MSc Research Report

UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

Evaluation of Cluster Analysis and Latent Class Analysis in Clustering

By

Tatenda Murisa

Student Number: 879042

Supervisor

Dr Charles Chimedza

A research report submitted to the Faculty of Science, University of the Witwatersrand, in partial fulfilment of the requirements for the degree of Master of Science

School of Statistics and Actuarial Science

October 17, 2019

Abstract

The study compares the performance of latent class, K-means and hierarchical clustering on data with different degrees of cluster overlap. It also assesses how various standardisation methods affect the results of hierarchical and K-means clustering. Several distance and agglomeration methods are evaluated to observe how they perform depending on cluster overlap. Three artificial datasets were simulated whose clusters were poorly, moderately and well separated. These along with the seeds data were run through the three clustering methods. Several external validity indices were calculated for each cluster solution. The adjusted Rand index was used for comparison in the discussion because it is not affected by the number of clusters.

Results showed that Ward's method performed better compared to all other agglomeration methods and the Manhattan distance performed better across the different cluster types in hierarchical clustering. Latent class clustering performed better for poorly and well separated clusters. When the variance of the variables were comparable, K-means clustering with no standardisation performed well. Standardisation by the maximum value and z-score had the best cluster recovery when the variance of variables were large.

Definition of terms

1. ANN: Artificial neural network
2. ARI: Adjusted Rand Index
3. BIC: Bayesian information criterion
4. EDA: Exploratory data analysis
5. EM: Expectation maximisation
6. FM: Fowlkes-Mallows index
7. FMM: Finite mixture model
8. GMM: Gaussian mixture model
9. HC: Hierarchical clustering
10. ICL: Integrated complete-data likelihood
11. LC: Latent class
12. LCA: Latent class analysis
13. LCCA: Latent class cluster analysis
14. LPA: Latent profile analysis
15. LRTS: Likelihood ratio test statistic
16. MA: Morey and Agresti adjusted Rand index
17. ML: Maximum likelihood
18. MLE: Maximum likelihood estimator
19. MSE: Mean squared error
20. MVN: Multivariate Normal
21. PCA: Principal components analysis

22. RI: Rand index
23. SOM: Self organising map
24. VI: Variation of information

Contents

List of Figures

List of Tables

List of Appendices

Chapter 1

Introduction

1.1 Introduction

Clustering is the grouping of comparable objects into classes where the number and form of the classes are to be determined (Kaufman and Rousseeuw, 1990). It is one of the most important statistical techniques in multivariate analysis (Fraley and Raftery, 2007). Clustering is used in a variety of fields, such as psychology, biology, economics and engineering as a way of data reduction, generation of hypotheses and confirming groups that have a conceptual foundation (Hair et al., 2006). Several clustering methods which fall under broad categories such as partitioning methods, hierarchical, model based, density based and grid based clustering have been developed and studied (Rama et al., 2010).

Cluster analysis is usually associated with traditional techniques for instance the K-means (partitioning method) and hierarchical clustering. It can also be performed using latent class (LC) models which fall under the umbrella term finite mixture models (FMM). According to Masyn (2013) the use of FMMs dates back as far as 1894 when it was used by Karl Pearson. The use of the expectation maximisation (EM) algorithm in FMM by Dempster et al. (1977) simplified the process of estimating the model parameters

and hence increased its popularity. In addition, advances in computing power have led to increased use of latent class models which are viewed as state of the art techniques (Leisch, 2004; Fraley and Raftery, 2007). The EM algorithm for parameter estimation in LC models is highly computer intensive however, the astronomical increase in computer speed have made them practically applicable in recent times.

Traditionally, the latent class models in clustering are latent class analysis (LCA) and latent profile analysis (LPA). LCA models have both categorical latent and observed variables. LPA models have a categorical latent variable while the observed variables are continuous. Latent class clustering can also be performed on mixed-mode data (Everitt et al., 2011) where the data contains a combination of continuous and categorical observed variables. The use of latent class models in clustering has many alternative names in literature. These include model based clustering, Bayesian classification, latent class cluster analysis and the mixture likelihood approach (Magidson and Vermunt, 2002). These are collective names which include LCA, LPA and the mixed-mode data approach. From here on, the terms latent class (LC) cluster analysis or LC clustering will be used to refer to the use of FMM in clustering.

This research compares the performance of LC models, K-means and hierarchical clustering on data of varying degrees of cluster overlap and determining situations where it is preferable to use one method over the other. In LC models the sample data is assumed to come from a known statistical distribution or a mixture of probability distributions (Magidson and Vermunt, 2002). The K-means algorithm and hierarchical clustering do not make any distributional assumptions. However, all three techniques seek to minimise variation within a cluster and maximise between cluster variation.

A study by Steinley (2004) showed that cluster recovery is greatly improved by standardising the variables. This research will also compare the effects of different standardisation

methods on the K-means algorithm and hierarchical clustering. Clustering is an unsupervised technique where clusters are usually not known apriori. The response variable is unobserved thus, validation of the number of clusters is a challenge as there is no reference point to compare the performance of the model. Data with known cluster membership was used in this research in order to identify instances when LC, K-means and hierarchical clustering give the correct number of clusters.

1.2 Significance of the study

Nowadays data is being generated in large quantities at a fast rate and is often heterogeneous in nature. As a result, data with many observations (low-dimensional) or variables (high-dimensional) is very common. Analysis of each variable might not give a clear insight into the data generating process. Clustering can be used to group the variables or subjects into partitions before further analysis. This calls for techniques which accurately partition the data into clusters. It is therefore essential to identify the most effective way of clustering a particular dataset so that meaningful clusters are formed. This study compares the performance of latent class clustering, K-means and hierarchical clustering on data whose clusters have different degrees of overlap.

1.3 Aims and Objectives

1.3.1 Aims

The study compares K-means, hierarchical and LC clustering on clusters with different levels of overlap . It also seeks to compare the effectiveness of different standardising techniques on both K-means and hierarchical clustering. Lastly distance measures and agglomeration methods are compared in hierarchical clustering.

1.3.2 Objectives

The specific objectives of this study are to:

1. Compare the performance of hierarchical, K-means and LC clustering on data with varying degrees of cluster overlap.
2. Compare the results of different standardisation methods when using the K-means algorithm and hierarchical clustering.
3. Compare results of hierarchical clustering under different distance measures and agglomeration methods.

1.4 Limitations and Assumptions

In the application of clustering techniques the latent classes are often unknown. A real world dataset was used and data was also simulated such that the latent classes were known. This was done so that the results of different clustering solutions can be easily compared. In practice, the latent classes are usually unknown so it is more challenging to settle on the best clustering solution.

1.5 Layout of the report

The literature review in Chapter two reviews the theory behind hierarchical clustering, the K-means algorithm and LC models. Measures of assessment of model fit are also discussed in this chapter. A review of previous studies is done and lastly the R package MixSim which is used for data simulation is discussed. Chapter three describes how the datasets are simulated. A description of the Seeds data set is also given. The chapter ends by laying out what will be carried out during exploratory data analysis (EDA), hierarchical, K-means and LC clustering. Chapter four starts by discussing EDA results of the three simulated datasets and the Seeds data. Hierarchical clustering is then carried out to compare the distance measures, agglomeration and standardisation methods. The K-means algorithm and LC clustering are then carried out on all four datasets.

Chapter 2

Literature Review

2.1 Introduction

Hierarchical and K-means clustering are discussed in this section. This is followed by a review of latent class clustering. The chapter also focuses on measures of assessing model fit. A review of literature on latent class cluster analysis studies is done and the chapter concludes by looking at the MixSim package that is used for data simulation.

2.2 Hierarchical Clustering

Clustering techniques seek to partition objects into naturally occurring groups. Although these groups are not known beforehand, a researcher might have some background knowledge based on previous research. The difference between clustering and classification techniques is that in the latter the groups are known and the objective is to correctly classify an object based on the observed variables. The former seeks to partition objects into previously unknown categories based on some common properties.

Clustering techniques make use of measures of similarity or closeness which state how far apart two items are. When clustering items, distance reflects the similarity between them. For variables, the sample correlation is used as a measure of similarity, however,

clustering of variables is equivalent to non-parametric factor analysis (Rupp, 2013). The distance between the r-dimensional vectors $\mathbf{y}$ and $\mathbf{z}$ of observations can be measured in several ways which include:

$$\text{Euclidean distance: } d(\mathbf{y}, \mathbf{z}) = \sqrt{(\mathbf{y} - \mathbf{z})'(\mathbf{y} - \mathbf{z})} \ , \tag{2.1}$$

$$\text{Statistical distance: } d(\mathbf{y}, \mathbf{z}) = \sqrt{(\mathbf{y} - \mathbf{z})'\hat{\mathbf{\Sigma}}^{-\mathbf{1}}(\mathbf{y} - \mathbf{z})} \ , \tag{2.2}$$

where $\hat{\mathbf{\Sigma}}$ is the sample covariance matrix.

$$\text{Minkowski metric: } d(\mathbf{y}, \mathbf{z}) = \left[\sum_{i=1}^{r} |y_i - z_i|^m \right]^{1/m} , \tag{2.3}$$

where, m is an integer.

$$\text{Canberra metric: } d(\mathbf{y}, \mathbf{z}) = \sum_{i=1}^{r} \frac{|y_i - z_i|}{(y_i + z_i)} \ . \tag{2.4}$$

$$\text{Manhattan distance: } d(\mathbf{y}, \mathbf{z}) = \sum_{k=1}^{r} |y_{ik} - z_{ik}|. \tag{2.5}$$

Hierarchical clustering (HC) follows two approaches, either agglomerative or divisive clustering. Agglomerative HC begins with each point being a cluster and the points are merged to form clusters until a single cluster with all items is formed. The divisive HC approach does the opposite by beginning with a single cluster of all items and then separating them sequentially until each item is a cluster. This research will follow the agglomerative approach. The procedure is carried out as follows:

- If the number of items is N, the procedure starts with each item being a cluster.
- Two items which are most similar (shortest distance) are merged to form a cluster.
- Distances between clusters are calculated using a specified linkage and clusters with

the smallest distances are merged.

- The procedure is repeated $N-1$ times until a single cluster with all the N items remain.

A linkage is a method that is used to define the distance between two clusters. There are several available linkage methods that can be used; Johnson and Wichern (2007) discuss the single, complete and average linkage because they are not prone to inversion. Clusters are formed by grouping the closest pair of observations; so as more sub-clusters merge the distance between them increases. Inversion is when a cluster is formed at a smaller distance than that of previously merged clusters. This will result in a dendrogram whose scale is not monotonically increasing.

With the single linkage algorithm, the distance between two clusters is the minimum length between any pair of points, one coming from each cluster. In the complete linkage case the distance between two clusters is the maximum distance between pairs of points in the clusters. Finally, the average linkage uses the average distance between all point pairs to calculate the distance between two clusters. A disadvantage of the single linkage method is that a few points forming a bridge between clusters may cause the clusters to merge while that for the average linkage is that it does not perform well on elongated clusters (Rokach and Maimon, 2005). Clusters that are longer and have a short with are called elongated clusters. The average linkage is however, affected by a change in the measure of similarity (Johnson and Wichern, 2007).

Another variation of hierarchical clustering is Ward's method whose algorithm seeks to merge clusters in such a way that the increase in total within cluster variation is minimised. When clusters are merged the within cluster variation increase. The algorithm thus seeks to merge clusters which result in the least increase in total variance. Hierarchical clustering methods do not require the number of clusters to be specified at the beginning and their output which is a dendrogram (Figure 4.14 in Appendix C) provides a good

visualisation of the results.

One disadvantage of the hierarchical clustering technique is that once an item is allocated to an incorrect cluster it cannot be reallocated to another. The time required to carry out the algorithm increases exponentially as the sample size gets bigger (Rokach and Maimon, 2005). Also, when there are more than two pairs of clusters whose inter-cluster distances are equal, the methods can produce multiple solutions (Johnson and Wichern, 2007).

2.3 K-means clustering

The K-means clustering algorithm belongs to a broad category of techniques called non-hierarchical clustering algorithms. It is a partitioning method which allocates an object into exactly one cluster of the K mutually exclusive clusters. Johnson and Wichern (2007) state that the K-means algorithm is used to group objects rather than variables and it can be applied to large data sets. The K-means algorithm minimises the distance between points within a cluster in order to maximise similarity within a cluster and minimize similarity between clusters. Points in the same cluster must be more similar compared to points in different clusters.

In carrying out the K-means algorithm it is important to specify the number of clusters, K. The basic K-means algorithm works as follows:

1. Randomly select K points as the initial centroids for each cluster.
2. K clusters are formed by assigning points to a cluster whose centroid is closest.
3. Centroids for the new clusters are calculated and the points are reassigned to ensure that they are in a cluster whose centroid is closest.
4. The process is repeated until the cluster centroids remain constant and there is no more reassignment of items. At this stage the criterion is said to have converged.

The quality of a clustering solution depends on the choice of initial centroids. Some initial centroids will adjust themselves to the "true" centroids of the clusters while some may not. It is therefore, recommended to try out several random initial centroids to check for the consistency of the solutions.

In addition to the starting points, the effectiveness of a clustering algorithm also depends on the distance measure. There is no distance measure that works well with all types of data (Rama et al., 2010). Several distance measures can be tried out to see which one gives a better clustering solution.

The results of the K-means algorithm are affected by the variance of variables. Large variation gives a variable more weight thus it tends to be more influential in the analysis than those with less variation. In order to reduce the effect of unequal weighting, the variables need to be standardised. However, Milligan and Cooper (1988) mention that there are varying views towards equal weighting. Those against the idea argue that the different weights represent information that defines the clusters. There are various ways of standardising variables which result in different clustering solutions. This study will investigate the effects of various standardisation methods.

The following standardisation methods were compared by Milligan and Cooper (1988). In all the data sets the variables were uncorrelated. The first two methods involved division by the sample standard deviation (s). Z_1 is the standard normal z-score while Z_2 is similar to Z_1, the difference being that the scores have not been centred. The two are linear functions of each other, hence produce the same similarity matrix and clustering solution. They are given by:

$$\text{Z-score: } Z_1 = (Y - \bar{Y})/s, \text{ and} \tag{2.6}$$

$$Z_2 = Y/s, \tag{2.7}$$

where, Y is the original data value and $\bar{Y}$ the sample mean.

The transformation Z_3 which involves division by the maximum value of the variable, $\max(Y)$, results in proportions (for variables with positive values). According to Milligan and Cooper (1988) this measure does not equalize the variance of the variables and it is prone to outliers.

$$Z_3 = Y/\text{Max}(Y). \tag{2.8}$$

The standardisations Z_4 and Z_5 involve division by the range (maximum-minimum) and result in constant variance across the variables. They are however, both affected by outliers. These two standardisations are linear functions of each other therefore result in the same similarity matrix and clustering solution. The standardisations are defined as:

$$Z_4 = Y/(\text{Max}(Y) - \text{Min}(Y)), \text{and} \tag{2.9}$$

$$Z_5 = (Y - \text{Min}(Y))/(\text{Max}(Y) - \text{Min}(Y)), \tag{2.10}$$

where, $\min(Y)$ is the minimum value of the variable.

The transformation Z_6 is obtained by dividing by the sum of the values of the random variable, ΣY. It results in a constant mean but non-constant variance across the variables.

$$Z_6 = Y/\Sigma Y. \tag{2.11}$$

The rank transformation Z_7 where, rank(Y) is the rank of an observation results in a constant mean and variance across all variables. Although it reduces the impact of

outliers, it results in loss of information about the magnitude of differences (Milligan and Cooper, 1988).

$$Z_7 = \text{Rank}(Y). \tag{2.12}$$

In their comparison of the above standardisation methods, Milligan and Cooper (1988) used hierarchical clustering with the single, complete and average linkage methods as well as Ward's minimum variance method. Standardisation methods involving division by the range were more effective in all the different cluster structures and clustering methods. The popular z-score standardisation surprisingly was not the most effective method whilst the rank method was the least effective in all the cases. Results from hierarchical clustering are compared using the corrected Rand index, which is discussed in detail later under the assessing of model fit section.

After running the clustering algorithm, the solution must be validated to see if it fits the data well. A key challenge identified by George and Parthiban (2014) is that most partitions created are based on the assumptions of the algorithm therefore, may not be the best. The solution depends on criterion such as the value of K, initial partitions and similarity measures.

2.4 Latent Class Cluster Analysis

2.4.1 Finite Mixture Models

Finite mixture models (FMM) express the distribution of one or more variables as a mixture of a finite number of component distributions. FMM assume that the population is heterogeneous with respect to some observed variables. FMM is a broad term which includes models such as latent class analysis, latent profile analysis, growth mixture models, latent state models and regression mixture models (Masyn, 2013). A common property of these models is that the components of the mixture are not directly observed and the response variable is assumed to be influenced by the latent variable. This report

focuses on FMM with a single categorical latent variable and continuous observed or indicator variables. This will be referred to as latent class cluster analysis.

In finite mixture modelling data is assumed to come from a finite mixture of components. The probability density function of FMM with continuous indicator variables is given by

$$f(\mathbf{y}_i|\theta) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{y}_i|\theta_k) \tag{2.13}$$

where, $\mathbf{y}_i$ are the values of observed indicator variables for object i, K is the number of latent classes, θ_k's are the model parameters for latent class k and the mixing proportion π_k is the probability that an observation belongs to the k^{th} mixture.

It can be observed in Equation 2.13 that the distribution of $\mathbf{y}_i$ is a mixture of class specific densities $f_k(\mathbf{y}_i|\theta_k)$. The component densities may come from different parametric distributions although for most applications the same family of distributions is assumed (Scrucca et al., 2016). The normal or Gaussian distribution is commonly used as a distribution of the components. This research makes use of Gaussian mixture models (GMM) from the simulated datasets and it is also assumed for the seeds dataset. In this case, the probability density function of the k^{th} component is given by

$$f(\mathbf{y}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{\exp\{-\frac{1}{2}(\mathbf{y}_i - \boldsymbol{\mu}_k)^T {\boldsymbol{\Sigma}_k}^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_k)\}}{\sqrt{\det(2\pi\boldsymbol{\Sigma}_k)}} \tag{2.14}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean vector and covariance matrix.

Clusters in Gaussian mixture models are centred at the mean and are ellipsoidal (or spherical) in shape with a high density of points near the mean (Fraley and Raftery, 2007). In the case of a Normal distribution the variance-covariance matrix of the multivariate normal (MVN) distribution can take several forms. Vermunt and Magidson (2002) state that in addition to latent classes differing with respect to the location it allows them to

differ with respect to the homogeneity of the responses to the indicator variables. Banfield and Raftery (1993) proposed an eigen value decomposition of the covariance matrix in order to set constraints across the mixture components. The decomposition is given by

$$\boldsymbol{\Sigma}_k = \alpha_k \boldsymbol{B}_k \boldsymbol{S}_k \boldsymbol{B}_k^T \tag{2.15}$$

where, the scalar α_k controls the volume of the ellipsoid, the diagonal matrix $\boldsymbol{S}_k$ specifies the shape of the density contours and the orthogonal matrix of eigen vectors $\boldsymbol{B}_k$ determines the orientation of the ellipsoid. Therefore, the volume, shape and orientation defines the geometric properties of a cluster.

Three letters are used to specify the volume, shape and orientation of the model in that order, for example the model **EVV** has equal volume, variable shape and variable orientation. **E** stands for equal or same, while **V** is for varying or changing; in addition **I**, the identity matrix, is used with respect to the shape and orientation of the mixtures. In Table 2.1, **I** appears only twice in the shape column. In this instance, the data is one dimensional and it indicates equal shape of the spherical distributions. With respect to the orientation, **I** indicates that the clusters are parallel to the coordinate axes. Table 2.1 gives all the 14 different parameterisations that are available in R statistical software package Mclust version 5.

Latent class models are not only limited to continuous indicator variables but can also be specified for indicator variables on different measurement scales. A combination of continuous, nominal and ordinal indicator variables can also be included in the model. The model for mixed-mode data takes the form;

$$f(\mathbf{y}_i|\theta) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{J} f_k(\mathbf{y}_{ij}|\theta_{jk}), \tag{2.16}$$

where J is the number of indicator variables. The model assumes local independence.

Table 2.1: Covariance matrix decomposition

Model	$\boldsymbol{\Sigma}_k$	Distribution	Volume	Shape	Orientation
EII	$\alpha \boldsymbol{I}$	Spherical	Same	Same	-
VII	$\alpha_k \boldsymbol{I}$	Spherical	Changing	Same	-
EEI	$\alpha \boldsymbol{S}$	Diagonal	Same	Same	Parallel to axes
VEI	$\alpha_k \boldsymbol{S}$	Diagonal	Changing	Same	Parallel to axes
EVI	$\alpha \boldsymbol{S}_k$	Diagonal	Same	Changing	Parallel to axes
VVI	$\alpha_k \boldsymbol{S}_k$	Diagonal	Changing	Changing	Parallel to axes
EEE	$\alpha \boldsymbol{B}\boldsymbol{S}\boldsymbol{B}^T$	Ellipsoid	Same	Same	Same
EVE	$\alpha \boldsymbol{B}\boldsymbol{S}_k\boldsymbol{B}^T$	Ellipsoid	Same	Changing	Same
VEE	$\alpha_k \boldsymbol{B}\boldsymbol{S}\boldsymbol{B}^T$	Ellipsoid	Changing	Same	Same
VVE	$\alpha_k \boldsymbol{B}\boldsymbol{S}_k\boldsymbol{B}^T$	Ellipsoid	Changing	Changing	Same
EEV	$\alpha \boldsymbol{B}_k\boldsymbol{S}\boldsymbol{B}_k^T$	Ellipsoid	Same	Same	Changing
VEV	$\alpha_k \boldsymbol{B}_k\boldsymbol{S}\boldsymbol{B}_k^T$	Ellipsoid	Changing	Same	Changing
EVV	$\alpha \boldsymbol{B}_k\boldsymbol{S}_k\boldsymbol{B}_k^T$	Ellipsoid	Same	Changing	Changing
VVV	$\alpha_k \boldsymbol{B}_k\boldsymbol{S}_k\boldsymbol{B}_k^T$	Ellipsoid	Changing	Changing	Changing

The model specifies a univariate distribution for each indicator variable depending on the measurement scale. Poisson, binomial and negative binomial distributions can be used for count data; the multinomial distribution for nominal or ordinal variables and normal, student-t, gamma or log-normal distributions for continuous variables (Vermunt and Magidson, 2002).

In the case when there is local dependence within a class the model can also be adjusted so that the local independence assumption can be relaxed. In this case the multivariate normal, multinomial and multivariate poisson distributions can be used to model the correlated variables instead of the univariate distributions. Including local dependencies in latent class analysis helps to avoid too many clusters in a solution and objects are better classified into clusters (Vermunt and Magidson, 2002).

2.4.2 Parameter Estimation

The latent class model parameters to be estimated are the class specific means, variances, covariances and mixing proportions. The maximum-likelihood (ML) approach is used for parameter estimation. The maximum-likelihood estimators (MLE) for most mixture

models cannot be easily found by calculus methods, they are however solved using iterative methods such as the expectation maximisation (EM) where a solution of parameters is used to estimate an even better solution (Masyn, 2013). The expectation step involves calculating the expected values of missing observations using current parameter values. Once the full data set is obtained the parameters are then found using the MLE approach.

Finite mixture modelling was carried out by Karl Pearson in 1894 using the method of moments to estimate the model parameters (Masyn, 2013). There was limited application of mixture models due to the computationally intensive method of moments. The break through in mixture modelling came after the work of Dempster et al. (1977) who used the EM algorithm for estimation of incomplete data using ML estimation. Since latent class membership can be regarded as missing data, the EM algorithm can be used to find ML parameters of latent class models. The likelihood function is a function of the parameters of a statistical model given a set of observed data. The ML estimators are parameter values that maximises the likelihood of the observed data.

The likelihood function for data with n observations assuming a Gaussian mixture model is given by

$$\prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \phi(y_i; \mu_k, \Sigma_k), \tag{2.17}$$

where π_k is the mixing proportion and K is the number of components which is fixed for a particular model (Fraley and Raftery, 2007). The maximum likelihood estimates of the model which are the model parameters π_k, μ_k and Σ_k are estimated using the EM algorithm in finite mixture models. LC cluster analysis can be done in R using the Mclust package which carries out EM initialised by hierarchical model based clustering.

Masyn (2013) states that the log likelihood functions of most mixture models are complex having multiple local maxima, saddle points or unbounded solutions. For this reason the algorithm may fail to converge or may do so at a local maxima not a global maxima. The

iterations may continue for long because of the nature of the surface and therefore a rule for stopping is needed. The rules that are used are the maximum number of iterations (say 100) or a convergence criterion.

2.4.3 Expectation Maximisation

Dempster et al. (1977) showed in their paper that the EM algorithm had applications in a variety of areas such as censored data, FMMs, variance component estimation , hyper parameter estimation and factor analysis. The observations are treated as incomplete data and the latent variable is not directly observed but inferred through the observed variables. In latent class models the observed variables are a result of the latent variable not the other way around.

