1741 fa   0.1002 fa   0.1346
fa    0.1268 fa   0.1419 fa   0.1186 fa   0.2598 fa   0.1346
ifa   0.1419 ifa  0.1909 ifa  0.3047 ifa  0.2255 ifa  0.2432
ifa   0.1419 ifa  0.1098 ifa  0.1419 ifa  0.1741 ifa  0.1799
ifa   0.2940 ifa  0.1346 ifa  0.1620 ifa  0.1098 ifa  0.1489
ifa   0.1186 ifa  0.1346 ifa  0.1799 ifa  0.3047 ifa  0.1741
ward 0.1555 ward 0.1855 ward 0.2755 ward 0.1741 ward 0.2064
ward 0.2113 ward 0.1346 ward 0.1909 ward 0.2064 ward 0.1855
ward 0.2755 ward 0.1855 ward 0.2064 ward 0.2255 ward 0.2475
ward 0.1741 ward 0.2014 ward 0.1855 ward 0.2755 ward 0.2064
raw   0.2389 raw  0.2014 raw  0.3568 raw  0.3184 raw  0.3116
raw   0.2162 raw  0.1346 raw  0.1962 raw  0.1741 raw  0.2014
raw   0.3568 raw  0.2717 raw  0.3507 raw  0.1681 raw  0.1855
raw   0.2113 raw  0.1555 raw  0.2014 raw  0.3568 raw  0.1741
;
run;

ods rtf;
proc glm data=test;
class grp;
model ca=grp;
means grp/ tukey ;
run;
ods rtf close;
```

```
************************************************************************
*                                SAS OUTPUT                            *
************************************************************************;
```

The GLM Procedure

Class Level Information

Class	Levels	Values
grp	5	fa ifa lca raw ward

Number of Observations Read	100
Number of Observations Used	100

Dependent Variable: ca

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	0.17949622	0.04487406	15.79	<.0001
Error	95	0.26992689	0.00284134		
Corrected Total	99	0.44942311			

R-Square	Coeff Var	Root MSE	ca Mean
0.399393	29.77937	0.053304	0.178997

Source	DF	Type I SS	Mean Square	F Value	Pr > F
grp	4	0.17949622	0.04487406	15.79	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
grp	4	0.17949622	0.04487406	15.79	<.0001

Tukey's Studentized Range (HSD) Test for ca

NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

Alpha	0.05
Error Degrees of Freedom	95
Error Mean Square	0.002841
Critical Value of Studentized Range	3.93274
Minimum Significant Difference	0.0469

Means with the same letter are not significantly different.

Tukey Grouping		Mean	N	grp
	A	0.23908	20	raw
	A			
B	A	0.20545	20	ward
B				
B	C	0.18075	20	ifa
	C			
D	C	0.15442	20	fa
D				
D		0.11529	20	lca