Let $\mathbb{X}$ and $\mathbb{Y}$ be the samples spaces of the latent and observed variables respectively; $\mathbf{x}$ (complete data) and $\mathbf{y}$ (incomplete data) are the respective realisations. It is assumed that there is a mapping from $\mathbb{X}$ to $\mathbb{Y}$ defined by $\mathbf{x} \rightarrow \mathbf{y}(\mathbf{x})$. A family of sampling densities for the complete and incomplete data with parameters $\boldsymbol{\theta}$ are given by $f(\mathbf{x}|\boldsymbol{\theta})$ and $g(\mathbf{y}|\boldsymbol{\theta})$ respectively, are connected by the relationship

$$g(\mathbf{y}|\boldsymbol{\theta}) = \int_{\mathbb{X}(\mathbf{y})} f(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x}. \tag{2.18}$$

EM seeks to find a value of $\boldsymbol{\theta}$ which maximises $g(\mathbf{y}|\boldsymbol{\theta})$ given an observed $\mathbb{Y}$ by making use of $f(\mathbf{x}|\boldsymbol{\theta})$ (Dempster et al., 1977). There are many forms of $f(\mathbf{x}|\boldsymbol{\theta})$ but for Gaussian finite mixtures it follows a multivariate normal distribution.

The EM algorithm operates in a cycle where, given a parameter estimate $\boldsymbol{\theta}_p$ after p iterations, it estimates $\boldsymbol{\theta}_{p+1}$ in the next iteration. The algorithm is described by Dempster et al. (1977) as follows: The expectation step involves estimating the complete data statistics $\mathbf{t}(\mathbf{x})$ by

$$\mathbf{t}^{(p)} = E[\mathbf{t}(\mathbf{x})|\mathbf{y}, \boldsymbol{\theta}_p]. \tag{2.19}$$

The maximisation step then finds $\boldsymbol{\theta}_{p+1}$ by solving the equations,

$$E[\mathbf{t}(\mathbf{x})|\boldsymbol{\theta}] = \mathbf{t}^{(p)}. \tag{2.20}$$

Equation 2.20 is the general form of the likelihood equations for maximum likelihood estimation for data which comes from a regular exponential distribution which includes the multivariate normal distribution. In addition Dempster et al. (1977) proved that the likelihood is increased by successive iterations and convergence of the solution shows that there is a stationary point of the likelihood function.

2.4.4 Allocation of latent classes

LC cluster analysis seeks to partition observations into distinct groups. An individual must belong to one and only one latent class. One method that is commonly used to classify subjects into latent classes is the use of classification or posterior probabilities. The posterior probability is the conditional probability of membership to a particular latent class given the observed response pattern (Collins and Lanza, 2010). There is uncertainty in the latent class membership.

An object is therefore assigned to a class whose posterior probability is the largest. The ideal situation is to have a high classification certainty where, there is a high probability in only one latent class and low probabilities in the other latent classes. A posterior probability nearer to 1 indicates that there is high certainty that an item belongs to a certain latent class while probabilities close to 0 also indicate with high certainty that an observation does not belong to a particular latent class. Posterior probabilities of around 0.5 indicate that latent class separation is poor.

2.5 Model Selection

This section discusses the criteria for model selection that is used for hierarchical, K-means and latent class clustering. Mclust uses the Bayesian information criterion (BIC), integrated complete-data likelihood (ICL) and the adjusted Rand index for model selection. Five indices based on counting pairs are calculated for each analysis. These are the Rand, Hubert and Arabie (HA) also referred to as the adjusted Rand index (ARI), Morey and Agresti (MA), Fowlkes-Mallows (FM) and the Jaccard index. These five indices compare the quality of the partition obtained by an algorithm to the original partitions of the data set. Their values all lie between 0 (indicating poor cluster recovery) and 1 (for perfect cluster recovery). Rodriguez et al. (2019) say that the number and size of clusters may introduce bias in some of the indices.

2.5.1 Bayesian Information Criterion

The BIC takes the log likelihood of the fitted model and subtracts a penalty based on the estimated number of parameters. As the number of estimated parameters increase, the penalty also increases due to over fitting of the model. In Mclust, the leading model has the largest BIC value. Scrucca et al. (2016) defines BIC as follows:

$$\text{BIC} = 2l_{M,K}(\boldsymbol{x}|\hat{\boldsymbol{\Theta}}) - v\log(n), \tag{2.21}$$

where $l_{M,K}(\boldsymbol{x}|\hat{\boldsymbol{\Theta}})$ is the maximum value of the likelihood function for model M, K the number of mixtures, n the sample size and v the number of parameters estimated. Mclust calculates BIC values for all models whose covariance matrices are non singular up to 9 clusters. In most literature the BIC formula is multiplied by -1 which makes a model with the smallest BIC the optimal. However, the BIC tends to select the number of mixtures which approximate the density well rather than the clusters (Scrucca et al., 2016).

2.5.2 Integrated Complete Data Likelihood

An alternative to BIC used in Mclust is the integrated complete-data likelihood criterion (ICL) which penalises the BIC through an entropy term which measures cluster overlap (Scrucca et al., 2016). The ICL is calculated as:

$$\text{ICL}_{M,K} = \text{BIC}_{M,K} + 2\sum_{i=1}^{n}\sum_{k=1}^{K} c_{ik}\log(z_{ik}), \tag{2.22}$$

where z_{ik} is the conditional probability that x_i arises from the k^{th} mixture component and c_{ik} take the value 1 if the i^{th} unit belongs to the k^{th} cluster and 0 otherwise. M is the name of the model and K the number of mixture components. Scrucca et al. (2016) say that ICL shows good results when cluster overlap is not too much. Just like BIC, Mclust also calculates by default ICL values for all models up to nine clusters.

2.5.3 Likelihood Ratio Testing

Mclust also makes use of the likelihood ratio test (LRT) for model selection. For each component covariance parameterisation it tests hypothesis $H_0 : K = K_0$ against $H_1 : K = K_{0+1}$ where, K_{0+1} is a model with one more component that model K_0. The LRT statistic (LRTS) is given by:

$$\text{LRTS} = -2\log\{L(\hat{\boldsymbol{\Theta}}_{K_0})/L(\hat{\boldsymbol{\Theta}}_{K_1})\} = 2\{l(\hat{\boldsymbol{\Theta}}_{K_1}) - l(\hat{\boldsymbol{\Theta}}_{K_0})\}, \tag{2.23}$$

where $\hat{\boldsymbol{\Theta}}_{K_i}$ is the MLE of parameter $\boldsymbol{\Theta}$ calculated under H_0 or H_1. Mclust makes use of bootstrap sampling to compute p-values for the test. The procedure outlined by Scrucca et al. (2016) is as follows: a bootstrap sample is generated by simulating data from the fitted model under the null hypothesis, LRTS is calculated for the sample after fitting models under the null and alternative hypothesis, these steps are repeated several times (G) to obtain the bootstrap null distribution of LRTS. The p-value is then estimated by

$$p - \text{value} \approx \frac{1 + \sum_{i=1}^{G} I(LRTS_b \geq LRTS_{obs})}{G+1}, \tag{2.24}$$

Table 2.2: Comparing Two Partitions

		Partition V				
	Class	v_1	v_2	$\cdots$	v_C	Total
Partition U	u_1	n_{11}	n_{12}	$\cdots$	n_{1C}	n_{1+}
	u_2	n_{21}	n_{22}	$\cdots$	n_{11}	n_{2+}
	$\vdots$	$\vdots$	$\cdots$		$\vdots$	$\vdots$
	u_R	n_{R1}	n_{R2}	$\cdots$	n_{RC}	n_{R+}
	Total	n_{+1}	n_{+2}	$\cdots$	n_{+C}	n_{++}

where $LRTS_b$ and $LRTS_{obs}$ are the statistics for the bootstrap and observed samples respectively and I(.) is the indicator function.

2.5.4 Rand Index (RI)

The Rand index (RI) is a commonly used measure in comparing clustering methods. It is the ratio of the number of element pairs which are assigned to the same cluster to the total number of element pairs. Given a set $S = \{O_1, O_2, ..., O_n\}$ with n distinct objects, suppose $U = \{u_1, u_2, ..., u_R\}$ and $V = \{v_1, v_2, ..., v_C\}$ are two different partitions of S with $\cup_{i=1}^{R} u_i = \cup_{j=1}^{C} v_j = S$ and $u_i \cap u_{i'} = v_j \cap v_{j'} = \emptyset$ for $1 \leq i \neq i' \leq R$ and $1 \leq j \neq j' \leq C$. If n_{ij} is the number of objects common to u_i and v_j, then Table 2.2 from Hubert and Arabie (1985) can be used to show the overlap of partitions U and V.

There are $\binom{n}{2}$ distinct object pairs in Table 2.2 which can be classified into four categories as:

1. w - objects in the pair are placed in the same cluster in U and in the same cluster in V,
2. x - objects in the pair are placed in the same cluster in U and in different clusters in V,
3. y - objects in a pair are placed in different cluster in U and in the same clusters in V,
4. z - objects in a pair are placed in different clusters in U and in different clusters in V.

The objects w and z show agreements while x and y represent disagreement from a pair of partitions. The Rand index is defined from these measures as

$$\text{Rand Index} = \frac{w+z}{\binom{n}{2}}. \tag{2.25}$$

The Rand index lies in the interval $[0, 1]$ where values close to 0 indicate low similarity whilst those close to 1 indicate similar clustering in the partitions. The Rand index is a probability of agreement between two partitions. However, the Rand index has some drawbacks. Santos and Embrechts (2009) say that the expected value of the Rand index is not constant, that is, given a null model it does not have a fixed value. In addition the Rand index approaches the upper bound as the number of clusters increase. Hubert and Arabie (1985) also say that the relative size of the index cannot be compared as it is not normalised to lie within fixed bounds. Gates and Ahn (2017) state that as sample size increases, the Rand index is influenced more by the number of pairs classified into different clusters which leads to less influence of pairs classified into the same clusters.

2.5.5 Adjusted or corrected Rand Index (ARI)

A better measure of cluster validity which is corrected for chance proposed by Hubert and Arabie (1985) is the Adjusted or corrected Rand index given by

$$\text{Adjusted Rand Index} = \frac{\binom{n}{2}(w+z) - [(w+x)(w+y)+(y+z)(x+z)]}{\binom{n}{2}^2 - [(w+x)(w+y)+(y+z)(x+z)]}, \tag{2.26}$$

or alternatively in summation notation as

$$\frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_{i}\binom{n_{i\cdot}}{2}\sum_{j}\binom{n_{\cdot j}}{2}/\binom{n}{2}}{\frac{1}{2}[\sum_{i}\binom{n_{i\cdot}}{2} + \sum_{j}\binom{n_{\cdot j}}{2}] - \sum_{i}\binom{n_{i\cdot}}{2}\sum_{j}\binom{n_{\cdot j}}{2}/\binom{n}{2}} \tag{2.27}$$

has an expectation of zero and a maximum value of 1. A larger value of the corrected Rand index implies higher agreement between two partitions therefore, good clustering results. Values closer to zero indicate less agreement between partitions hence poor

clustering results. An advantage of the ARI is that it is unaffected by the number of clusters (Rodriguez et al., 2019).

2.5.6 Morey and Agresti Adjusted Rand Index (MA)

The Morey and Agresti (1984) adjusted Rand index (MA) is an adjustment to the Rand index to correct for chance given by:

$$\text{MA} = \frac{2\Sigma\Sigma P_{ij}^2 - 2(\Sigma P_{i\cdot}^2)(\Sigma P_{\cdot j}^2)}{\Sigma P_{i\cdot}^2 + \Sigma P_{\cdot j}^2 - 2(\Sigma P_{i\cdot}^2)(\Sigma P_{\cdot j}^2)}, \tag{2.28}$$

where, P_{ij} is the probability that an item is allocated to cluster i in the first partition and cluster j in the second partition. Hubert and Arabie (1985) showed that the expected index in the MA was small there by creating positive bias. For this reason, when reference is made to the adjusted Rand index, it is the Hubert and Arabie adjusted Rand index (ARI).

2.5.7 Fowlkes-Mallows Index (FM)

Milligan and Cooper (1986) define the Fowlkes-Mallows (FM) index as:

$$\text{FM} = \frac{w}{\sqrt{(w+x)(w+y)}}. \tag{2.29}$$

The FM index lies between 0 (poor cluster recovery) and 1 (perfect cluster recovery). A draw back of the FM is that it tends to produce high values when there are a small number of clusters (Wagner and Wagner, 2007; Rodriguez et al., 2019).

2.5.8 Jaccard Index

The Jaccard index is defined by Rodriguez et al. (2019) as:

$$\text{Jaccard} = \frac{w}{w+x+y}. \tag{2.30}$$

It is similar to the Rand index with the only difference being that objects that are placed in different clusters in both partitions are not included. Its values have the same interpretation as the FM index.

2.5.9 Variation of Information (VI)

The VI is a method for comparing two partitions of the same data which makes use of information gain or loss. We begin by defining the terms that will be used in this section. Let $D = \{D_1, D_2, ..., D_K\}$ and $D' = \{D'_1, D'_2, ..., D'_{K'}\}$ be two different partitions of the same data set of size n. The sizes of clusters D_k and D'_k are n_k and $n_{k'}$ respectively, where $\sum_{k=1}^{K} n_k = \sum_{k'=1}^{K'} n_{k'} = n$. The probability of an outcome being in cluster D_k is a discrete random variable with K possible values. If each point has an equal chance of being selected, then the probability of an outcome being in cluster D_k is given by

$$P(k) = \frac{n_k}{n}. \tag{2.31}$$

Cover and Thomas (2012) defines entropy as a measure of uncertainty in a random variable or the amount of information needed to describe a random variable. If we are more certain about a random variable then less information is required to describe it. The more uncertain we are, the more information is needed to describe the random variable. The uncertainty of an outcome belonging to a particular cluster which is its entropy is defined by

$$H(D) = -\sum_{k=1}^{K} P(k)\log P(k). \tag{2.32}$$

The probability that a point belongs to cluster D_k in D and to $D'_{k'}$ in D' is

$$P(k, k') = \frac{|D_k \cap D'_{k'}|}{n}. \tag{2.33}$$

The mutual information between two clusterings is a measure of the amount of information

that one cluster has about another (Meilă, 2003). The mutual information is defined as

$$I(D, D') = \sum_{k=1}^{K} \sum_{k'=1}^{K'} P(k, k') \log \frac{P(k, k')}{P(k)P'(k')}. \tag{2.34}$$

Meilă (2003) proposed the variation of information criterion defined as

$$VI(D, D') = [H(D) - I(D, D')] + [H(D') - I(D, D')]. \tag{2.35}$$

Cover and Thomas (2012) state that entropy depends on probabilities of a random variable and not on its actual values. Therefore the VI does not depend directly on the sample size. Meilă (2003) argues that this makes the VI a good measure when clustering across data sets. The VI takes a value of 0 when $D = D'$ and is at most $2\log K^*$ where, K^* is the maximum number of clusters in D or D'.

2.6 Review of literature on latent class analysis studies

Latent class variable models such as factor analysis are used widely in social, behavioural and health sciences (Collins and Lanza, 2010). In the factor analysis model, the latent variable is continuous and normally distributed. Collins and Lanza (2010) state that extensive work has been done on continuous latent variable models but cite growing interest in the use of categorical latent variable models such as LCA and LPA.

Kent et al. (2014) made a comparison of two-step cluster analysis and LCA. The results showed that all the methods performed perfectly well in identifying subgroups on simulated data where the group membership of each observation was known. Results from real data sets were however different with varying numbers of subgroups formed. This highlights the challenges of clustering in identifying meaningful clusters when presented with real data sets.

Traditional clustering methods are K-means clustering and hierarchical cluster analysis. Although these methods produce good results, their performance is reduced when data is on different measurement scales. These clustering techniques use distance as a measure of similarity or dissimilarity, therefore the variables with high variation are more influential in the clustering algorithm. Such data may need to be standardised or scaled before analysis can begin. Different standardisation methods produce different results. On the other hand, LCA is a probability based clustering method which does not make use of distance in the clustering algorithms and as such does not require the data to be standardised. Thus, scaling is an issue in K-means and hierarchical clustering but LCA results are the same whether the variables are scaled or not.

DiStefano and Kamphaus (2006) compared LCA to K-means clustering to identify groups of child behavioural adjustment. LCA gave three clusters while K-means cluster analysis resulted in seven clusters. The results, though very different, provided unique insights into child behaviour. In a similar study, Eshghi et al. (2011) compared traditional clustering methods, Kohonen self organising maps (SOM) and latent class models when group membership was not known apriori. SOM is a type of artificial neural network (ANN) that is used to project a multidimensional data set into a few dimensions, usually two (Eshghi et al., 2011). In two dimensions, it is easy to visualise the data and to identify any clusters. The optimal number of clusters obtained from SOM and cluster analysis was eight while latent class models resulted in seven clusters. As a result of the different number of clusters obtained, group membership was also different which presented challenges in interpreting the solutions. Eshghi et al. (2011) therefore concluded that, in cases where the clusters are not known apriori, the interpretation of the results is largely dependent on the method used.

Magidson and Vermunt (2002) compared the results of K-means and latent class clustering in a population with two latent classes. The independent variables were normally distributed and the class membership of each observation was known. Results from latent class

clustering and K-means were compared to those of discriminant analysis to evaluate the efficiency of each model. Latent class clustering results were better than those of K-means. Standardisation of the variables improved the K-means results but not up to the level of latent class clustering. In their study they used only one data set and the degree of cluster overlap was not specified.

According to Magidson and Vermunt (2002), one of the advantages of latent class clustering is that rigorous statistical tests can be performed since it is model based. These tests may be used to compare models, determine the number of clusters and assess other model features. The K-means method makes use of mean squared error (MSE) metrics to determine the number of clusters and to choose one model over another. An additional advantage of a model based method such as LCCA is that the parameters can be restricted in an effort to get a more parsimonious solution (Magidson and Vermunt, 2002); for example parameter values such as the probability of a particular response in a latent class may need to be fixed. LCCA is able to accommodate simple and complex distributional forms of the observed variables, making it highly flexible.

In the above cases it is really difficult to say one method is superior to another. The study by Kent et al. (2014) showed that latent class analysis and K-means clustering gave comparable results. Although the study by DiStefano and Kamphaus (2006) produced different results, they both gave unique insights into the data, which was valuable. The fact that interpretation of results is dependent on the technique used further complicates the issues of choosing one method over the other.

2.7 Data Simulation

Synthetic data sets were simulated using the R package MixSim which was developed by Melnykov et al. (2012). The package can be used to simulate a Gaussian or non-Gaussian mixture with varying degrees of overlap with the option of adding outliers and noise. In

addition the clusters can be set to be spherical or non-spherical and homogeneous or non-homogeneous. Clusters differ with respect to features such as overlap, orientation, presence of noise, multi-modality and elongation; these features in turn affect the performance of clustering algorithms. Parallel distribution plots which show the degree of overlap of finite mixture models can also be plotted by the MixSim package.

The idea of using pairwise overlap to quantify the degree of cluster overlap was developed by Maitra and Melnykov (2010). They used average or maximum overlap to set the degree of complexity. In order to understand this concept better, consider a random variable $\boldsymbol{Y}$ whose distribution $f(\boldsymbol{y})$ is a finite mixture of Gaussian densities.

The distribution of $\boldsymbol{Y}$ is given by:

$$f(\boldsymbol{y}) = \sum_{k=1}^{K} \pi_k \phi(\boldsymbol{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.36}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean vector and covariance matrix of the k^{th} component respectively with a multivariate Gaussian density function $\phi(\boldsymbol{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. The mixing proportion for the k^{th} component is π_k which is the probability that $\boldsymbol{Y}_i$ is an element of the k^{th} component. $\phi()$ follows a multivariate normal distribution with probability density function:

$$\phi(\boldsymbol{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{p}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_k) \Big\} \tag{2.37}$$

Maitra and Melnykov (2010) defined the overlap between the i^{th} and j^{th} component as $\omega_{ij} = \omega_{i|j} + \omega_{j|i}$, where $\omega_{i|j}$ is the misclassification probability that $\boldsymbol{Y}$ was classified as coming from the i^{th} component when in fact it belongs to the j^{th} component. Similarly $\omega_{j|i}$ is the misclassification probability that $\boldsymbol{Y}$ was classified as belonging to the j^{th} component when the true origin is the i^{th} component. The misclassification probability is given by

$$\omega_{j|i} = \mathrm{P}[\pi_i \phi(\boldsymbol{y}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) < \pi_j \phi(\boldsymbol{y}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) | \boldsymbol{Y} \sim N_p(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)] \tag{2.38}$$

Maitra and Melnykov (2010) sought to estimate parameters $\{(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) : k = 1, 2, 3, ..., K\}$ which satisfied some degree of clustering complexity in terms of the overlap defined by maximum ($\breve{\omega}$) or average ($\bar{\omega}$) overlap. They further show that for any pair of mixing proportions π_i and π_j the value of ω_{ij} approaches 1 as the cluster overlap increases i.e as they become more homogeneous. Therefore a value of ω_{ij} closer to zero indicates that the two clusters are well separated while for poor separation the value is close to one.

MixSim simulates a data set by first simulating a mixture model. It allows the user to specify values for the average ($\bar{\omega}$) or maximum pairwise overlap ($\breve{\omega}$). Maitra and Melnykov (2010) recommend using both measures of cluster overlap when simulating Gaussian mixtures as maximum overlap is influenced by a single pair of clusters whilst average overlap may be influenced by a few clusters with a large degree of overlap. In addition bounds of the mean vector $\boldsymbol{\mu}_k$ and a lower limit for the mixing proportions π_k can be set by the user. Cluster sizes are then determined based on the mixing proportions followed by drawing random samples from each multivariate normal distribution in the mixture; at the same time allowing the user to have control on the outliers, noise variables and distribution of variables using the Box-Cox transformation.

MixSim can also be used to estimate the overlap of clusters when given the parameters of a GMM (Melnykov et al., 2012). This helps the researcher to understand the level of complexity of the clusters as well as having clues on potentially problematic clusters.

It is not easy to draw plots for multivariate data because of the high number of dimensions. Alternative methods that are used involve projecting the data into lower dimensions (one or two) and then drawing a plot. This however may result in loss of information. To better understand a mixture distribution, parallel distribution (PD) plots are used to

highlight the the similarities and difference in the components.

The interaction of the components of a mixture model can be visualised using a PD plot. The function pdplot() in MixSim allows the user to construct the plot after initially estimating the parameter of the mixture. The plot allows one to observe whether the mixtures are well separated or not.

The work by Maitra and Melnykov (2010) demonstrated that well separated clusters have pairwise overlap of less than 0.05 while moderate overlap was between 0.05 and 0.1 ; pairwise overlap of more than 0.1 indicate poorly separated clusters.

2.8 Conclusion

The next chapter describes the methodology that will be carried out in analysing the data. Techniques discussed in this chapter will be used .

Chapter 3

Methodology

3.1 Introduction

This Chapter details the procedures that are followed in the data analysis. The artificial datasets are simulated and then followed by exploratory data analysis (EDA) which is used to describe the data and checking for the possible presence of clusters. EDA helps the researcher to get a deeper insight into the data. Visual inspection of the data through bar charts, histograms and scatter plots is carried out to check for clustering tendencies in the data. Principal components analysis is performed on the data and the first 2 dimensions are plotted to get a better layout of the clusters. This is then be followed by hierarchical, K-means and LC clustering.

3.2 Data Simulation

Three data sets were simulated using the MixSim package in R statistical software. Table 3.1 summarises the properties of the simulated datasets. The 3 clusters structures were well separated (W), moderately separated (M) and poorly separated (P). The K-means algorithm works well for spherical clusters, therefore, all the simulated clusters were spherical so that no technique is disadvantaged by the type of data used. All the clusters were set to be non homogeneous in order to add complexity to the clusters by having

Table 3.1: Properties of simulated data sets

Property	Well Separated (W)	Moderate Separation (M)	Poorly Separated (P)
Average overlap	0.03	0.07	0.17
Maximum overlap	0.045	0.09	0.2
Clusters	3	4	3
Variables	4	4	5
Spherical	TRUE	TRUE	TRUE
Homogeneous	FALSE	FALSE	FALSE
Random seed	2013	2008	1995
Minimum π_k	0.2	0.1	0.2
Size	700	600	500

different orientation and volume.

The average ($\bar{\omega}$) and maximum ($\breve{\omega}$) overlap were set to 0.03 and 0.045 respectively for the well separated data. Dataset W has four variables and three clusters; the clusters were set to be spherical but non homogeneous. Data-set M had a average overlap of 0.07 and a maximum overlap of 0.09 with four variables and four clusters. The poorly separated data-set had an average and maximum overlap of 0.17 and 0.2 respectively. Three clusters and five variables were selected for this cluster.

Different seeds were used for the generation of each data set so that the results could be replicated. MixSim starts by creating a mixture with specified parameters followed by the generation of a dataset whose size is set by the user. Datasets W, M and P were of size 700, 600 and 500 respectively. In order to avoid clusters with very few observations, the minimum mixing proportions π_k were set to 0.1 for dataset M and 0.2 for the other two datasets.

3.3 Seeds data

The seeds data was obtained from the University of California, School of Information and Computer Science (UCI) machine learning repository (Dua and Graff, 2017). It contains seven geometric properties measured from wheat seeds. The wheat was collected from

experimental fields and the measurements were taken at the Institute of Agrophysics of the Polish Academy of Sciences (Charytanowicz et al., 2010). Seventy seeds were randomly selected from each of the three wheat types namely Kama, Rosa and Canadian.

X-ray images of the kernels of the sampled seeds were taken and printed. The kernel is the inner-most soft part of the seed. Measurements of the following geometric properties in mm were then taken from the images: area, perimeter, compactness (comp) $C = 4\pi A/P^2$, length of kernel (K length), width of kernel (K width), asymmetry coefficient (asy coeff) and the length of kernel groove (K groove) (Charytanowicz et al., 2010). Label 1 represents the Kama wheat variety, label 2 is Rosa variety and label 3 is for the Canadian wheat variety.

3.4 Exploratory data analysis (EDA)

A summary of the variables in the study was done in R. It provides the quartiles, mean, median, minimum and maximum values which are important in determining the distribution of the variables. When the variable is categorical the frequency of each response is provided. The variance of each variable is also calculated as this will be used to check the link between variation and standardisation techniques.

In addition to the summary, histograms and density will be plotted for each continuous variable. These may provide an early indication of clustering tendencies in the data. Multi-modality is the situation where a variable has more than one mode. The presence of multi-modality in the univariate plots indicates the possible presence of clusters. It is vital to note that a pattern in a single variable may not be a true reflection of the pattern in the complete set of variables (Everitt et al., 2011). The univariate plots may show no evidence of possible clusters but combine to form clusters in the multivariate distribution.

A scatter plot shows how two variables are related. Scatter plots will be used to visually

inspect for the presence of clusters. They will be presented in the form of a scatter plot matrix which makes it easier to compare one variable to the others. In addition to identifying clustering patterns the scatter plots will also be used to check for the possible presence of outliers.

Scatter plots are ideal when the data set has a few variables. When there are hundreds or thousands of variables, the number of scatter plots also increases making visual inspection difficult. In cases where there are many variables, principal components analysis (PCA) can used to condense the data into fewer dimensions. The first few principal components can then be used to produce a scatter plot matrix with lower dimensions. The plot gives a suggestion on the presence or absence of clusters in the data.

A distance matrix is used to visually assess the presence of clusters in data. It is found by calculating the distance between observation pairs. Commonly used distance measures are the Euclidean and Manhattan distances (Equation 2.1 and 2.5). Items which are more similar have a distance closer to 0 between them and are displayed in consecutive order. This is seen as squares of the same colour (orange in this report) along the diagonal of the distance matrix. For each dataset, a distance matrix is plotted. Box and whisker plots will also be used to check for outliers and the distribution of the variables.

The MixSim package was used to estimate the overlap parameters and construct a parallel density plot for the seeds data. The overlap parameters are used to give an extent of cluster overlap in the data whilst the parallel density plot shows how the clusters are separated in the dimensions.

3.5 Hierarchical Clustering

Hierarchical clustering is carried out in order to evaluate agglomeration methods, distance measures and standardisation techniques. The agglomeration methods that were used

are Ward, single, complete and average linkage. For each of these four methods, the Euclidean, Manhattan, Canberra and Minkowski distance measures were used.

In addition various standardisation methods were explored in combination with the distance measures and linkages. Hierarchical clustering was performed initially with no standardisation followed by standardisation by the z-score, range and maximum value. For each dataset a total of 64 analyses were performed and the cluster recovery indices RI, ARI, MA, FM, VI and the Jaccard were calculated. Models with different numbers of clusters were estimated and the best results collected.

3.6 K-means clustering

The R package NbClust (Charrad et al., 2014) was used to decide on the ideal number of clusters in the datasets. It calculates 30 different cluster validity indices that are used to estimate the value of K in the K-means algorithm. The output is a bar graph summary of the proposed number of clusters.

The K-means algorithm was applied on the artificial datasets and the seeds data. The algorithm was applied to the raw data and then to the data standardised by the z-score, range and maximum value. Again the 5 indices of cluster recovery were calculated for comparison purposes.

3.7 Latent Class Cluster Analysis

LC cluster analysis was performed on the simulated datasets and on the seeds data set. After analysis, the results were compared to those of hierarchical and K-means clustering by looking at the cluster recovery indices (Equations 2.25, 2.26, 2.28, 2.29, and 2.30) of the models. The objective will be to identify in what instance one algorithm outperforms the other.

LC cluster analysis was performed in R using the package Mclust. This package was designed to perform clustering, classification and density estimation of Gaussian mixture models. The program first calculates BIC values for all the covariance matrix parameterisations using the EM algorithm and then gives a summary of three best models according to BIC values. A model is then fitted for the best BIC value. Options exist of fitting specific models as required by the user.

When the model is fitted, Mclust has a powerful tool for displaying and visualising the fitted model where data is projected to lower dimensions. The plots assist in viewing the structure and characteristics of the clusters. When performing dimension reduction in Mclust the tuning parameter can be set to 1. This results in dimensions being estimated using information from the cluster means only and gives projected data maximal separation (Scrucca et al., 2016).

3.8 Conclusion

The results of the methods outlined in this chapter are presented in the next chapter. Various methods of presentation such as tables and graphs are used to illustrate the results.

Chapter 4

Analysis

4.1 Introduction

Procedures outlined in Chapter 3 are applied to the datasets in this section. The artificial datasets well separated (W), moderately separated (M) and poorly separated (P) clusters are generated using the MixSim package. These datasets along with the seeds data (S) were then run through hierarchical, K-means and latent class clustering algorithms.

4.2 Analysis results for data set P

4.2.1 Exploratory data analysis on P

Data set P had 500 observations belonging to 3 clusters with sizes 149, 113 and 238. The box and whisker plots for the variables in P are shown in Figure A.1 in Appendix A. The distributions of all the variables are symmetrical as the median line is at the centre of the box. There are possible outliers in the data which are indicated by the points marked beyond the whisker ends. All of the five variables from **a** to **e** have at least one possible outlier. From the summary of P in Appendix A.1, the variance of the variables range from 530.5 (**c**) to 773.8 (**d**) and therefore, do not have large differences.

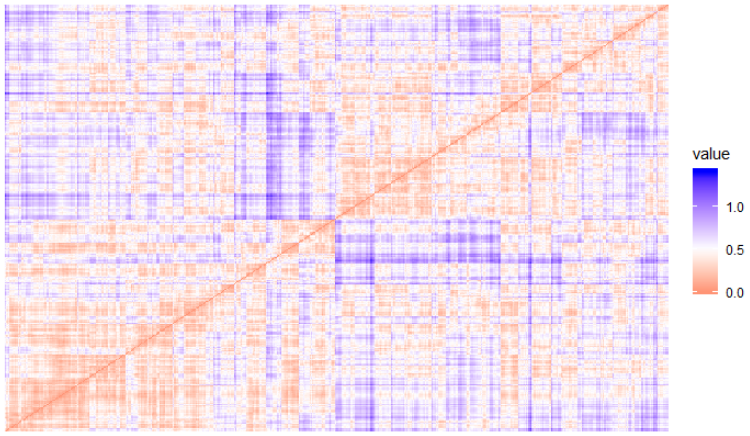

Figure 4.1: Distance matrix for P

The distance matrix in Figure 4.1 suggests the possible presence of clusters of similar observations indicated by the orange squares along the diagonal. A histogram gives an indication as to whether the data is clusterable. The presence of multiple peaks indicate possible clusters in the data. A density curve is super-imposed on the histogram to give a better visualisation of the distribution. For multivariate data, some patterns may not show in the univariate histograms but exist in the multivariate combination of the variables. Therefore, the patterns from the histograms may not be conclusive.

In Figure 4.2, the histograms for variables **b** and **d** are unimodal and their peaks are narrow which suggests that they have little influence in cluster formation. Variable **c** on the other hand shows two peaks which suggest the presence of two clusters. Variable **a** has one huge peak and a tiny one on the extreme right hand side. Variable **e** has a single peak, however, it is almost flat at the top which is the case when several high peaks combine. It therefore shows that there maybe clusters in the data but there is no indication of how many.

PCA was carried out to reduce the data dimensions so that a 2-dimensional plot of the data can be plotted. Table A.1 in Appendix A shows that the first 2 PCs account for only about 54% of the total variation in the data. This suggests that the 2-dimensional

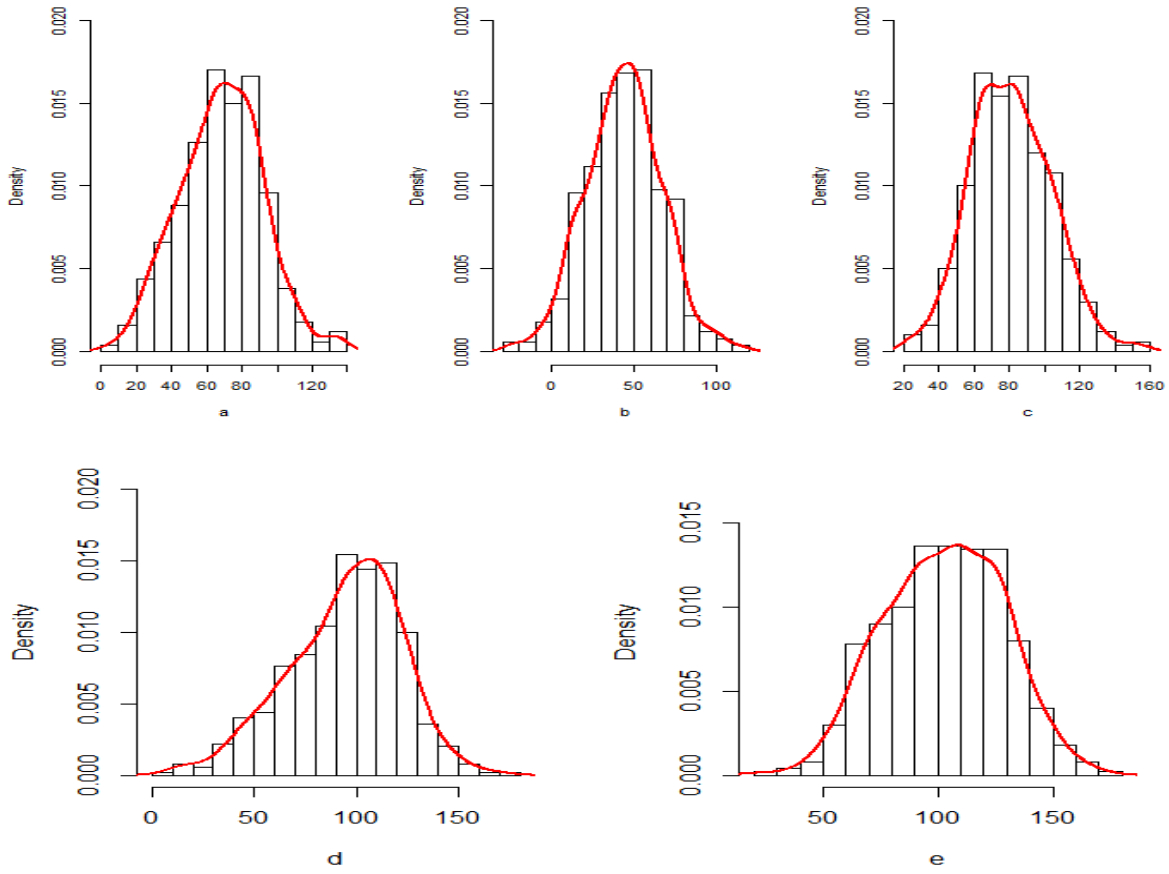

Figure 4.2: Histograms for variables in data set P

plot might not give a clear insight into the data.

Figure 4.3 is a 2-dimensional plot for dataset P. Clearly, the PCs do not differentiate the clusters well. The first PC separates cluster 3 from clusters 1 and 2 fairly. Separation in the second PC is poor. The pairs plot in Figure A.2 in Appendix A supports the poor separation in the PCA plot. For all the variable pairs there is no plot showing clear separation of clusters.

4.2.2 Hierachical Clustering on P

Table A.2 in Appendix A shows the cluster validity indices obtained by performing hierarchical clustering on dataset P. Generally the values of all the 5 indices are low (ARI less than 0.455) indicating poor cluster recovery. The best cluster recovery was

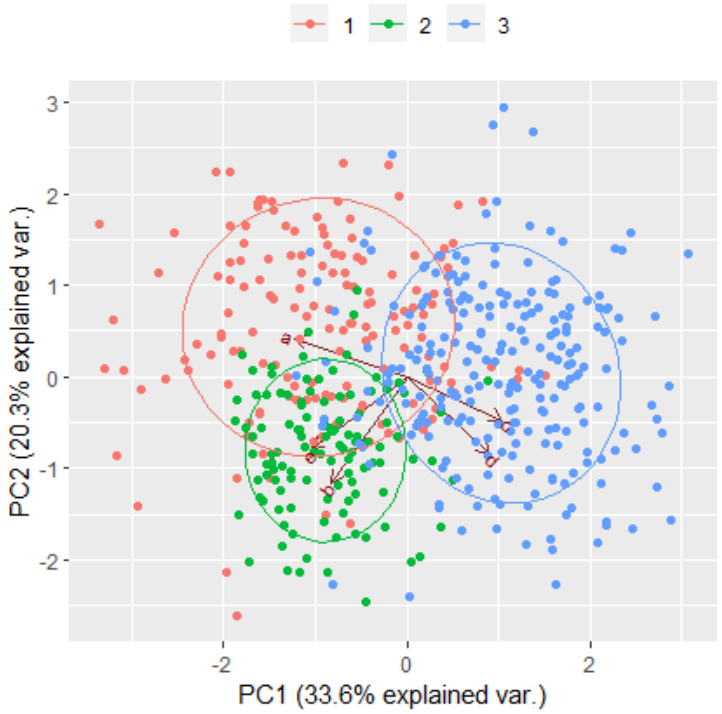

Figure 4.3: Principal components plot for P

achieved with standardisation by the range, using the Manhattan distance and with Ward's method. The dendrogram for the best recovery method is shown in Figure 4.4. The second best results were both achieved with standardisation by the range with Euclidean and Minkowski distance using Ward's method. In fact, all the results from Ward's method were superior to results from other agglomeration methods.

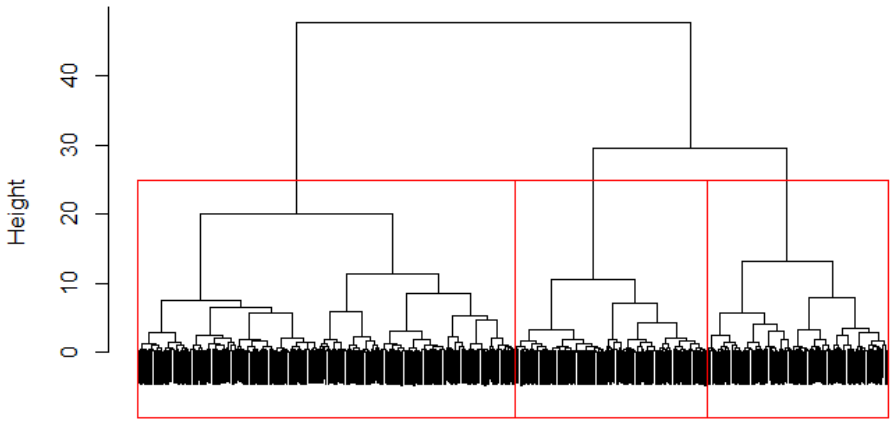

Figure 4.4: Dendrogram for best results of P

The ARI values for the average, complete and single linkages were very low (-0.015

Table 4.1: K-means and LC Results for data set P

			Rand	ARI	MA	FM	Jaccard	VI
K-means	2 cluster	No standardisation	0.710	0.420	0.422	0.674	0.499	1.065
		Z-score	0.709	0.417	0.418	0.671	0.496	1.081
		Range	0.708	0.417	0.418	0.671	0.496	1.092
		Maximum	0.663	0.327	0.328	0.618	0.440	1.233
	3 cluster	No standardisation	0.774	0.505	0.507	0.679	0.514	1.174
		Z-score	0.686	0.315	0.318	0.559	0.388	1.425
		Range	0.688	0.319	0.321	0.560	0.389	1.437
		Maximum	0.631	0.200	0.202	0.488	0.323	1.611
LC	3 cluster	No standardisation	0.824	0.617	0.619	0.755	0.606	
	4 cluster	No standardisation	0.810	0.579	0.581	0.723	0.564	

to 0.293)across all distance measures and standardisation methods. The single linkage method's extremely small (-0.004 to -0.001) negative values indicate its inability to recover the clusters well.

4.2.3 K-means clustering on P

Figure A.3 in Appendix A shows results of the NbClust indices used to decide on the best number of clusters. Eight indices suggested 2 clusters while seven suggested 3 clusters and therefore these were used as the values of K. Models with $K = 2$ and $K = 3$ were fitted using the four standardisation methods. The K-means algorithm was run several times until the highest ARI value appeared consistently for each combination of K and standardisation method. The K-means solution depends on the starting points hence, several runs helps in identifying a consistently good solution. Results with the best cluster recovery indices were taken for each value of K and standardisation method.

The best cluster recovery was achieved by the 3 cluster solution with no standardisation with an ARI value of 0.505 (see Table 4.1). This model also had the smallest variation of information (VI) index of all the 3 cluster solutions. The cluster plot for this solution which is shown in, Figure 4.5 shows the great extent of cluster overlap of this dataset as expected. The cluster centres, indicated by the large symbols actually lie in the concentration ellipse of another cluster. Surprisingly, the remainder of the 3 cluster

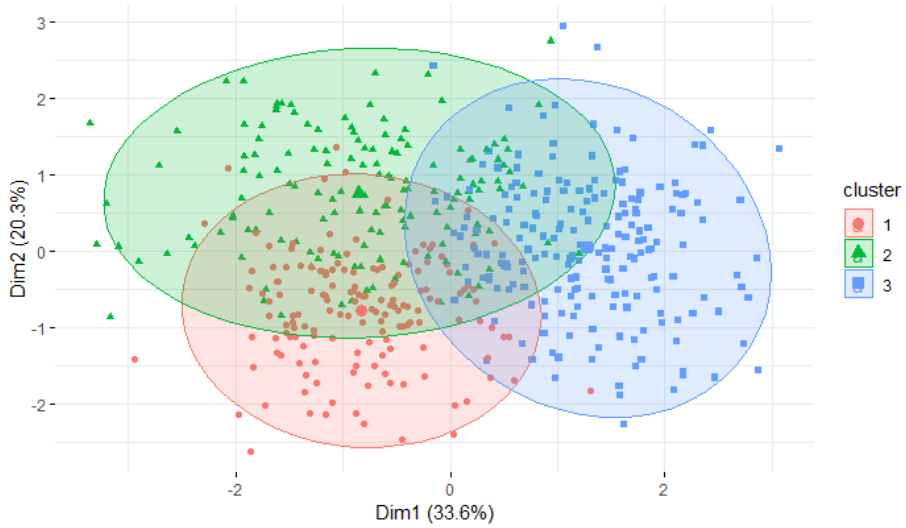

Figure 4.5: Cluster plot for K-means best result for P

solution results with standardisation have lower cluster recovery indices compared to all the 2 cluster solutions.

4.2.4 Latent class cluster analysis on P

LCCA was performed on dataset P and the results are displayed in Appendix A.3. The results showed that model VII had the best BIC value with 3 clusters. This was followed by VEI with 3 clusters and VII with 4 clusters. The VII model with 3 clusters also had the best ARI. The LRTS in Table A.3, Appendix A also showed the presence of 3 clusters while the ICL in Appendix A.3 was showing a single cluster. The BIC graph has a maximum value at 3 clusters while the one for ICL has a maximum value at 1 cluster (see Figures A.4 and A.5 in Appendix A). A summary of the best model is given in Appendix A.2 shows that that the size of the estimated clusters are 162, 222 and 116.

The 2-dimensional plot of the 3 cluster model shown in Figure A.6 in Appendix A (where Dir1 and Dir2 are the first and second principal components respectively) is better than the one for the K-means result because some cluster centres are not in the concentration ellipses of others except for 1 cluster. Uncertainty plots displays the uncertainty in

converting a conditional probability from EM to a classification in model-based clustering. On the plot, a darker colour indicates a region of high uncertainty. Therefore, there is a high degree in classification uncertainty for points lying in regions with dark colours. The uncertainty plot in Figure A.7 in Appendix A shows that there are a lot of misclassified points as a high number of points are in the dark coloured area and the uncertainty bounds are wide. Cluster 1 is not well separated from the other two clusters in both the first and second dimension as can be seen in Figures A.8 and A.9 in Appendix A. From the results of dataset P, LC clustering had the best cluster recovery followed by the K-means and hierarchical clustering.

4.3 Analysis results for data set M

4.3.1 Exploratory data analysis on M

Dataset M has 600 observations and 4 clusters whose sizes are 145, 64, 306 and 85. Box and whisker plots for the moderately separated clusters are shown in Figure B.1 in Appendix B. Variables **a** and **b** are nearly symmetrical with **c** and **d** showing signs of skewness. There are a lot of points beyond the extent of the whisker ends for all the variables which indicate possible outliers. The summary for dataset M in Appendix B.1 shows that variable **d** has the least variance of 380 while the largest is 791 for variable **b**.

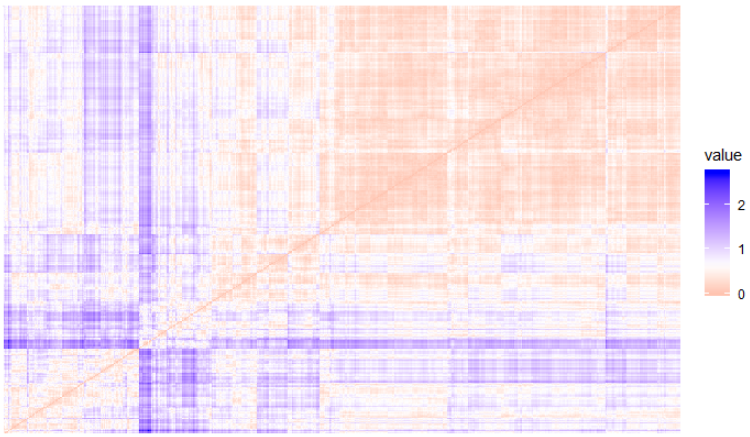

Figure 4.6: Distance matrix for M

The distance matrix in Figure 4.6 shows clusters of similar observations indicated by the orange squares along the diagonal. The histogram for variable **a** in Figure 4.7 has one dominant peak and two are less pronounced suggesting the possible presence of 3 clusters.

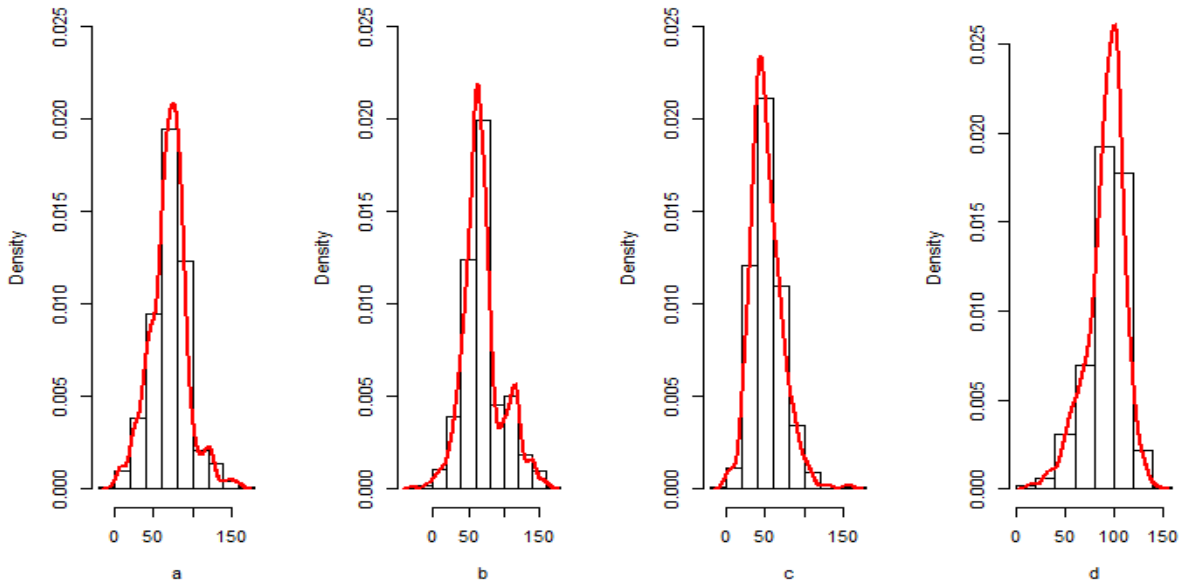

Figure 4.7: Histograms for variables in data set M

Variable **b** has 3 peaks indicating the possible presence of 3 clusters. Variables **c** and **d** have tiny peaks which are not well pronounced. These 2 variables might have less contribution in the cluster structure compared to variables **a** and **b**.

The pairs plot in Figure B.2 in Appendix B shows that variables **a** and **b** separate the 4 clusters well. In the other remaining plots there is fairly good separation of 3 of the 4 clusters. Table B.1 in Appendix B shows that the first 2 principal components account for about 64% of the variance in the data. The principal components plot in Figure 4.8 indicates that the first PC clearly separates clusters 1 and 2. Clusters 3 and 4 are poorly separated in the first PC. The second PC shows poor cluster separation.

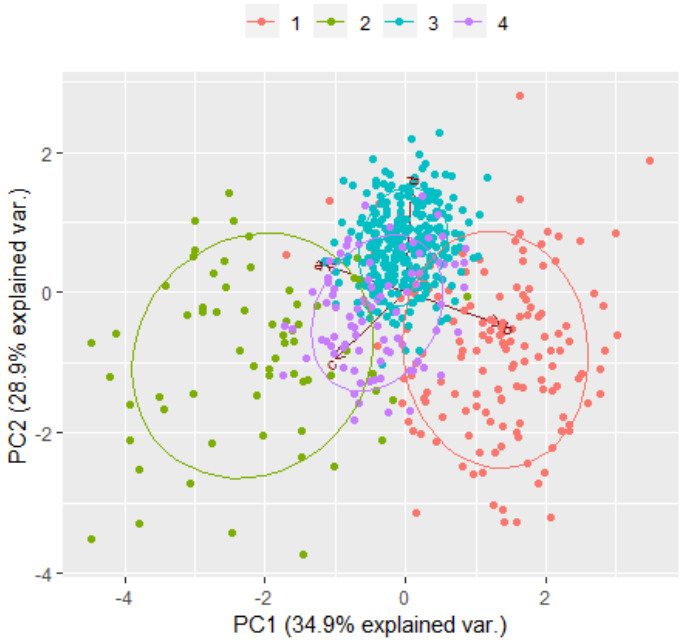

Figure 4.8: Principal components plot for M

4.3.2 Hierarchical clustering on M

Table B.2 in Appendix B shows hierarchical clustering results for data set M. Values of the cluster recovery indices are better than for dataset P. The best ARI achieved was 0.788 with Ward's method, Manhattan distance and standardisation by the maximum value. The dendrogram for this model is shown in Figure. Just like in dataset P, Ward's agglomeration out-performed all the other methods across all variations. All distance methods had ARI values of above 0.7 with Ward's agglomeration except for the Canberra distance which had lower values. The worst ARI value for the Canberra distance was 0.397 with a z-score standardisation.

The second best results (ARI =0.773) were achieved with no standardisation, Ward's agglomeration with both Minkowski and Euclidean distances. The single linkage had the worst results of all the linkages with its ARI values being almost zero. The complete linkage achieved a maximum ARI value of 0.519 for Euclidean distance with no standardisation; Manhattan distance with Z-score and no standardisation and Minkowski distance with no standardisation. The average linkage's best result (ARI= 0.512) was achieved with

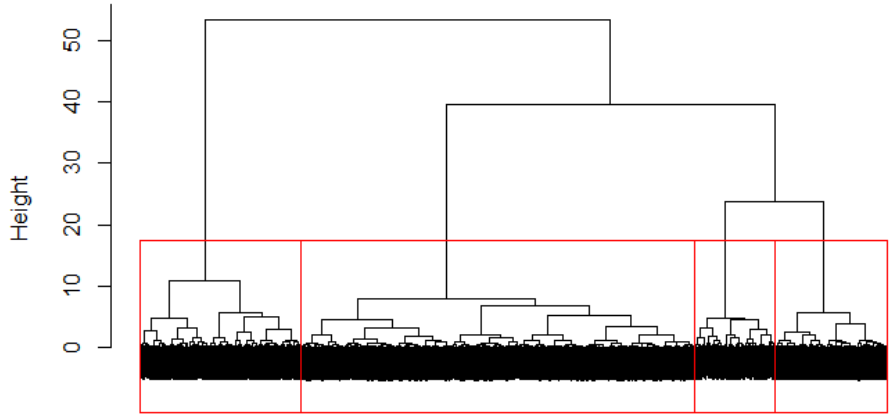

Figure 4.9: Dendrogram for best results for M

the Canberra distance and Z-score standardisation.

4.3.3 K-means clustering on M

Figure B.3 in Appendix B shows the optimal number of clusters to be three. The K-means algorithm was carried out for the 2, 3 and 4 cluster solutions and the results displayed in Table 4.2. All the models for the 2 cluster solution had ARI values of less than 0,4. The best result for the 3 cluster solution with an ARI value of 0.62 was with the Z-score standardisation. The four cluster solution achieved ARI values greater than 0.73 for all standardisation methods. The model with no standardisation had the best cluster recovery (ARI=0.78) closely followed by standardisation with the maximum value (ARI=0.778). The VI index values were lower for the 4 cluster solutions indicating that they resulted in the least loss of information. Based on the VI index, K-means with standardisation by the maximum had the best result followed by that with no standardisation.

Although the best results were achieved by the 4 cluster model, most indices were suggesting a three cluster model for dataset M. The 2-dimensional plot of the best solution, Figure 4.10, shows that clusters 1 and 2 overlap significantly and the cluster centres for both are in each other's ellipse. This explains why most indices were suggesting a three cluster solution.

Table 4.2: K-means and LC Results for data set M

			Rand	ARI	MA	FM	Jaccard	VI
K-means	2 clusters	No standardisation	0.621	0.280	0.282	0.617	0.424	1.242
		Z-score	0.648	0.337	0.338	0.653	0.460	1.281
		Range	0.631	0.310	0.311	0.642	0.446	1.268
		Maximum	0.625	0.291	0.292	0.625	0.432	1.222
	3 clusters	No standardisation	0.687	0.391	0.392	0.666	0.482	1.089
		Z-score	0.817	0.620	0.621	0.774	0.624	1.002
		Range	0.759	0.500	0.502	0.699	0.532	1.326
		Maximum	0.714	0.446	0.447	0.702	0.519	0.992
	4 clusters	No standardisation	0.899	0.780	0.781	0.859	0.752	0.662
		Z-score	0.882	0.743	0.744	0.835	0.717	0.726
		Range	0.878	0.736	0.737	0.832	0.711	0.762
		Maximum	0.898	0.778	0.778	0.857	0.750	0.660
LC	4 clusters	No standardisation	0.763	0.450	0.452	0.623	0.448	
	5 clusters	No standardisation	0.819	0.565	0.567	0.700	0.518	
	6 clusters	No standardisation	0.809	0.534	0.536	0.678	0.485	

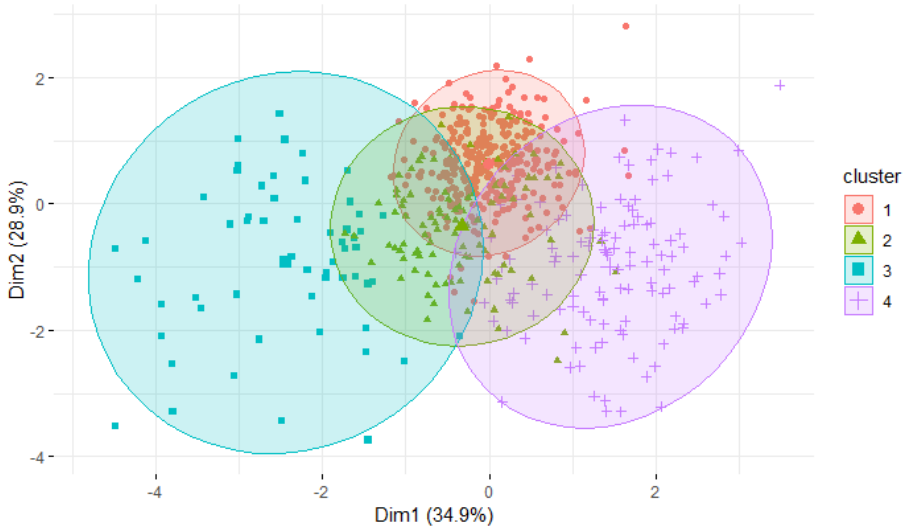

Figure 4.10: Cluster plot for K-means best result for M

4.3.4 Latent class cluster analysis on M

In LC cluster analysis BIC values are first generated to give possible models to consider. The top three models according to the BIC values were VII, VEI both with 5 clusters and VII with 6 clusters (see Appendix B.2). Surprisingly, a 4 cluster model did not appear in the top 3 models. The ICL values (Appendix B.2) indicated the presence of 2 possible clusters while the LRTS for VII showed 2 and LRTS for VEI indicated 5 clusters (Tables B.4 and B.3 in Appendix B). Changes in the BIC and ICL values across the across the 9 different components are shown in Figures B.4 and B.5 in Appendix B. A

2-dimensional plot (Figure B.6 in Appendix B) for the 5 cluster model shows 2 clusters with good separation while the other three remaining clusters have significant overlap.

Figure B.7 in Appendix B shows that there is a lot of uncertainty in the 3 clusters in the middle as all their observations lie in the uncertainty bounds. This is confirmed by the density plots in Figures B.8 and B.9 in Appendix B which show clusters 2, 3 and 5 not separating in the first dimension. Furthermore the second dimension does not separate any clusters at all. Model summaries for the 4 and 5 cluster solutions are given in Appendix B.3 and Appendix B.4. It seems most of the observations from clusters 1 and 2 in the three class model were misclassified to form the fifth cluster in the 5 class model.

4.4 Analysis results for data set W

4.4.1 Exploratory data analysis on W

Dataset W has 700 observations with three clusters of sizes 137, 184 and 379. The box and whisker plot for variable **b** shows that it is negatively skewed. All the other variables are nearly symmetrical with possible outliers also present. Variable **d** has the highest number of observations that might be outliers. The summary of the data in Appendix C.1 shows that variable **c** has the least variance of 259 while variable **b** has the largest variance of 848. The range of the variances in dataset W is larger compared to that of datasets P and M.

The distance matrix in Figure 4.11 has 3 distinctive orange squares along the diagonal indicating the presence of clusters. The histograms for dataset W in Figure 4.12 stand out from the previous ones because all the density curves either show more than one peak or have a distinctive bulge.

The histogram for variables **a** and **c** have a bulge to the right hand side of the peak while

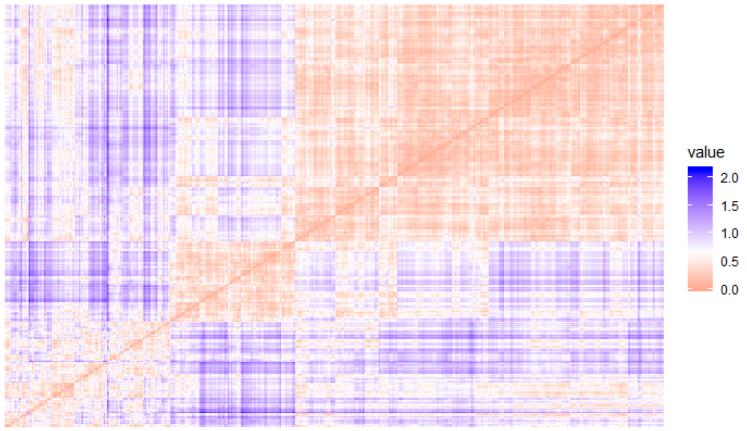

Figure 4.11: Distance matrix for W

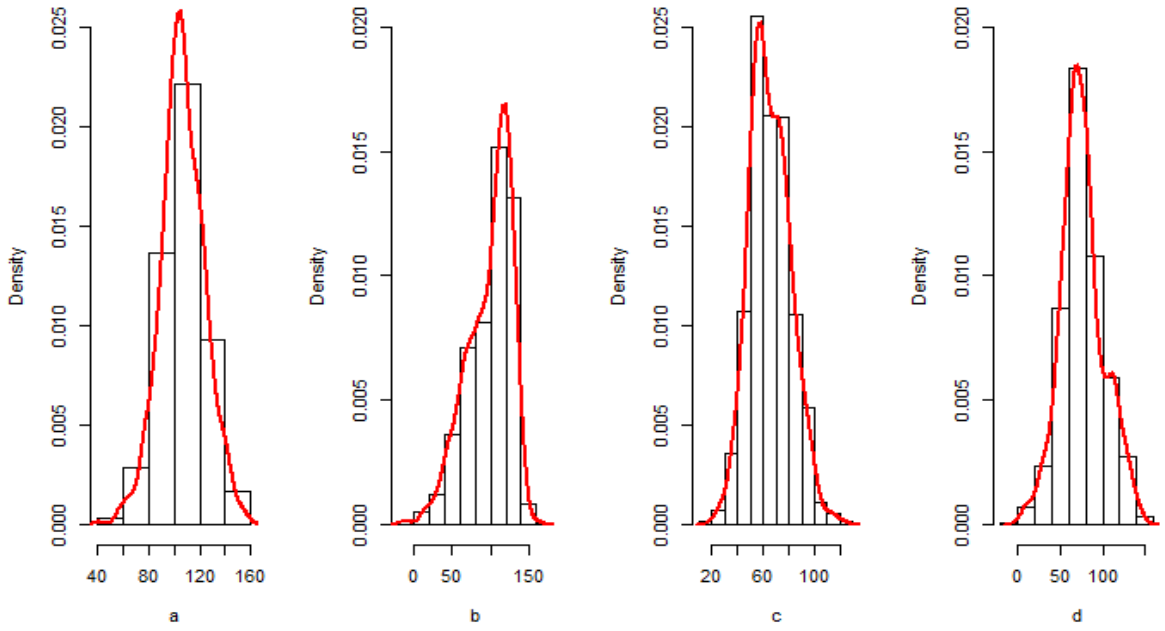

Figure 4.12: Histograms for variables in data set W

the one for variable **b** has a bulge on the left side of the peak. All the 4 charts seem to suggest the presence of 2 larger clusters. The pairs plot in Figure C.2 in Appendix C shows that variables **b** and **d** separate the three cluster well. On the other hand variables **a** and **c** are not showing any cluster separation.

The first two PCs account for 69.6% of the variation (Table C.1 in Appendix C). Thus the two dimensional plot is expected to give better information about the clusters compared to the ones for datasets P and M.

Figure 4.13 shows good cluster separation and supports the presence of 2 larger groups visible from the histograms. Clusters 1 and 3 are much closer together than cluster 2. The first PC separates cluster 1 from clusters 2 and 3 well, the second PC also separates cluster 2 and 3 very well. The plots show that there is little overlap between cluster 1 and the other clusters. The overlap between cluster 2 and 3 is also not extensive.

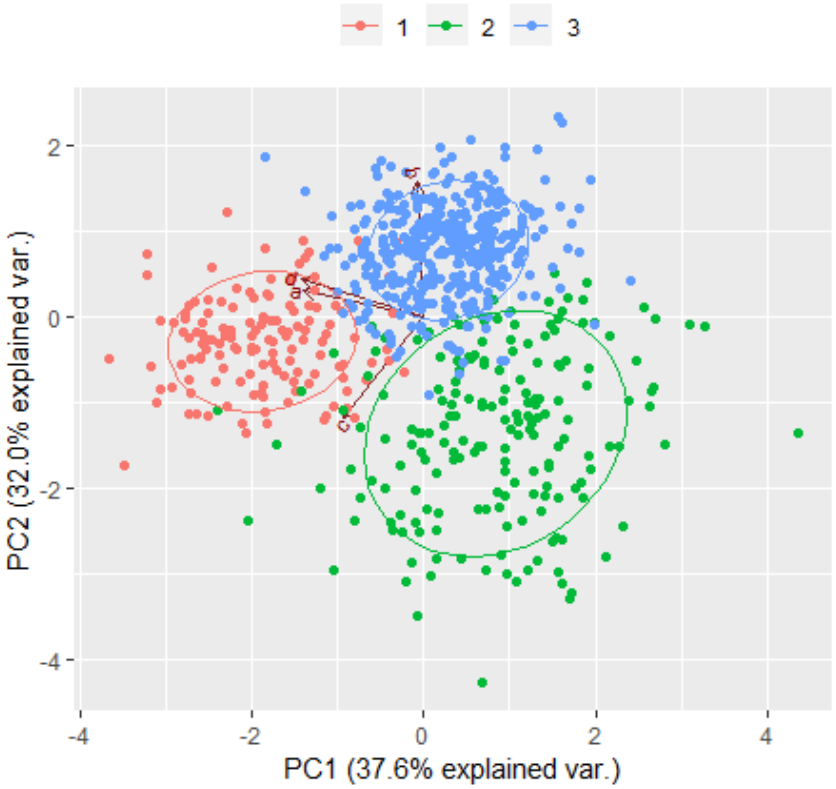

Figure 4.13: Principal components plot for W

4.4.2 Hierarchical clustering on W

Results for hierarchical clustering are displayed in Table C.2 in Appendix C. Ward's agglomeration method produced the best results for hierarchical cluster analysis. The best model was a 3 cluster solution (ARI=0.892) with Manhattan distance and standardisation by the maximum value. This was closely followed by the z-score standardisation with Manhattan distance (ARI=0.885). The dendrogram for this model in Figure 4.14 clearly shows the appropriateness of a 3 cluster solution.

Table 4.3: K-means and LC Results for data set W

			Rand	ARI	MA	FM	Jaccard	VI
K means	3 cluster	No standardisation	0.703	0.422	0.423	0.705	0.531	0.899
		Z-score	0.935	0.865	0.865	0.920	0.851	0.407
		Range	0.937	0.869	0.869	0.922	0.856	0.394
		Maximum	0.943	0.882	0.882	0.930	0.869	0.347
LC	3 cluster	No standardisation	0.961	0.918	0.919	0.951	0.907	
	4 cluster	No standardisation	0.833	0.630	0.631	0.764	0.597	

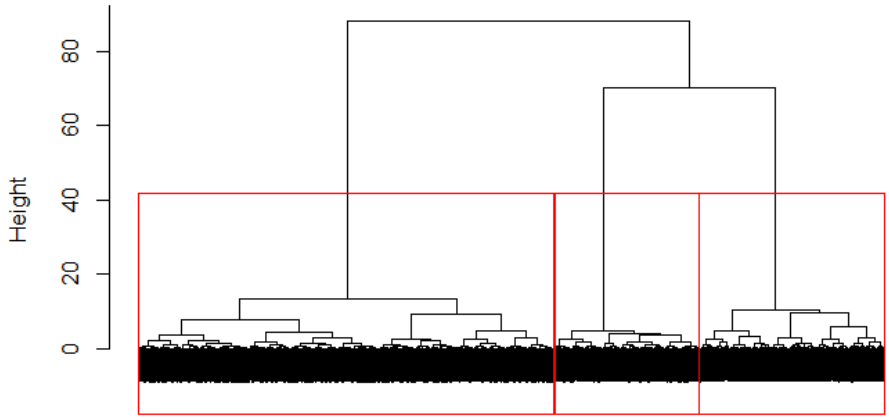

Figure 4.14: Dendrogram for best results for W

The single linkage algorithm produced the worst results with all ARI values being almost zero. The average linkage also produced extremely small (0.003 to 0.01) ARI values except for the Canberra distance and z-score standardisation which had a higher value (0.686). The complete linkage algorithm produced identical results for both the Euclidean and Minkowski distance with the best results (ARI=0.594) being with z-score standardisation.

4.4.3 K-means clustering on W

The majority of the NbClust indices (Figure C.3 in Appendix C) suggested 3 clusters for dataset W. The results for K-means clustering are displayed in Table 4.3. Standardisation by the maximum values gave the highest ARI (0.882), followed by standardisation by the range (0.869) , z-score (0.856) and lastly with no standardisation (0.422). The model with the highest ARI had the least VI index. The plot for the best model in Figure 4.15 shows little overlap between the clusters compared to the solutions for datasets P and M.

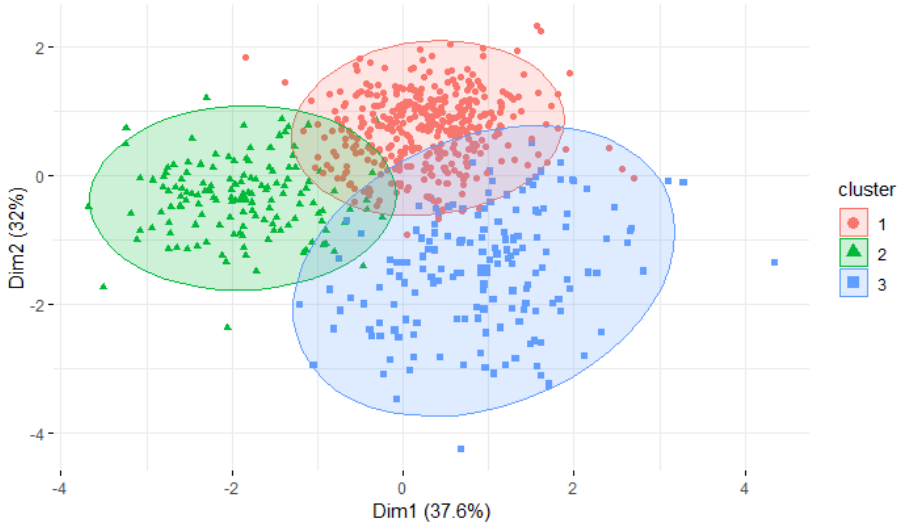

Figure 4.15: Cluster plot for K-means best result for W

4.4.4 Latent class cluster analysis on W

LCCA results for dataset W are displayed in Table 4.3. The VII model with 3 clusters had the best ARI value. The two dimensional plot for the model in Appendix C, Figure C.6 shows even less overlap between clusters than the plot for K-means clustering. The uncertainty plot (Figure C.7 in Appendix C) shows that most of the misclassified observations are closer to the uncertainty boundaries. For well separated clusters the uncertainty boundaries are thinner compared to those for P and M. The density plots in Figures C.8 and C.9 in Appendix C shows how well clusters 1 and 2 are separated from cluster 3 in the first dimension. Cluster 1 and 2 are then clearly separated in the second dimension.

The BIC, ICL (Appendix C.3) and LRTS (Table C.3 in Appendix C) all suggested a three cluster model for the data. This is also clearly visible from the graphs in Figures C.4 and C.5 in Appendix C. The cluster sizes of the model from the summary in Appendix C.2 are very close to the true cluster sizes. This supports the good recovery indicated by the ARI. Of the three methods, LCCA had the highest ARI value followed by hierarchical clustering and the K-means method was last.

4.5 Analysis results for Seeds data set

4.5.1 Exploratory data analysis on seeds data

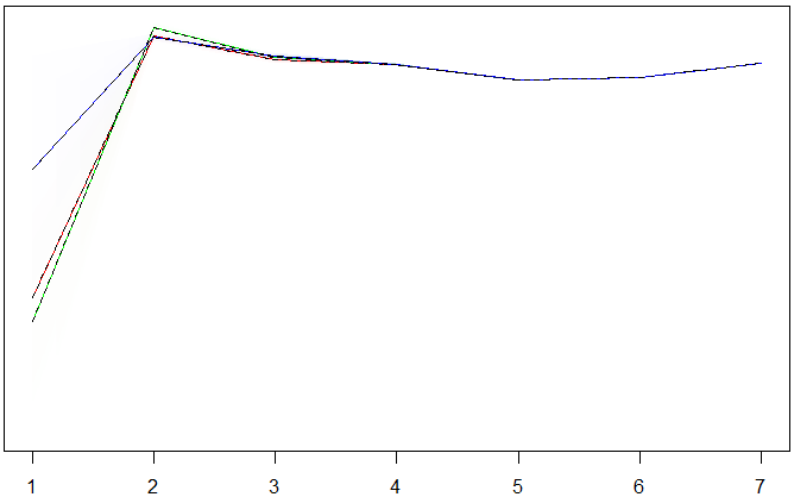

Figure 4.16: Parallel density plot for seeds data

The seeds data contains three clusters each with 70 observations. The MixSim package was used to estimate the overlap parameters of this data and the results are displayed in Appendix D.2. The average and maximum overlap were 0.014 and 0.033 respectively indicating well separated clusters.

Figure 4.16 shows a parallel density plot for the seeds data plotted in MixSim. Lines on a parallel density plot represent clusters in multidimensional data. The distance between the lines in each dimension indicates how well the clusters are separated in that dimension. The three clusters are well separated in the first dimension and two of them are expected to be closer to each other. In the second dimension there is little separation between the clusters and two of them almost overlap completely. The presence of two clusters closer to each other and a separate third is clearly visible in the distance matrix in Figure 4.17.

Figure D.1 in Appendix D shows the box and whisker plots for the seeds data. The distribution for compactness is nearly symmetrical. For the remaining variables there is some degree of skewness in the distributions. Asymmetry coefficient is the only variable with possible outliers, two observations. The variables in Seeds are on different scales, for

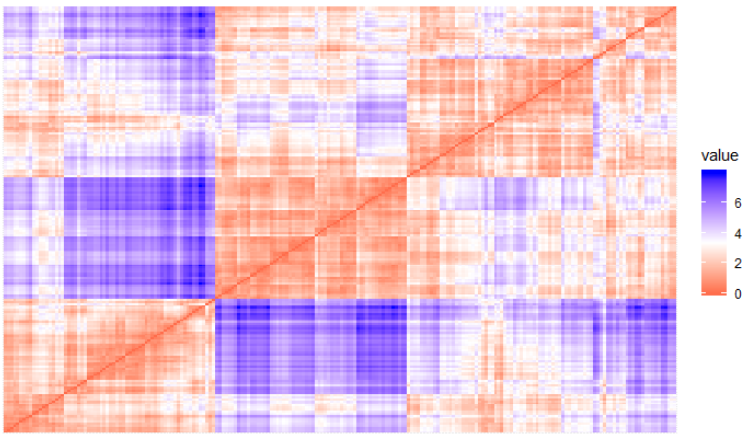

Figure 4.17: Distance matrix for S

instance compactness ranges between 1 and 186, while kernel width ranges between 2.6 and 4. As a result the variances also have a wider range compared to the other datasets considered (see Appendix D.1).

The histograms for area and perimeter have 2 peaks with a small bulge on one of the peaks indicating the possible presence of three clusters.The histogram for kernel groove length has two smooth peaks indicating two clusters. Kernel length histogram indicates two clusters with its single peak and bulge on the right hand side. Four bumps on the kernel width histogram suggest 4 clusters. The asymmetry coefficient histogram has two peaks which suggests at least 2 clusters. The flatness of the compactness histogram at the top indicates presence of clusters but because there are no visible peaks, there is no indication on the number of clusters.

PCA results in Table D.1 in Appendix D shows that the first PC accounts for about 72% of the variation in the data. This is seen in Figure 4.19 where the first PC separates the three clusters very well. Clusters 1 and 3 are located in the same region which explains why some of the histograms indicated the presence of 2 clusters. The second PC which explains 17% of the variation separates cluster 1 from clusters 2 and 3 although the separation is poor.

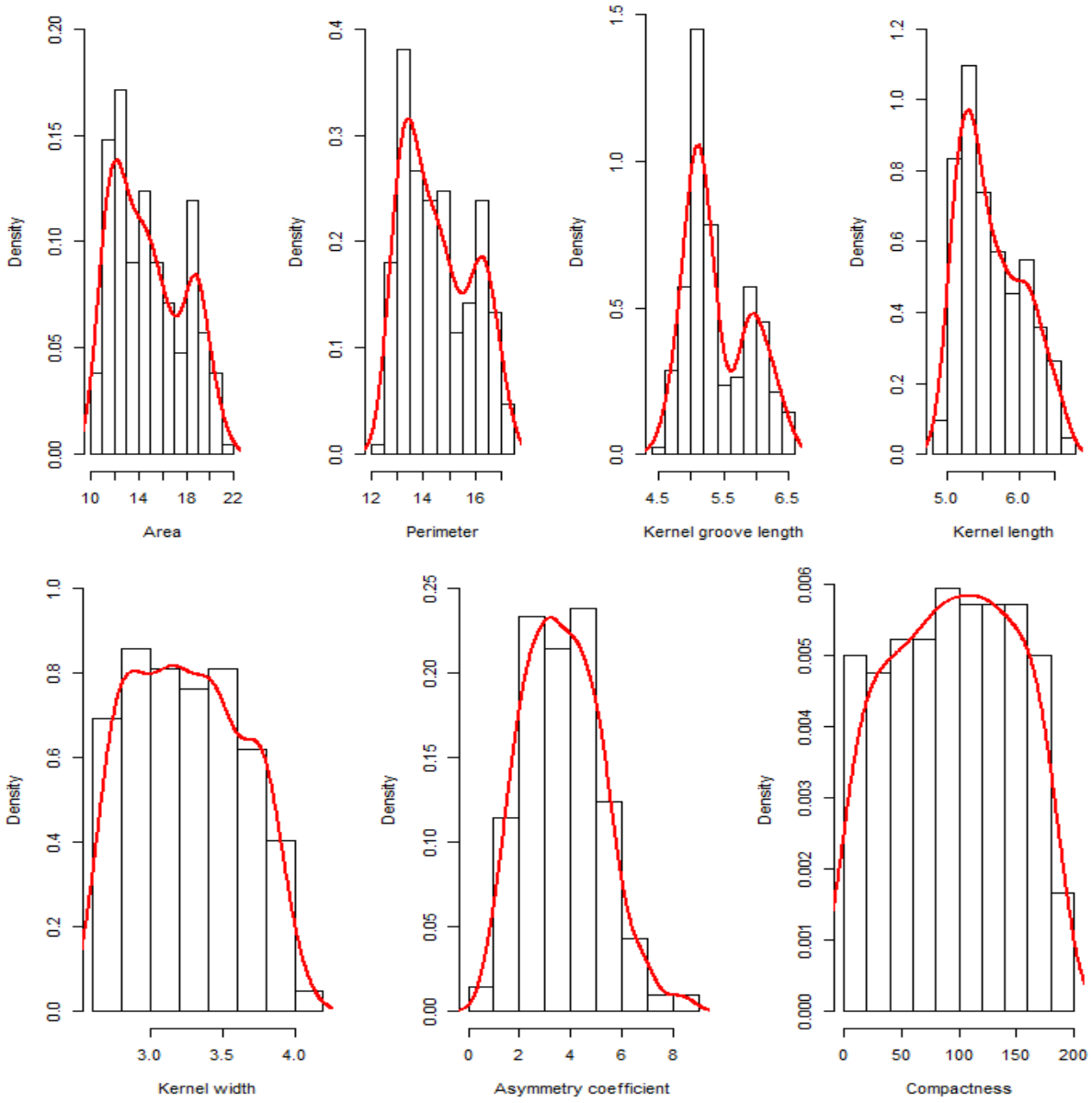

Figure 4.18: Histograms for variables in the seeds data set

In the pairs plot for the seeds data in Figure D.2 in Appendix D, most plots show good separation of the clusters. The only plot which does not show good separation is for compactness and asymmetric coefficient. Incidentally the histograms for these two variables do not have more pronounced peaks or bumps compared to the other five variables.

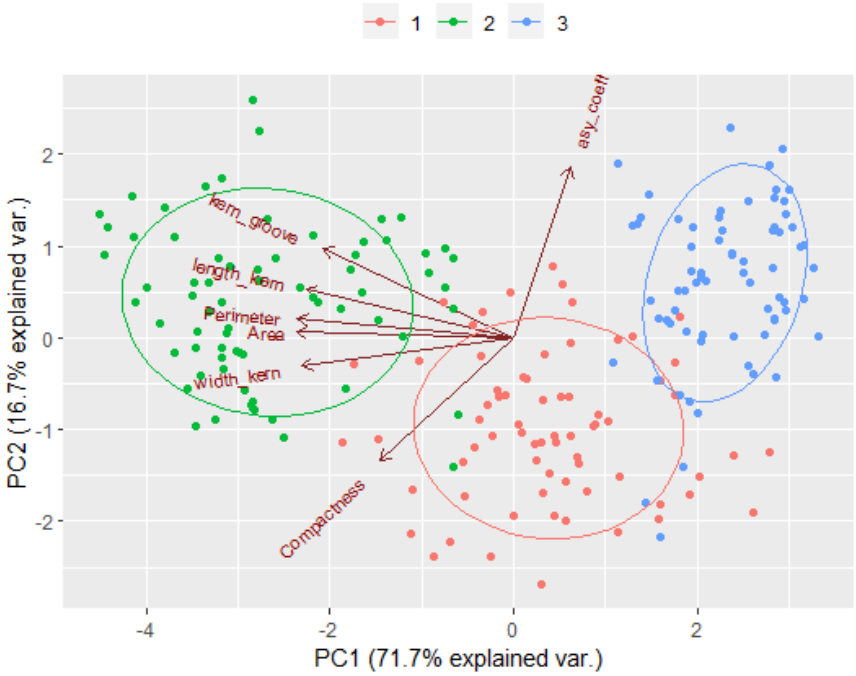

Figure 4.19: Principal components plot for seeds data

4.5.2 Hierarchical clustering on seeds data

The best hierarchical clustering results for the seeds data were achieved with Ward's agglomeration (see Figure 4.20 and Table D.2 in Appendix D). For Ward's method, both the Euclidean and Minkowski distances produced identical results across all standardisation methods, the largest ARI being 0.724 from the z-score standardisation. The second best results (ARI= 0.680) were obtained when the data was standardised by the range.

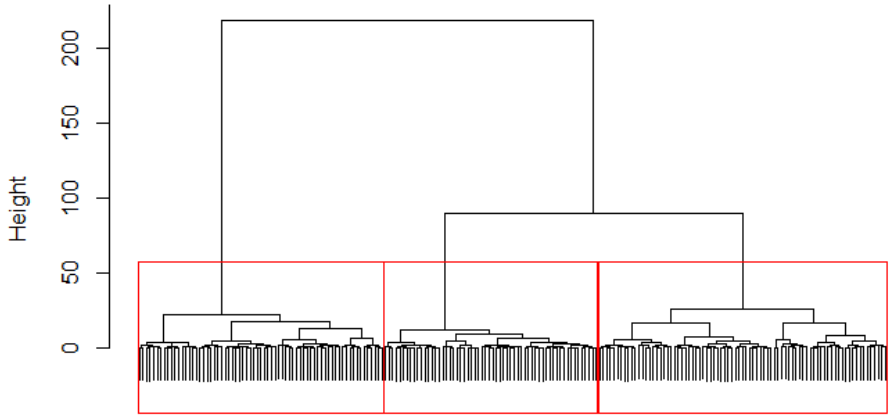

Figure 4.20: Dendrogram for best results for S

Table 4.4: K-means and LC Results for data set S

			Rand	ARI	MA	FM	Jaccard	VI
K means	2 cluster	No standardisation	0.587	0.172	0.177	0.511	0.334	1.503
		Z-score	0.752	0.513	0.516	0.731	0.551	0.692
		Range	0.743	0.493	0.496	0.716	0.536	0.748
		Maximum	0.632	0.272	0.277	0.579	0.394	1.304
	3 cluster	No standardisation	0.641	0.189	0.197	0.458	0.297	1.773
		Z-score	0.900	0.774	0.776	0.848	0.737	0.598
		Range	0.864	0.693	0.695	0.794	0.659	0.742
		Maximum	0.670	0.294	0.299	0.554	0.380	1.490
LC	3 cluster	No standardisation	0.893	0.758	0.761	0.839	0.723	
	4 cluster	No standardisation	0.860	0.683	0.686	0.787	0.649	

Again the single linkage method had the worst results with all ARI values being equal to zero. The complete linkage had a best ARI value of 0.642 with the Manhattan distance and standardising by the maximum value. The average linkage also had fairly good results with Manhattan distance and both z-score and range standardisation.

4.5.3 K-means clustering on seeds data

Figure D.3 in Appendix D shows the number of clusters given by NbClust indices. Seven indices suggested 2 clusters while ten suggested 3 clusters.

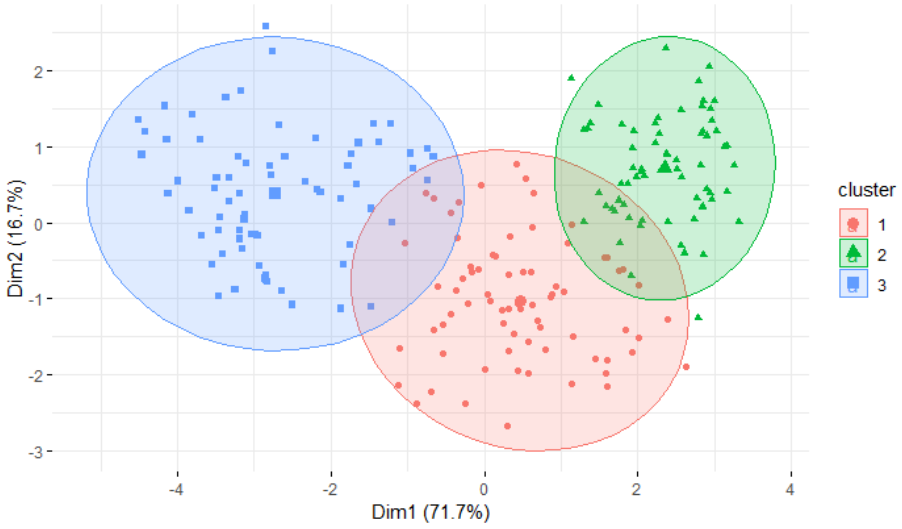

Figure 4.21: Cluster plot for K-means best result for S

Table 4.4 shows results for K-means and LC cluster analysis. K-means algorithm models for both 2 and 3 clusters were fitted and compared. The 3 cluster model had by far the

best ARI value of 0.774 with z-score standardisation. This model also had the lowest VI index. The second best result (ARI=0.693) was also a 3 cluster model with range standardisation. The 2-dimensional plot for the best solution in Figure 4.21 shows that there is little overlap between the clusters. The cluster centres are well separated with no centre being in a concentration ellipse of another cluster.

4.5.4 Latent class cluster analysis on seeds data

Figures D.4 and D.5 in Appendix D show that the BIC and ICL values for the top models are almost constant across all the number of components. Both the BIC and ICL (see Appendix D.5) values show that the best model is a 4 cluster (VEV) while a 3 cluster (VEV) solution is third best. The LRTS (Table D.3 in Appendix D) also found the 4 cluster model to be the best. Both models were fitted and the 3 class model (ARI=0.758) had better cluster recovery compared to the 4 cluster model (ARI=0.683). The model summaries for the 3 and 4 cluster solutions are shown in Appendix D.3 and D.4. In the 4 cluster model, the size of cluster 1 is five while the minimum size of the remaining 3 clusters is 51. This suggests that cluster 1 may contain misclassified observations.

Furthermore, a comparison of the contours and the uncertainty plots (Figures D.6, D.10, D.7 and D.11 in Appendix D) for the 2 solutions shows that, for the 3 cluster model there is good cluster separation and the majority of misclassified points are closer to the uncertainty boundaries. In the 4 cluster model plots, 2 of the 5 observations from cluster 1 are far away from all the other observations while the other 3 are on the intersection of clusters 2 and 4. The density plots for the 3 cluster solution (Figures D.8 and D.9 in Appendix D) show good cluster separation in both the first and second dimension. On the other hand, Figures D.12 and D.13 in Appendix D show poor separation in the first dimension while the second dimension looks better as clusters 3 and 4 are well separated. There is very little separation between clusters 1 and 2 in the second dimension.

4.6 Conclusion

Results from the data analysis were presented in this chapter. Some of the extra output is in the Appendices. The next Chapter discusses the results followed by the findings and recommendations.

Chapter 5

Summary, Conclusion and Recommendation

5.1 Introduction

Results from Chapter 4 are summarised by dataset in this chapter. Conclusions drawn from the findings are presented and then recommendations given. Limitations to the study are given at the end which give possible directions which future studies might take.

5.2 Summary

5.2.1 Poorly separated clusters (P)

Exploratory data analysis was done to check if clusters existed in the data. For the clusters with poor separation in P it was observed that evidence of the presence of clusters was less pronounced in the histogram and distance matrix. The first two PCs explained about 54% of the variation. The PCA plot showed a great extent of cluster overlap. Without the prior knowledge of cluster membership it would have been difficult to suspect the presence of clusters as no points were in isolated locations. The pairs plot also highlighted the poor clustering as there was no visible separation of the observations from the plots.

ARI values for hierarchical clustering were generally lower (at most 0.455), 0.744 was the maximum Rand index and 0.675 the maximum FM value. It can be misleading to look at the Rand and FM indices and conclude that cluster recovery is good when it is not. The ARI, MA and Jaccard truly reflect the poor cluster recovery.

Ward's method with range standardisation and Manhattan distance produced the best hierarchical clustering results. The Canberra distance had poor recovery when cluster overlap was high. The single, complete and average linkage all had poor cluster recovery. The NbClust indices had misleading results. The majority of the indices suggested 2 clusters instead of the actual 3. The best K-means result was with no standardisation.

The graph of the BIC values correctly shows a peak at the ideal number of clusters (k=3). The ICL results were not accurate as they indicated one cluster. Cluster plots for the models obtained by K-means and LC cluster analysis do not clearly show the distinct clusters although the one for LCCA is better. LC cluster analysis (ARI=0.617) had the best results of all three methods followed by K-means (0.505) and lastly hierarchical clustering (0.455).

5.2.2 Moderately separated clusters (M)

The histogram for two of the four variables showed distinct peaks indicating the presence of clusters. The orange squares on the distance matrix were more pronounced compared to that of P. The pairs plot for M is better than that for P because there was some degree of isolation of the clusters on the plot. Of the four clusters, one kept on appearing on top of the others in the pairs plot. Such a cluster would be difficult to identify when the membership is not known. PCA results of M were more encouraging with the first two PC's accounting for 64% of the variation.

The values for the cluster recovery indices were better than those obtained from P. In hierarchical clustering Ward's agglomeration gave the best cluster recovery overall. The maximum ARI was achieved with Manhattan distance and standardisation by the maximum value. The Canberra distance had the lowest ARI values. The NbClust indices were not accurate, suggesting 3 instead of 4 clusters. The best K-means results were with no standardisation closely followed by standardisation by the maximum value.

BIC values were not accurate suggesting 5 clusters while the 4 cluster solutions were not even in the top three models. The ICL and LRTS were also not reliable in determining the number of clusters. Hierarchical clustering produced the best results followed by the K means algorithm, both models with 4 clusters. LCCA had the worst results and its best model was a five cluster solution.

5.2.3 Well separated clusters (W)

The 3 orange squares on the distance matrix are more pronounced than those for datasets P and M. Although each histogram has no multiple peaks, they each have significant bumps on the sides which make them non smooth. Some variables clearly show good cluster separation in the pairs plot. The first two PC's account for 70% of the variation and the PC plot shows clear separation of the clusters.

Ward's method performed well in hierarchical clustering and the best (ARI=0.892) result was with Manhattan distance and standardisation by the maximum value. The single linkage had the worst results. The optimal number of clusters was predicted correctly by the NbClust indices with the majority indicating three clusters. For the K-means algorithm, standardisation by the maximum value produced the best (ARI=0.882) results. The 2-dimensional plot for the 3 cluster model from the K-means algorithm had less cluster overlap. LCCA had the best (ARI=0.918) cluster recovery results which is reflected in the 2-dimensional plots with even less overlap compared to that of the

K-means model. For well separated clusters, the BIC, ICL and LRTS were all accurate in stating the correct number of clusters.

5.2.4 Seeds data set

The average and maximum overlap for seeds data estimated in MixSim were both less than 0.05 indicating that the clusters are well separated. The parallel density plot shows three clusters in the first dimension and that two of them are expected to be closer to each other. The distance matrix also clearly reflects the same structure of clusters as seen in the parallel density plot. Five of the seven histogram have multiple peaks or flatter tops indicating the presence of clusters. The first two PC's account for 72% of the variation, this is well reflected in the PC plot as the three clusters are well separated.

In hierarchical clustering, Ward's method again had the best cluster recovery. Both Euclidean and Minkowski distance with z-score standardisation had the best (ARI=0.724) recovery. The optimal number of clusters were correctly predicted by the NbClust indices. The K-means 3 cluster model plot shows less overlap confirming the overlap values estimated in MixSim. BIC values indicated a 4 cluster model instead of a 3 cluster one. Upon fitting, the 3 cluster model had the best cluster recovery.

5.3 Conclusion

5.3.1 Objective 1

LCCA performed well for poorly separated clusters compared to K-means and hierarchical clustering. For moderately separated clusters hierarchical clustering gave the best results and LCCA the worst. For well separated data LCCA had the best cluster recovery, followed by hierarchical and then the K-means clustering. The Seeds dataset which is well separated had the best recovery with K-means followed by LC and then hierarchical clustering.

5.3.2 Objective 2

K-means with no standardisation performed well for poorly separated clusters, variables with comparable variance. Although all results for the moderately separated clusters were similar, the K-means with no standardisation had the best results. For well separated clusters, K-means with standardisation by the maximum value had best cluster recovery. The seeds data which has well separated clusters had the best cluster recovery with a z-score standardisation.

5.3.3 Objective 3

Ward's agglomeration method has the best results for all levels of cluster overlap, single linkage method has worst. The Canberra distance does not perform well for poorly and moderately separated clusters, for well separated clusters, its results are comparable to other distance measures. Manhattan distance generally has good cluster recovery across all different cluster overlap. For well separated clusters all distance measures and standardisation methods have good cluster recovery and in addition the range and maximum value standardisation out-performed the other standardisation methods. When some variables have large variance (seeds data), hierarchical clustering with no standardisation did not perform well except when used with the Canberra distance. The z-score standardisation out-performed the other methods in this case.

5.3.4 Other findings

In the research a lot of results were consistent with what is found in literature. EDA gave a true reflection of the expected clusters in the data. For poorly clustered data the distance matrix, histograms, pairs plots and PCA plots did not show the presence of clusters clearly. In the well separated clusters, these plots showed the presence of clusters well. Therefore, EDA is a necessary step before carrying out the actual analysis.

The results also confirmed that the ARI is a better measure of cluster recovery than the

RI, FM, MA and Jaccard indices. Its values are not inflated so it is easy to make relative comparisons between different clustering algorithms. The RI had value of around 0.6 even though the cluster recovery was poor.

Ward's method produced the best results, which is consistent with literature. The standardisation by the range produced good results which agrees with the findings of Milligan and Cooper (1988). For the data sets whose variables had low variation, clustering with no standardisation sometimes produced good results. This confirms that standardisation might not be essential when there is low variation in the variables.

The HA, MA and Jaccard indices give a better measure of cluster recovery while the Rand and FM indices give high values which give a false impression of good cluster recovery. For poorly and moderately separated clusters most NbClust indices used to determine the number of cluster were misleading.

In the study by Magidson and Vermunt (2002), LC performed better than the K-means algorithm. In this study however, the results were mixed depending on the degree cluster of cluster overlap. For the Seeds and moderately separated data sets the K-means algorithm performed better than LC.

5.4 Recommendation

The findings of the research emphasises the need for the researcher to carry out EDA. Histograms and distance matrix give an insight of how well the clusters are separated. A summary of the variables assists in determining if there is greater variation between the variables. The dendrogram clearly shows the levels of cluster formation, therefore should be used to better understand the clusters in the data. For example, Figure 4.9 illustrates visually why some validity indices suggested 3 or 4 clusters for dataset M.

Ward's method performed well in hierarchical clustering. When trying out different

linkages it is a good idea to include it. The ARI, MA and Jaccard indices are good measures to use when assessing cluster recovery because their values are not inflated like the Rand and FM index. When carrying out research these indices can be relied upon to give an accurate extent of cluster recovery.

The research affirms challenges that are associated with cluster analysis. As cluster membership is unknown, it is more challenging for one to determine the correct number of clusters. Although LCCA is more of a modern technique, the traditional hierarchical and K-means clustering still have a role to play as they sometimes outperform LCCA. Therefore in carrying out clustering, the researcher must not abandon the traditional techniques but use them to complement LC analysis.

5.5 Limitations

This research explored a few datasets with limited variation. Possible avenues to explore will be to use different seeds for each simulated dataset to get several variations in the cluster layout. Several values of the average and maximum overlap could also be used for each broad category like moderately separated clusters. A full simulation study may also be necessary to eliminate sampling bias that may influence results.

Bibliography

Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian Clustering. *Biometrics*, pages 803–821.

Charrad, M., Ghazzali, N., Boiteau, V., and Niknafs, A. (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software*, 61(6):1–36.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., and Żak, S. (2010). Complete Gradient Clustering Algorithm for Features Analysis of X-ray Images. In *Information Technologies in Biomedicine*, pages 15–24. Springer.

Collins, L. M. and Lanza, S. T. (2010). *Latent Class and Latent Transition Analysis: With Applications in the Social, Behavioral, and Health Sciences*. John Wiley & Sons, Hoboken, NJ, 1st edition.

Cover, T. M. and Thomas, J. A. (2012). *Elements of Information Theory*. John Wiley & Sons, Hoboken, New Jersey, 2nd edition.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

DiStefano, C. and Kamphaus, R. (2006). Investigating Subtypes of Child Development: A Comparison of Cluster Analysis and Latent Class Cluster Analysis in Typology Creation. *Educational and Psychological Measurement*, 66(5):778–794.

Dua, D. and Graff, C. (2017). UCI Machine Learning Repository @ONLINE. `http://archive.ics.uci.edu/ml`.

Eshghi, A., Haughton, D., Legrand, P., Skaletsky, M., and Woolford, S. (2011). Identifying Groups: A Comparison of Methodologies. *Journal of Data Science*, 9(2):271–292.

Everitt, B., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. Wiley, Chichester, 5 th edition.

Fraley, C. and Raftery, A. E. (2007). Model-based Methods of Classification: Using the Mclust Software in Chemometrics. *Journal of Statistical Software*, 18(6):1–13.

Gates, A. J. and Ahn, Y.-Y. (2017). The Impact of Random Models on Clustering Similarity. *The Journal of Machine Learning Research*, 18(1):3049–3076.

George, G. and Parthiban, L. (2014). Cluster Validity Measurement Techniques. *International Journal of Computational Intelligence*, pages 91–96.

Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., and Tatham, R. L. (2006). *Multivariate Data Analysis*. Pearson Prentice Hall, Upper Saddle River, N.J, 6th edition.

Hubert, L. and Arabie, P. (1985). Comparing Partitions. *Journal of Classification*, 2(1):193–218.

Johnson, R. A. and Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*. Prentice Hall, Upper Saddle River, N.J., 6 th edition.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Hoboken, N.J, 1st edition.

Kent, P., Jensen, R. K., and Kongsted, A. (2014). A comparison of three clustering methods for finding subgroups in MRI, SMS or clinical data: SPSS Two-Step Cluster analysis, Latent Gold and SNOB. *BMC Medical Research Methodology*, 14(1):113.

Leisch, F. (2004). Flexmix: A General Framework for Finite Mixture Models and Latent Class Regression in R. *Journal of Statistical Software*, 11(8):1–18.

Magidson, J. and Vermunt, J. (2002). Latent Class Models for Clustering: A Comparison with K-means. *Canadian Journal of Marketing Research*, 20(1):36–43.

Maitra, R. and Melnykov, V. (2010). Simulating Data to Study Performance of Finite Mixture Modeling and Clustering Algorithms. *Journal of Computational and Graphical Statistics*, 19(2):354–376.

Masyn, K. E. (2013). Latent Class Analysis and Finite Mixture Modeling. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*, page 551. Oxford University Press.

Meilă, M. (2003). Comparing Clusterings by the Variation of Information. In *Learning Theory and Kernel Machines*, pages 173–187. Springer.

Melnykov, V., Chen, W.-C., and Maitra, R. (2012). *MixSim*: An R Package for Simulating Data to Study Performance of Clustering Algorithms. *Journal of Statistical Software*, 51(12):1.

Milligan, G. W. and Cooper, M. C. (1986). A Study of the Comparability of External Criteria for Hierarchical Cluster Analysis. *Multivariate Behavioral Research*, 21(4):441–458.

Milligan, G. W. and Cooper, M. C. (1988). A Study of Standardization of Variables in Cluster Analysis. *Journal of Classification*, 5(2):181–204.

Morey, L. C. and Agresti, A. (1984). The Measurement of Classification Agreement: An Adjustment to the Rand Statistic for Chance Agreement. *Educational and Psychological Measurement*, 44(1):33–37.

Rama, B., Jayashree, P., and Jiwani, S. (2010). A Survey on Clustering Current Status and Challenging Issues. *International Journal on Computer Science and Engineering*, 2(9):2976–2980.

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. d. F., and Rodrigues, F. A. (2019). Clustering Algorithms: A Comparative Approach. *PLoS ONE*, 14(1):e0210236.

Rokach, L. and Maimon, O. (2005). Clustering Methods. In *Data Mining and Knowledge Discovery Handbook*, pages 321–352. Springer.

Rupp, A. A. (2013). Clustering and Classification. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*, page 517. Oxford University Press.

Santos, J. M. and Embrechts, M. (2009). On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification. In *International Conference on Artificial Neural Networks*, pages 175–184. Springer.

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). *mclust 5*: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal*, 8(1):289.

Steinley, D. (2004). Standardizing Variables in K-means Clustering. In *Classification, Clustering, and Data Mining Applications*, pages 53–60. Springer.

Vermunt, J. K. and Magidson, J. (2002). Latent Class Cluster Analysis. *Applied Latent Class Analysis*, 11:89–106.

Wagner, S. and Wagner, D. (2007). *Comparing Clusterings: An Overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe.

Appendix A

Results for poorly separated data, P

Appendix A.1: Summary and PCA results for P

```
Summary: P

class         a                b                c                d                e
 1:149   Min.   :  2.0   Min.   :-25.7   Min.   : 21.4   Min.   :  9.3   Min.   : 20.4
 2:113   1st Qu.: 51.9   1st Qu.: 28.6   1st Qu.: 65.2   1st Qu.: 76.5   1st Qu.: 84.5
 3:238   Median : 69.0   Median : 44.1   Median : 80.3   Median : 98.1   Median :104.0
         Mean   : 68.5   Mean   : 44.1   Mean   : 81.2   Mean   : 94.5   Mean   :102.9
         3rd Qu.: 84.6   3rd Qu.: 58.7   3rd Qu.: 97.1   3rd Qu.:113.8   3rd Qu.:122.9
         Max.   :137.2   Max.   :116.9   Max.   :158.8   Max.   :171.1   Max.   :172.3

Variance: P

> var(mix1$a)
[1] 571.11
> var(mix1$b)
[1] 540.64
> var(mix1$c)
[1] 530.50
> var(mix1$d)
[1] 773.81
> var(mix1$e)
[1] 664.34

PCA results: P

Standard deviations (1, .., p=5):
[1] 1.296 1.007 0.921 0.874 0.833

Rotation (n x k) = (5 x 5):
     PC1    PC2     PC3      PC4     PC5
a -0.539  0.240 -0.0791 -0.10713 -0.7966
b -0.363 -0.660  0.2852  0.58964 -0.0609
c  0.461 -0.269  0.6425 -0.37014 -0.4066
d  0.398 -0.481 -0.7003  0.00956 -0.3458
e -0.455 -0.451 -0.0957 -0.70976  0.2770


Importance of components:
                        PC1   PC2   PC3   PC4   PC5
Standard deviation     1.296 1.007 0.921 0.874 0.833
Proportion of Variance 0.336 0.203 0.170 0.153 0.139
Cumulative Proportion  0.336 0.539 0.708 0.861 1.000
```

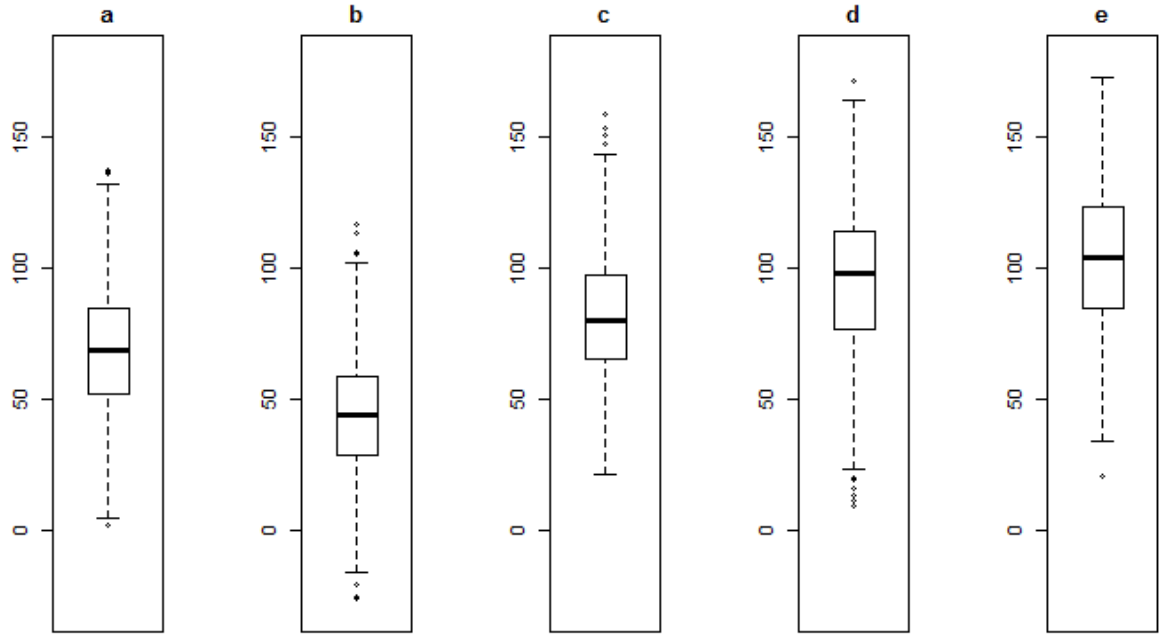

Figure A.1: Box and whisker plots for variables in data set P

Table A.1: PCA results for data set P

	PC1	PC2	PC3	PC4	PC5
Standard deviation	1.296	1.007	0.921	0.874	0.833
Percentage of Variance	33.6	20.3	17.0	15.3	13.9
Cumulative Percentage	33.6	53.9	70.8	86.1	100.0

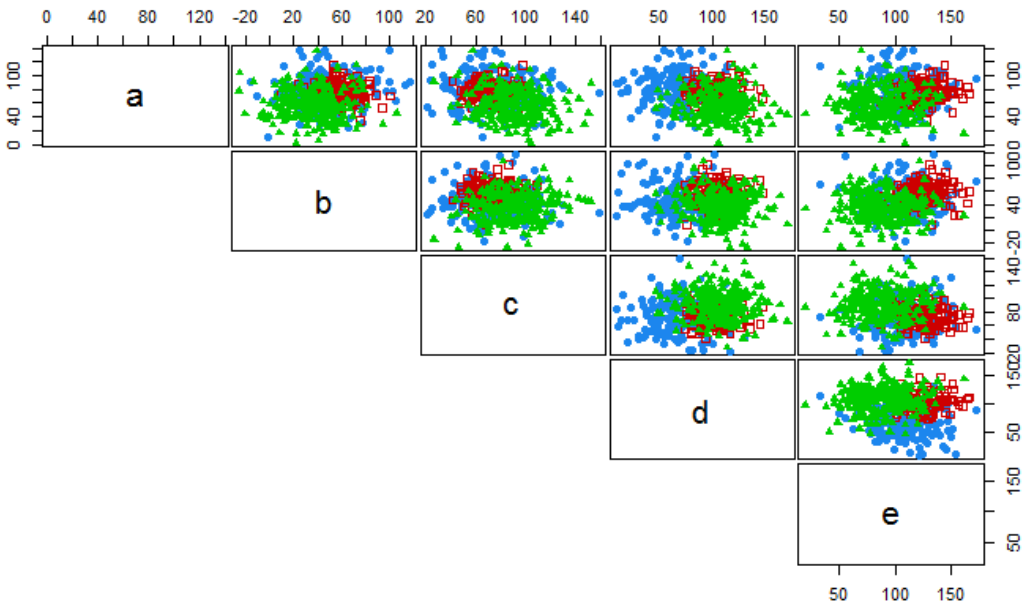

Figure A.2: Pairs plot for P

Table A.2: Hierarchical clustering results for data set P

Agglomeration	Distance	Standardisation	Rand	ARI	MA	FM	Jaccard
Ward	Euclidean	No standardisation	0.729	0.403	0.406	0.612	0.440
		Z- score	0.706	0.356	0.358	0.582	0.410
		Range	0.744	0.452	0.454	0.656	0.488
		Maximum	0.696	0.354	0.356	0.598	0.426
	Manhattan	No standardisation	0.740	0.437	0.439	0.641	0.472
		Z- score	0.736	0.427	0.429	0.633	0.463
		Range	0.710	0.401	0.402	0.643	0.471
		Maximum	0.746	0.455	0.457	0.657	0.489
	Canberra	No standardisation	0.657	0.267	0.270	0.541	0.371
		Z- score	0.612	0.160	0.163	0.465	0.303
		Range	0.657	0.267	0.270	0.541	0.371
		Maximum	0.657	0.267	0.270	0.541	0.371
	Minkowski	No standardisation	0.729	0.403	0.406	0.612	0.440
		Z- score	0.706	0.356	0.358	0.582	0.410
		Range	0.744	0.452	0.454	0.656	0.488
		Maximum	0.696	0.354	0.356	0.598	0.426
Single	Euclidean	No standardisation	0.366	-0.003	-0.003	0.600	0.363
		Z- score	0.366	-0.004	-0.004	0.599	0.362
		Range	0.366	-0.004	-0.004	0.599	0.362
		Maximum	0.367	-0.002	-0.002	0.600	0.363
	Manhattan	No standardisation	0.366	-0.003	-0.003	0.600	0.363
		Z- score	0.366	-0.003	-0.003	0.600	0.363
		Range	0.366	-0.003	-0.003	0.600	0.363
		Maximum	0.366	-0.003	-0.003	0.600	0.363
	Canberra	No standardisation	0.367	-0.001	0.000	0.602	0.364
		Z- score	0.368	0.000	0.000	0.602	0.364
		Range	0.367	-0.001	0.000	0.602	0.364
		Maximum	0.367	-0.001	0.000	0.602	0.364
	Minkowski	No standardisation	0.366	-0.003	-0.003	0.600	0.363
		Z- score	0.366	-0.004	-0.004	0.599	0.362
		Range	0.366	-0.004	-0.004	0.599	0.362
		Maximum	0.367	-0.002	-0.002	0.600	0.363
Complete	Euclidean	No standardisation	0.673	0.293	0.295	0.550	0.379
		Z- score	0.612	0.200	0.202	0.523	0.352
		Range	0.613	0.153	0.156	0.451	0.291
		Maximum	0.636	0.221	0.224	0.510	0.343
	Manhattan	No standardisation	0.574	0.168	0.169	0.542	0.362
		Z- score	0.571	0.160	0.162	0.536	0.357
		Range	0.586	0.150	0.152	0.496	0.327
		Maximum	0.646	0.281	0.283	0.583	0.406
	Canberra	No standardisation	0.375	-0.015	-0.015	0.575	0.349
		Z- score	0.487	0.018	0.020	0.471	0.297
		Range	0.375	-0.015	-0.015	0.575	0.349
		Maximum	0.375	-0.015	-0.015	0.575	0.349
	Minkowski	No standardisation	0.673	0.293	0.295	0.550	0.379
		Z- score	0.612	0.200	0.202	0.523	0.352
		Range	0.613	0.153	0.156	0.451	0.291
		Maximum	0.636	0.221	0.224	0.510	0.343
Average	Euclidean	No standardisation	0.384	0.011	0.011	0.597	0.364
		Z- score	0.368	-0.007	-0.007	0.592	0.358
		Range	0.388	0.012	0.013	0.594	0.363
		Maximum	0.367	-0.003	-0.003	0.598	0.362
	Manhattan	No standardisation	0.367	-0.003	-0.003	0.598	0.362
		Z- score	0.368	-0.006	-0.006	0.594	0.359
		Range	0.369	-0.005	-0.005	0.593	0.359
		Maximum	0.367	-0.003	-0.003	0.598	0.362
	Canberra	No standardisation	0.367	-0.001	0.000	0.602	0.364
		Z- score	0.631	0.260	0.262	0.578	0.400
		Range	0.367	-0.001	0.000	0.602	0.364
		Maximum	0.367	-0.001	0.000	0.602	0.364
	Minkowski	No standardisation	0.384	0.011	0.011	0.597	0.364
		Z- score	0.368	-0.007	-0.007	0.592	0.358
		Range	0.388	0.012	0.013	0.594	0.363
		Maximum	0.367	-0.003	-0.003	0.598	0.362

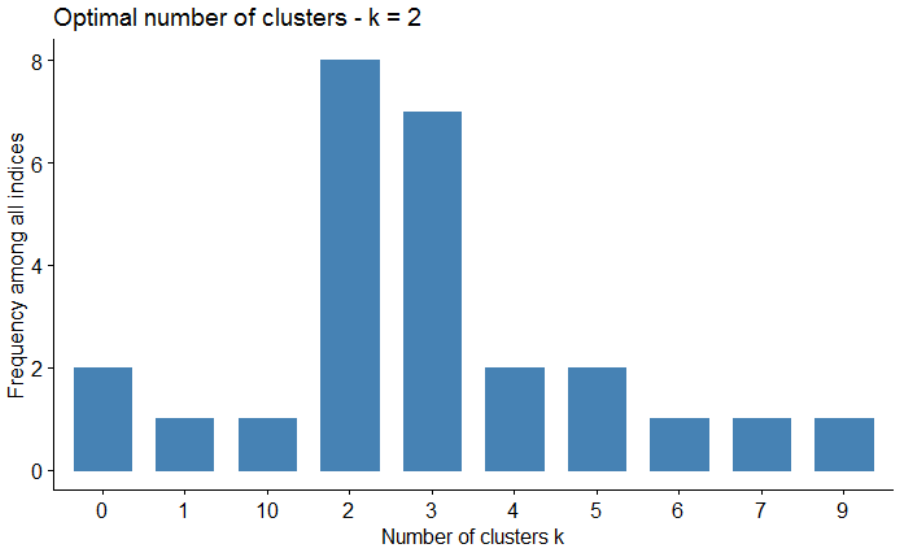

Figure A.3: Optimal number of clusters for P

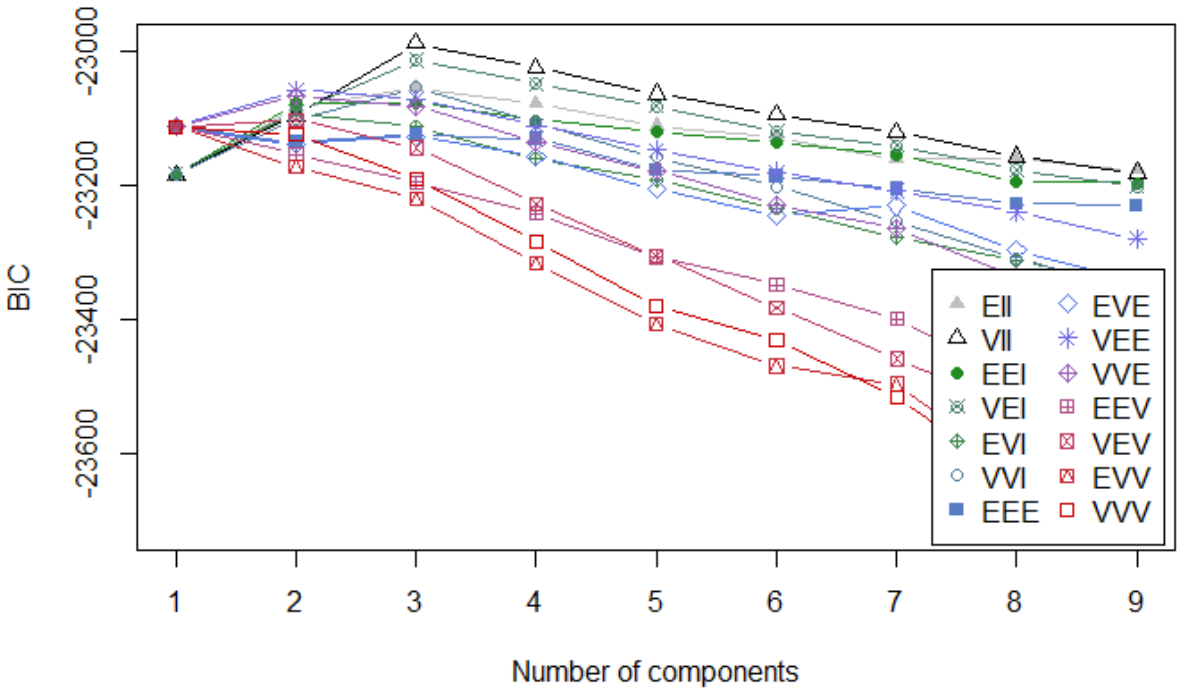

Figure A.4: BIC plot for P

Table A.3: Bootstrap sequential LRTS for P

Clusters	LRTS	Bootstrap p-value
1 vs 2	132.68	0.01
2 vs 3	150.61	0.01
3 vs 4	7.07	0.43

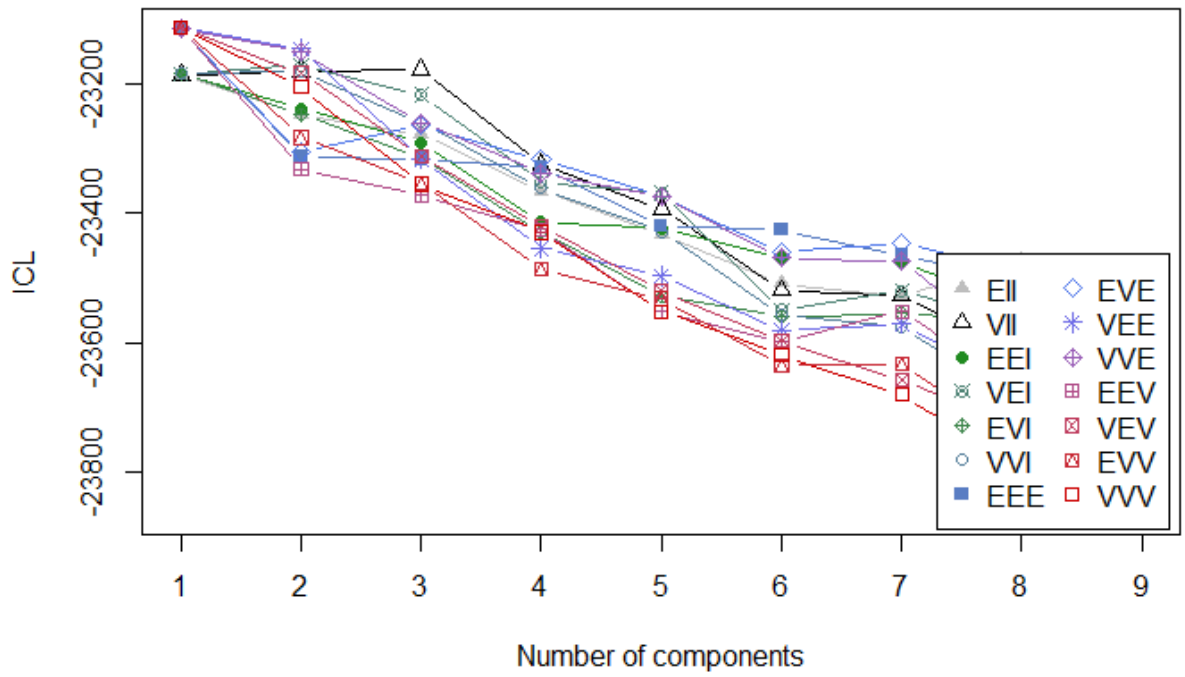

Figure A.5: ICL plot for P

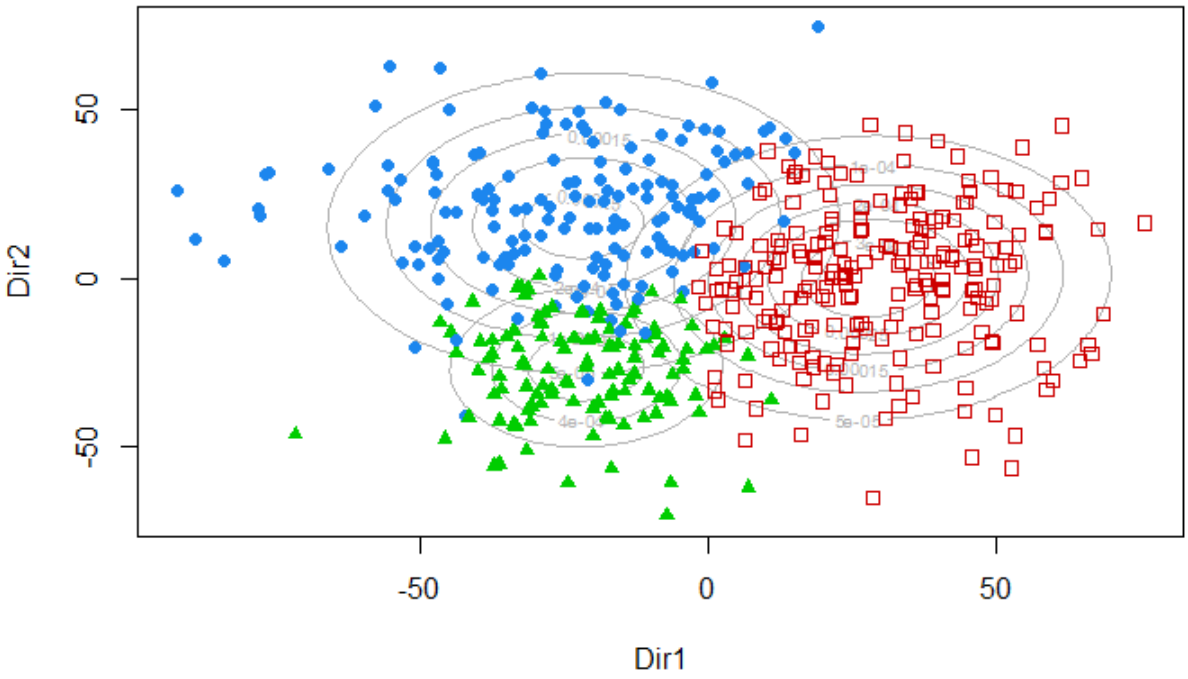

Figure A.6: LC cluster analysis: Contours plot for P

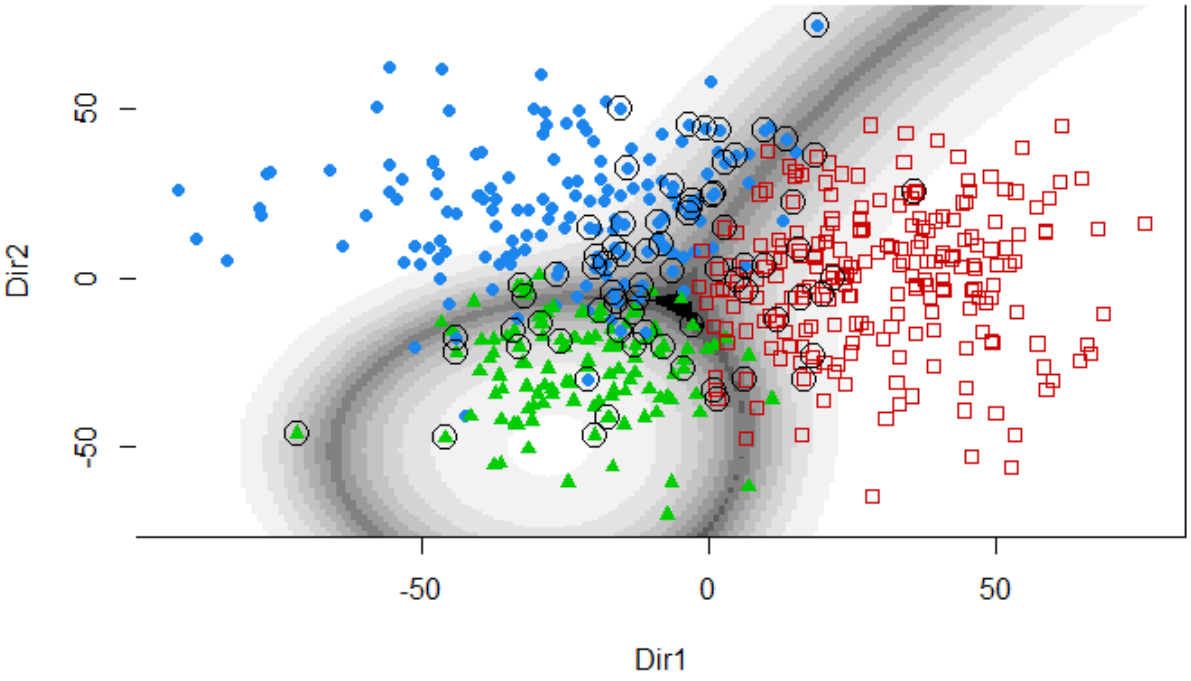

Figure A.7: LC cluster analysis: Uncertainty plot for P

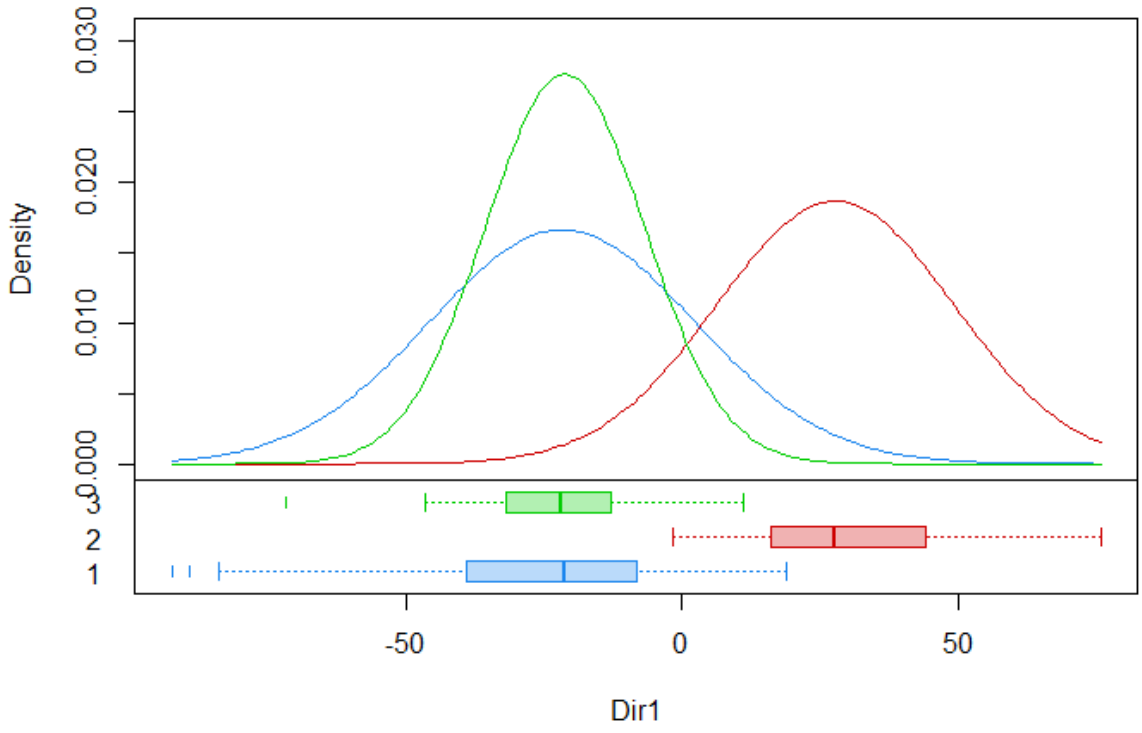

Figure A.8: LC cluster analysis: First dimension density plot for P

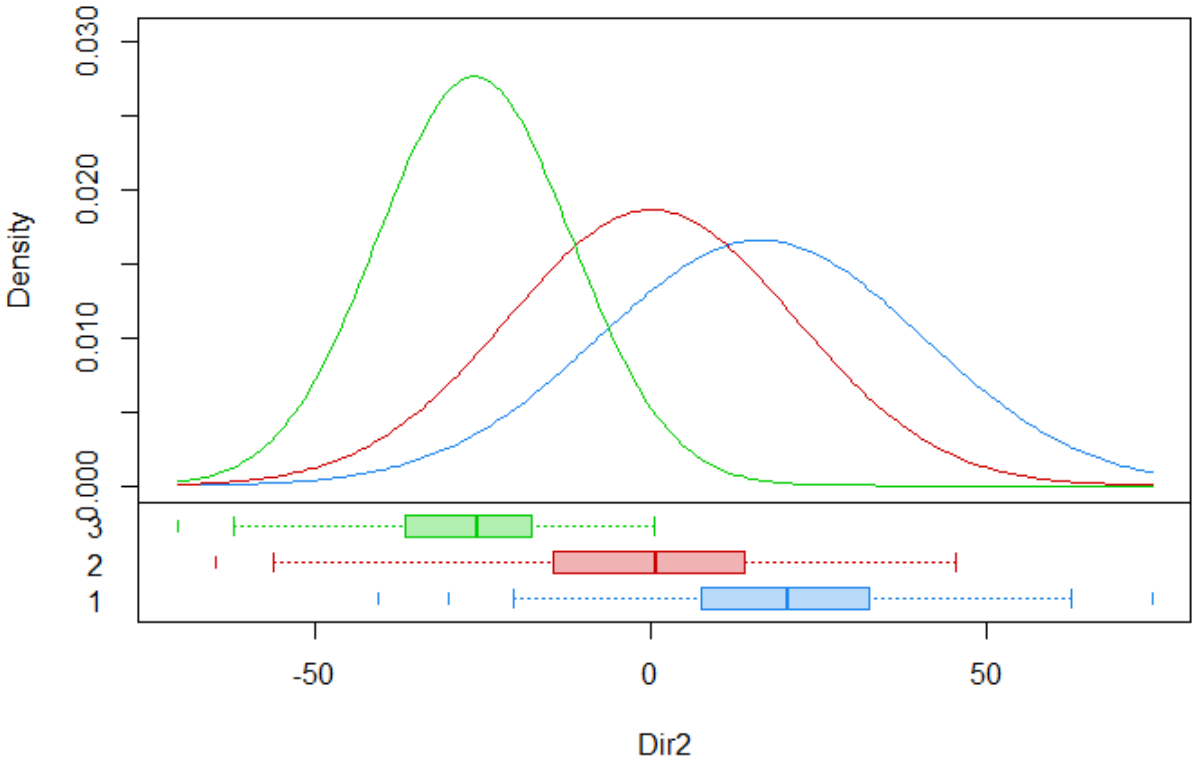

Figure A.9: LC cluster analysis: Second dimension plot for P

Appendix A.2: LC cluster analysis model summary for P: 3 components

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VII (spherical, varying volume) model with 3 components:

Clustering table:
  1   2   3
162 222 116

Mixing probabilities:
   1    2    3
0.35 0.44 0.22

Means:
  [,1] [,2] [,3]
a   79   55   78
b   45   36   59
c   75   91   71
d   70  110  103
e  105   89  128

Variances:
[,,1]
    a   b   c   d   e
a 578   0   0   0   0
b   0 578   0   0   0
c   0   0 578   0   0
d   0   0   0 578   0
e   0   0   0   0 578
[,,2]
    a   b   c   d   e
a 458   0   0   0   0
b   0 458   0   0   0
c   0   0 458   0   0
d   0   0   0 458   0
e   0   0   0   0 458
[,,3]
    a   b   c   d   e
a 208   0   0   0   0
b   0 208   0   0   0
c   0   0 208   0   0
d   0   0   0 208   0
e   0   0   0   0 208
```

Appendix A.3: BIC and ICL for P

```
Bayesian Information Criterion (BIC):
     EII    VII    EEI    VEI    EVI    VVI    EEE    EVE    VEE    VVE    EEV    VEV
1 -23185 -23185 -23184 -23184 -23184 -23184 -23113 -23113 -23113 -23113 -23113 -23113
2 -23085 -23096 -23079 -23090 -23093 -23102 -23136 -23139 -23059 -23067 -23154 -23102
3 -23055 -22989 -23078 -23014 -23113 -23054 -23124 -23128 -23072 -23082 -23196 -23144
4 -23079 -23026 -23103 -23049 -23161 -23108 -23131 -23157 -23111 -23137 -23243 -23228
5 -23113 -23064 -23121 -23083 -23193 -23159 -23177 -23207 -23148 -23178 -23307 -23306
6 -23129 -23095 -23135 -23119 -23234 -23202 -23187 -23247 -23181 -23229 -23349 -23384
7 -23161 -23122 -23156 -23142 -23278 -23255 -23206 -23231 -23209 -23265 -23400 -23459
8 -23161 -23158 -23195 -23178 -23312 -23309 -23228 -23297 -23240 -23338 -23489 -23530
9 -23178 -23182 -23197 -23202 -23349 -23361 -23231 -23344 -23280 -23387 -23553 -23597
     EVV    VVV
1 -23113 -23113
2 -23173 -23124
3 -23220 -23191
4 -23316 -23284
5 -23408 -23380
6 -23470 -23432
7 -23497 -23516
8 -23627 -23638
9 -23682 -23714

Top 3 models based on the BIC criterion:
 VII,3  VEI,3  VII,4
-22989 -23014 -23026

Integrated Complete-data Likelihood (ICL) criterion:
     EII    VII    EEI    VEI    EVI    VVI    EEE    EVE    VEE    VVE    EEV    VEV
1 -23185 -23185 -23184 -23184 -23184 -23184 -23113 -23113 -23113 -23113 -23113 -23113
2 -23247 -23180 -23237 -23171 -23247 -23181 -23311 -23306 -23145 -23150 -23332 -23183
3 -23277 -23177 -23291 -23216 -23314 -23259 -23316 -23264 -23316 -23260 -23372 -23311
4 -23366 -23324 -23413 -23351 -23428 -23361 -23329 -23316 -23454 -23337 -23425 -23420
5 -23432 -23393 -23422 -23369 -23528 -23428 -23420 -23375 -23497 -23374 -23552 -23520
6 -23509 -23519 -23469 -23550 -23559 -23558 -23425 -23459 -23581 -23469 -23598 -23597
7 -23530 -23528 -23474 -23520 -23555 -23576 -23465 -23446 -23571 -23473 -23552 -23658
8 -23472 -23619 -23548 -23571 -23568 -23692 -23500 -23492 -23672 -23626 -23675 -23735
9 -23478 -23593 -23495 -23605 -23607 -23726 -23484 -23546 -23703 -23659 -23722 -23780
     EVV    VVV
1 -23113 -23113
2 -23282 -23204
3 -23352 -23355
4 -23486 -23429
5 -23532 -23552
6 -23635 -23619
7 -23633 -23680
8 -23761 -23792
9 -23832 -23866

Top 3 models based on the ICL criterion:
 EEE,1  EEV,1  EVE,1
-23113 -23113 -23113
Best ICL values:
          EEE,1  EEV,1  EVE,1
ICL      -23113 -23113 -23113
ICL diff      0      0      0
```

Appendix B

Results for moderately separated data, M

Appendix B.1: Summary and PCA results for M

```
Summary: P

class          a                b                c                d
 1:145   Min.   : -1.5   Min.   :-29.3   Min.   : -2.6   Min.   : 13.4
 2: 64   1st Qu.: 57.6   1st Qu.: 54.2   1st Qu.: 38.9   1st Qu.: 83.3
 3:306   Median : 71.6   Median : 65.6   Median : 49.8   Median : 95.7
 4: 85   Mean   : 70.5   Mean   : 70.0   Mean   : 52.5   Mean   : 92.5
         3rd Qu.: 83.4   3rd Qu.: 79.7   3rd Qu.: 63.5   3rd Qu.:105.0
         Max.   :161.1   Max.   :161.9   Max.   :160.4   Max.   :142.9

Variance: P
> var(mix1$a)
[1] 580
> var(mix1$b)
[1] 791
> var(mix1$c)
[1] 432
> var(mix1$d)
[1] 380

PCA results:

Standard deviations (1, .., p=4):
[1] 1.181 1.076 0.928 0.767

Rotation (n x k) = (4 x 4):
      PC1    PC2    PC3   PC4
a   -0.555  0.211 -0.713 0.373
b    0.669 -0.246 -0.250 0.656
c   -0.494 -0.511  0.495 0.500
d    0.037  0.796  0.429 0.425

Importance of components:
                        PC1   PC2   PC3   PC4
Standard deviation     1.181 1.076 0.928 0.767
Proportion of Variance 0.349 0.289 0.215 0.147
Cumulative Proportion  0.349 0.638 0.853 1.000
```

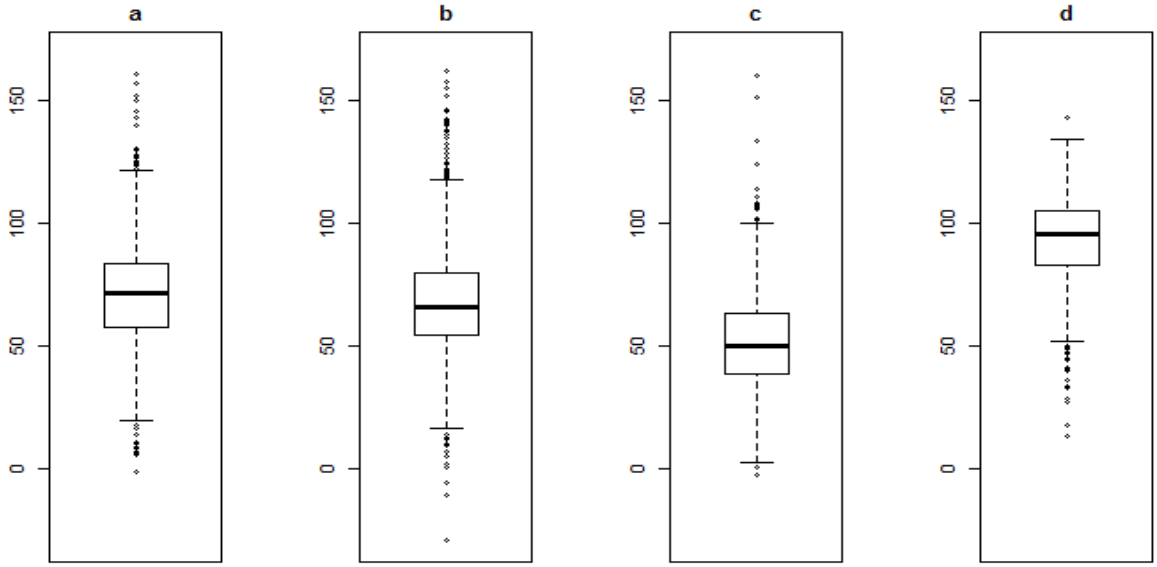

Figure B.1: Box and whisker plots for variables in data set M

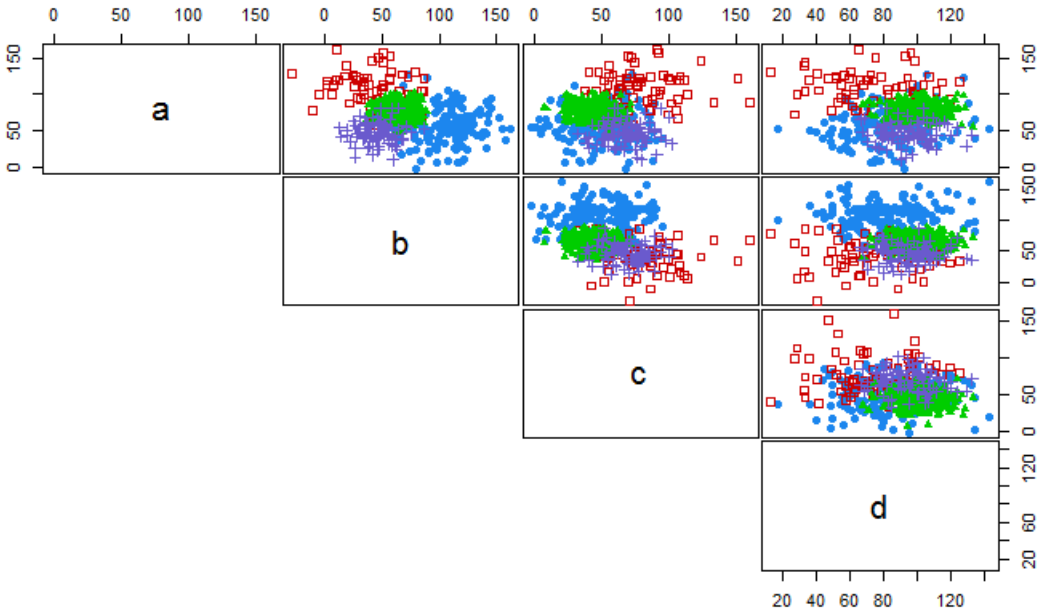

Figure B.2: Pairs plot for variables in M

Table B.1: PCA results for data set M

	PC1	PC2	PC3	PC4
Standard deviation	1.181	1.076	0.928	0.767
Percentage of Variance	34.9	28.9	21.5	14.7
Cumulative Percentage	34.9	63.8	85.3	100.0

Table B.2: Hierarchical clustering results for data set M

Agglomeration	Distance	Standardisation	Rand	ARI	MA	FM	Jaccard
Ward	Euclidean	No standardisation	0.896	0.773	0.773	0.853	0.743
		Z- score	0.891	0.762	0.763	0.847	0.734
		Range	0.891	0.765	0.766	0.850	0.739
		Maximum	0.892	0.765	0.765	0.848	0.736
	Manhattan	No standardisation	0.895	0.770	0.770	0.851	0.741
		Z- score	0.881	0.743	0.744	0.838	0.720
		Range	0.897	0.771	0.772	0.850	0.739
		Maximum	0.903	0.788	0.789	0.863	0.759
	Canberra	No standardisation	0.837	0.635	0.636	0.758	0.610
		Z- score	0.743	0.397	0.399	0.582	0.404
		Range	0.837	0.635	0.636	0.758	0.610
		Maximum	0.837	0.635	0.636	0.758	0.610
	Minkowski	No standardisation	0.896	0.773	0.773	0.853	0.743
		Z- score	0.891	0.762	0.763	0.847	0.734
		Range	0.891	0.765	0.766	0.850	0.739
		Maximum	0.892	0.765	0.765	0.848	0.736
Single	Euclidean	No standardisation	0.357	0.008	0.008	0.592	0.351
		Z- score	0.357	0.008	0.008	0.592	0.351
		Range	0.357	0.008	0.008	0.592	0.351
		Maximum	0.357	0.008	0.008	0.592	0.351
	Manhattan	No standardisation	0.359	0.010	0.010	0.592	0.352
		Z- score	0.365	0.015	0.015	0.593	0.353
		Range	0.365	0.015	0.015	0.593	0.353
		Maximum	0.365	0.015	0.015	0.593	0.353
	Canberra	No standardisation	0.358	0.009	0.009	0.591	0.351
		Z- score	0.349	-0.005	-0.005	0.585	0.345
		Range	0.358	0.009	0.009	0.591	0.351
		Maximum	0.358	0.009	0.009	0.591	0.351
	Minkowski	No standardisation	0.357	0.008	0.008	0.592	0.351
		Z- score	0.357	0.008	0.008	0.592	0.351
		Range	0.357	0.008	0.008	0.592	0.351
		Maximum	0.357	0.008	0.008	0.592	0.351
Complete	Euclidean	No standardisation	0.758	0.519	0.520	0.731	0.560
		Z- score	0.501	0.165	0.165	0.628	0.403
		Range	0.689	0.412	0.413	0.694	0.504
		Maximum	0.583	0.254	0.255	0.636	0.429
	Manhattan	No standardisation	0.800	0.560	0.561	0.713	0.554
		Z- score	0.780	0.544	0.545	0.728	0.566
		Range	0.734	0.461	0.463	0.688	0.514
		Maximum	0.737	0.443	0.445	0.656	0.485
	Canberra	No standardisation	0.637	0.210	0.213	0.494	0.327
		Z- score	0.580	0.077	0.080	0.400	0.250
		Range	0.637	0.210	0.213	0.494	0.327
		Maximum	0.637	0.210	0.213	0.494	0.327
	Minkowski	No standardisation	0.758	0.519	0.520	0.731	0.560
		Z- score	0.501	0.165	0.165	0.628	0.403
		Range	0.689	0.412	0.413	0.694	0.504
		Maximum	0.583	0.254	0.255	0.636	0.429
Average	Euclidean	No standardisation	0.414	0.067	0.067	0.604	0.369
		Z- score	0.518	0.192	0.192	0.642	0.416
		Range	0.410	0.061	0.061	0.602	0.367
		Maximum	0.712	0.449	0.450	0.711	0.525
	Manhattan	No standardisation	0.403	0.055	0.055	0.601	0.365
		Z- score	0.687	0.412	0.413	0.699	0.507
		Range	0.409	0.061	0.061	0.603	0.367
		Maximum	0.397	0.049	0.049	0.600	0.363
	Canberra	No standardisation	0.388	0.038	0.039	0.597	0.360
		Z- score	0.776	0.512	0.514	0.687	0.523
		Range	0.388	0.038	0.039	0.597	0.360
		Maximum	0.388	0.038	0.039	0.597	0.360
	Minkowski	No standardisation	0.414	0.067	0.067	0.604	0.369
		Z- score	0.518	0.192	0.192	0.642	0.416
		Range	0.410	0.061	0.061	0.602	0.367
		Maximum	0.712	0.449	0.450	0.711	0.525

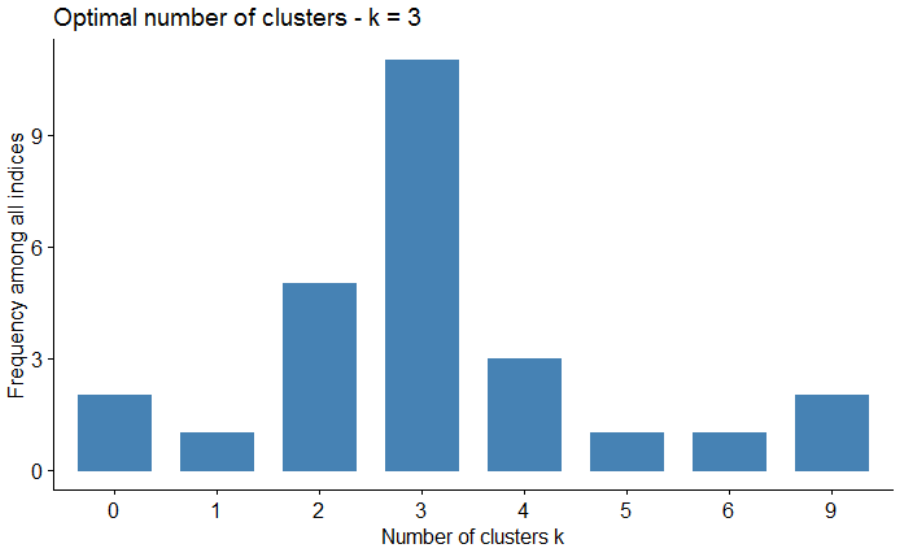

Figure B.3: Optimal number of clusters for M

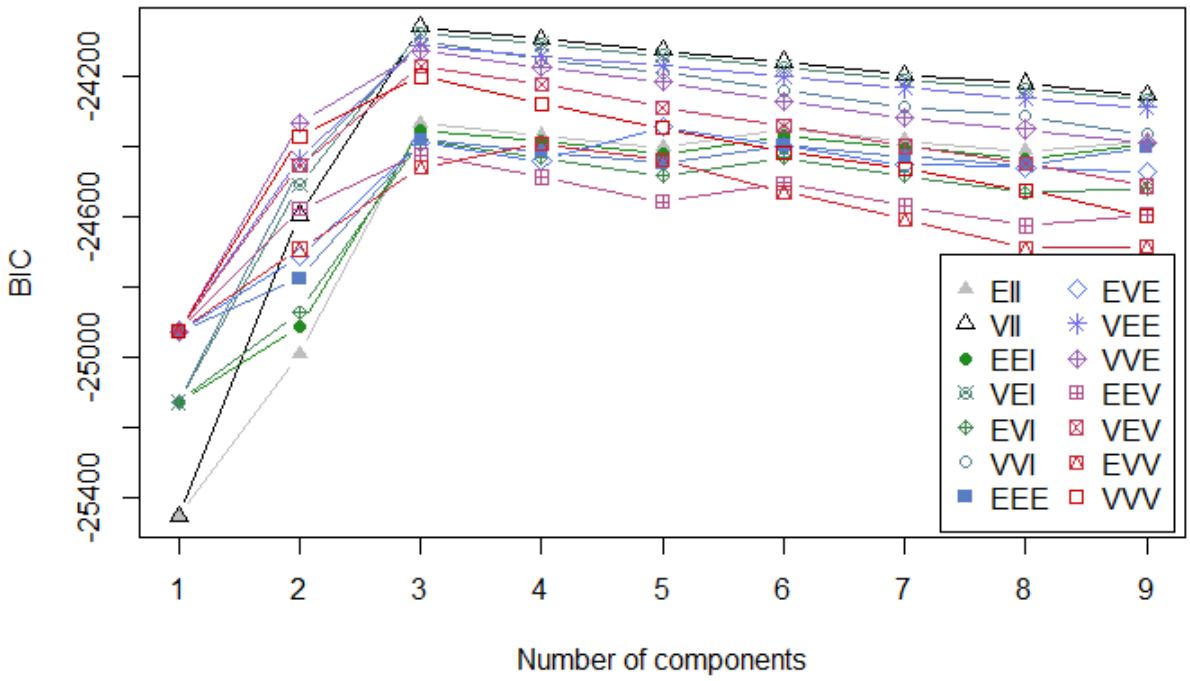

Figure B.4: BIC plot for M

Table B.3: VEI Bootstrap sequential LRTS for M

Clusters	LRTS	Bootstrap p-value
1 vs 2	644.72	0.01
2 vs 3	23.48	0.02
3 vs 4	85.52	0.01
4 vs 5	194.45	0.01
5 vs 6	7.92	0.38

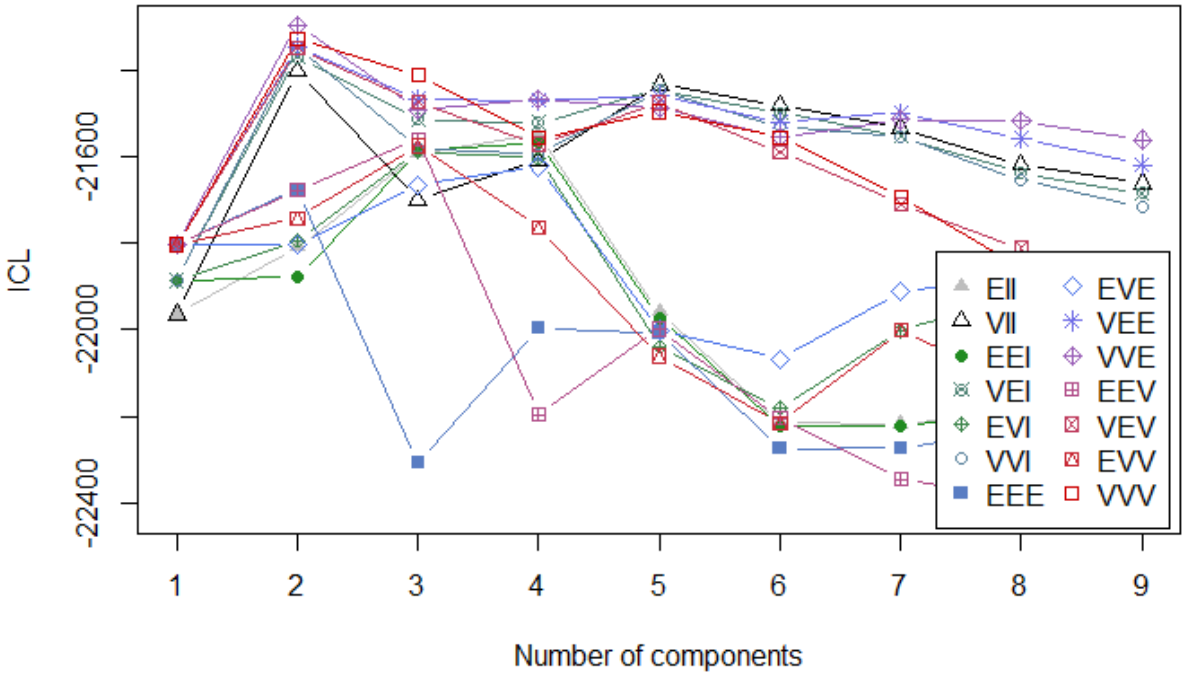

Figure B.5: ICL plot for M

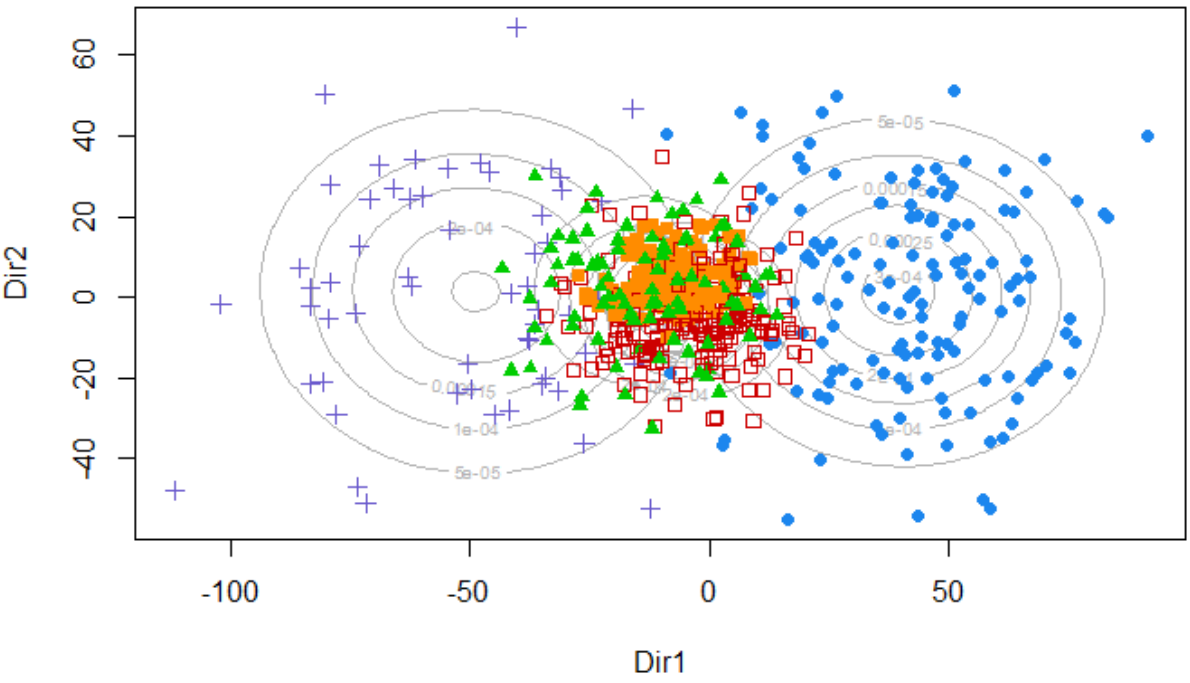

Figure B.6: LC cluster analysis: Contours Plot for 5 cluster solution for M

Table B.4: VII Bootstrap sequential LRTS for M

Clusters	LRTS	Bootstrap p-value
1 vs 2	686.81	0.01
2 vs 3	2.50	0.68

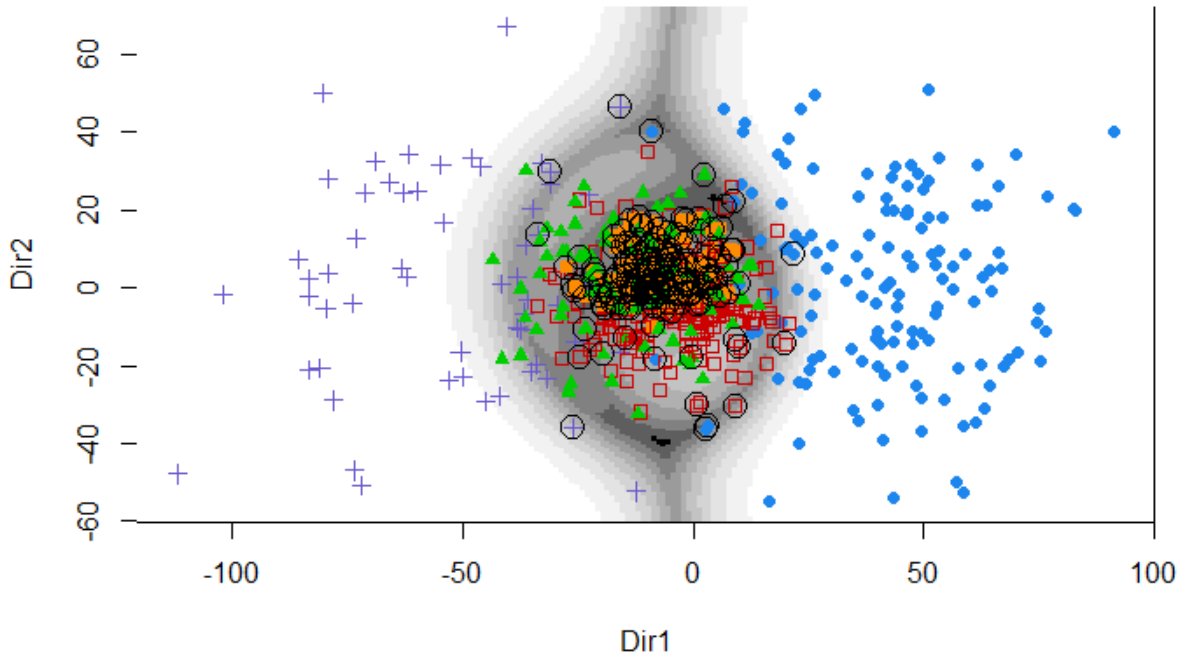

Figure B.7: LC cluster analysis: Uncertainty Plot for 5 cluster solution for M

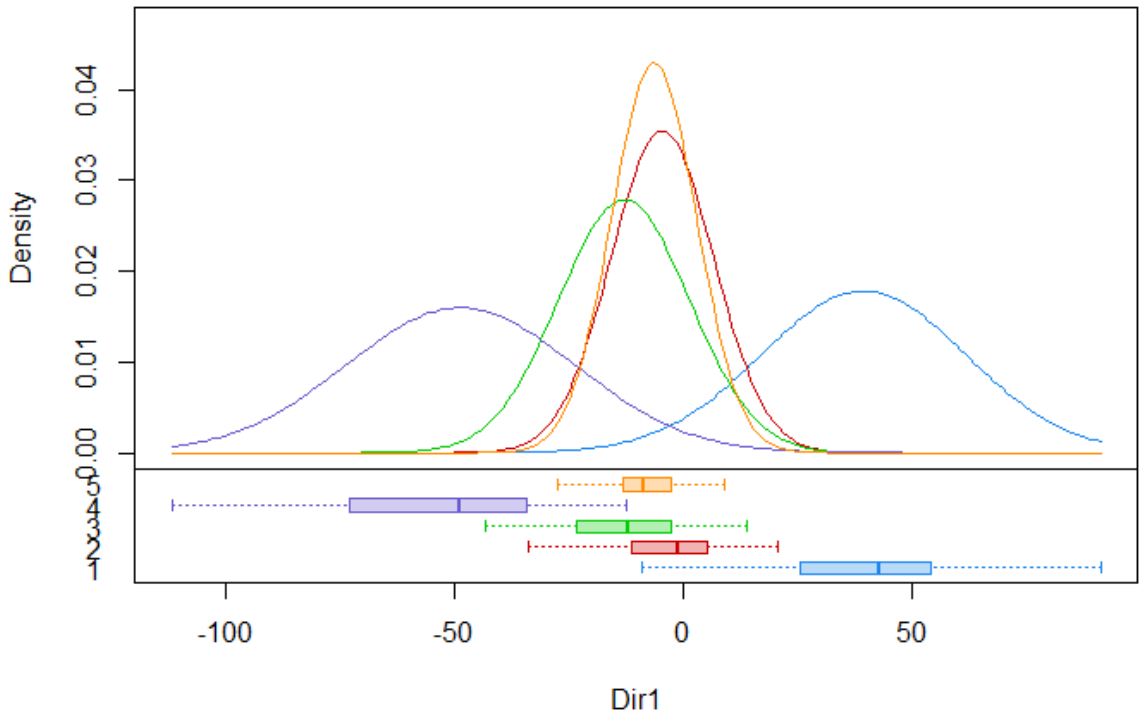

Figure B.8: LC cluster analysis: First dimension density plot for 5 cluster solution for M

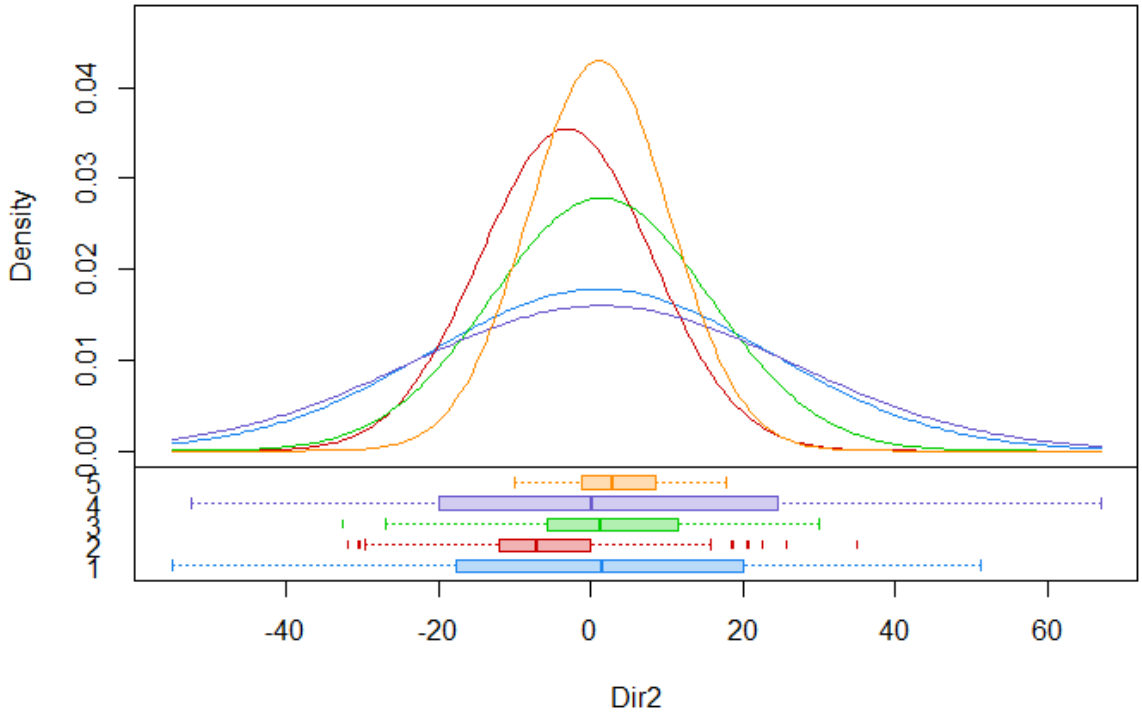

Figure B.9: LC cluster analysis: Second dimension plot for 5 cluster solution for M

Appendix B.2: BIC and ICL values for M

```
Bayesian Information Criterion (BIC):
     EII    VII    EEI    VEI    EVI    VVI    EEE    EVE    VEE    VVE    EEV    VEV
1 -21965 -21965 -21885 -21885 -21885 -21885 -21802 -21802 -21802 -21802 -21802 -21802
2 -21755 -21316 -21809 -21279 -21751 -21257 -21646 -21738 -21256 -21209 -21661 -21247
3 -21530 -21352 -21528 -21294 -21538 -21311 -21770 -21597 -21267 -21263 -21515 -21230
4 -21445 -21269 -21462 -21247 -21497 -21282 -21519 -21490 -21214 -21205 -21653 -21280
5 -21477 -21072 -21494 -21091 -21547 -21161 -21524 -21549 -21126 -21190 -21580 -21250
6 -21509 -21102 -21526 -21121 -21595 -21206 -21556 -21559 -21157 -21236 -21599 -21321
7 -21541 -21139 -21558 -21158 -21594 -21250 -21588 -21553 -21185 -21274 -21646 -21396
8 -21514 -21169 -21527 -21187 -21582 -21291 -21559 -21491 -21211 -21300 -21667 -21451
9 -21545 -21201 -21556 -21219 -21566 -21341 -21591 -21500 -21244 -21327 -21691 -21500
     EVV    VVV
1 -21802 -21802
2 -21693 -21243
3 -21539 -21230
4 -21618 -21312
5 -21648 -21318
6 -21682 -21384
7 -21642 -21461
8 -21713 -21550
9 -21753 -21612

Top 3 models based on the BIC criterion:
 VII,5  VEI,5  VII,6
-21072 -21091 -21102

Integrated Complete-data Likelihood (ICL) criterion:
     EII    VII    EEI    VEI    EVI    VVI    EEE    EVE    VEE    VVE    EEV    VEV
1 -21965 -21965 -21885 -21885 -21885 -21885 -21802 -21802 -21802 -21802 -21802 -21802
2 -21809 -21401 -21876 -21368 -21793 -21344 -21674 -21804 -21344 -21293 -21676 -21348
3 -21591 -21699 -21585 -21513 -21588 -21580 -22308 -21664 -21464 -21490 -21557 -21472
4 -21548 -21608 -21567 -21520 -21601 -21593 -21996 -21621 -21468 -21466 -22196 -21569
5 -21959 -21431 -21972 -21444 -22039 -21445 -22006 -22000 -21459 -21485 -21998 -21473
6 -22216 -21478 -22225 -21496 -22181 -21529 -22275 -22068 -21519 -21552 -22204 -21587
7 -22218 -21531 -22225 -21552 -22002 -21551 -22275 -21911 -21497 -21513 -22347 -21708
8 -22191 -21617 -22190 -21637 -21926 -21652 -22228 -21878 -21557 -21516 -22390 -21811
9 -22261 -21660 -22254 -21684 -21866 -21713 -22295 -21886 -21617 -21559 -22426 -21858
     EVV    VVV
1 -21802 -21802
2 -21741 -21325
3 -21573 -21409
4 -21764 -21556
5 -22061 -21494
6 -22216 -21553
7 -22001 -21691
8 -22127 -21855
9 -22113 -21893

Top 3 models based on the ICL criterion:
 VVE,2  VVV,2  VEE,2
-21293 -21325 -21344
Best ICL values:
          VVE,2    VVV,2    VEE,2
ICL      -21293 -21324.7 -21343.5
ICL diff       0    -31.5    -50.3
```

Appendix B.3: LC cluster analysis model summary for M: 4 components

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVE (ellipsoidal, equal orientation) model with 4 components:

Clustering table:
  1   2   3   4
215 155  86 144

Mixing probabilities:
    1     2     3     4
0.385 0.252 0.136 0.227

Means:
  [,1] [,2] [,3]  [,4]
a 71.8 75.9 46.0  77.0
b 85.2 64.8 45.7  64.6
c 58.8 41.3 66.9  45.6
d 81.0 95.7 96.0 106.5

Variances:
[,,1]
       a      b      c     d
a 1048.4 -674.3  284.1 -36.2
b -674.3 1376.3 -361.2  93.3
c  284.1 -361.2  705.1 -19.2
d  -36.2   93.3  -19.2 557.4
[,,2]
       a      b      c      d
a 110.21 -12.26 -14.68  -2.36
b -12.26 105.82  -7.67  -7.71
c -14.68  -7.67 122.84 -20.85
d  -2.36  -7.71 -20.85  82.08
[,,3]
       a      b      c      d
a 202.47 -24.27 -13.81   8.34
b -24.27 201.41  -9.77   3.52
c -13.81  -9.77 224.22 -15.85
d   8.34   3.52 -15.85 180.90
[,,4]
       a       b      c      d
a 90.085   0.812  -6.59  -7.98
b  0.812  86.636  -2.83 -10.47
c -6.587  -2.829  89.22 -12.62
d -7.982 -10.468 -12.62  72.23
```

Appendix B.4: LC cluster analysis model summary for M: 5 components

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VII (spherical, varying volume) model with 5 components:

Clustering table:
  1   2   3   4   5
142 175  94  58 131

Mixing probabilities:
    1     2     3     4     5
0.245 0.281 0.148 0.104 0.221

Means:
   [,1] [,2] [,3]  [,4]  [,5]
a  55.4 77.0 47.3 109.1  76.3
b 107.3 65.3 46.5  41.4  63.9
c  50.2 42.3 67.7  80.4  44.8
d  82.0 98.6 94.9  74.6 103.3

Variances:
[,,1]
    a   b   c   d
a 501   0   0   0
b   0 501   0   0
c   0   0 501   0
d   0   0   0 501
[,,2]
    a   b   c   d
a 126   0   0   0
b   0 126   0   0
c   0   0 126   0
d   0   0   0 126
[,,3]
    a   b   c   d
a 205   0   0   0
b   0 205   0   0
c   0   0 205   0
d   0   0   0 205
[,,4]
    a   b   c   d
a 624   0   0   0
b   0 624   0   0
c   0   0 624   0
d   0   0   0 624
[,,5]
     a    b    c    d
a 86.3  0.0  0.0  0.0
b  0.0 86.3  0.0  0.0
c  0.0  0.0 86.3  0.0
d  0.0  0.0  0.0 86.3
```

Appendix C

Results for well separated data, W

Appendix C.1: Summary and PCA results for W

```
Summary: W

class          a                 b                 c                 d
 1:137   Min.   : 40.44   Min.   :-15.35   Min.   : 18.97   Min.   : -0.26
 2:184   1st Qu.: 96.66   1st Qu.: 79.56   1st Qu.: 54.09   1st Qu.: 60.67
 3:379   Median :106.43   Median :107.16   Median : 63.78   Median : 73.90
         Mean   :107.07   Mean   : 99.23   Mean   : 65.61   Mean   : 76.46
         3rd Qu.:118.48   3rd Qu.:121.69   3rd Qu.: 76.15   3rd Qu.: 90.37
         Max.   :159.32   Max.   :160.78   Max.   :122.76   Max.   :148.60

Variance: W

> var(mix1$a)
[1] 297.99
> var(mix1$b)
[1] 848.21
> var(mix1$c)
[1] 258.89
> var(mix1$d)
[1] 651.71

PCA results: W

Standard deviations (1, .., p=4):
[1] 1.227  1.131  0.803  0.755

Rotation (n x k) = (4 x 4):
     PC1    PC2    PC3    PC4
a   -0.634  0.145  0.728  0.219
b   -0.034  0.769 -0.344  0.537
c   -0.430 -0.586 -0.421  0.543
d   -0.642  0.209 -0.418 -0.607

Importance of components:
                          PC1   PC2   PC3   PC4
Standard deviation     1.227 1.131 0.803 0.755
Proportion of Variance 0.376 0.320 0.161 0.143
Cumulative Proportion  0.376 0.696 0.857 1.000
```

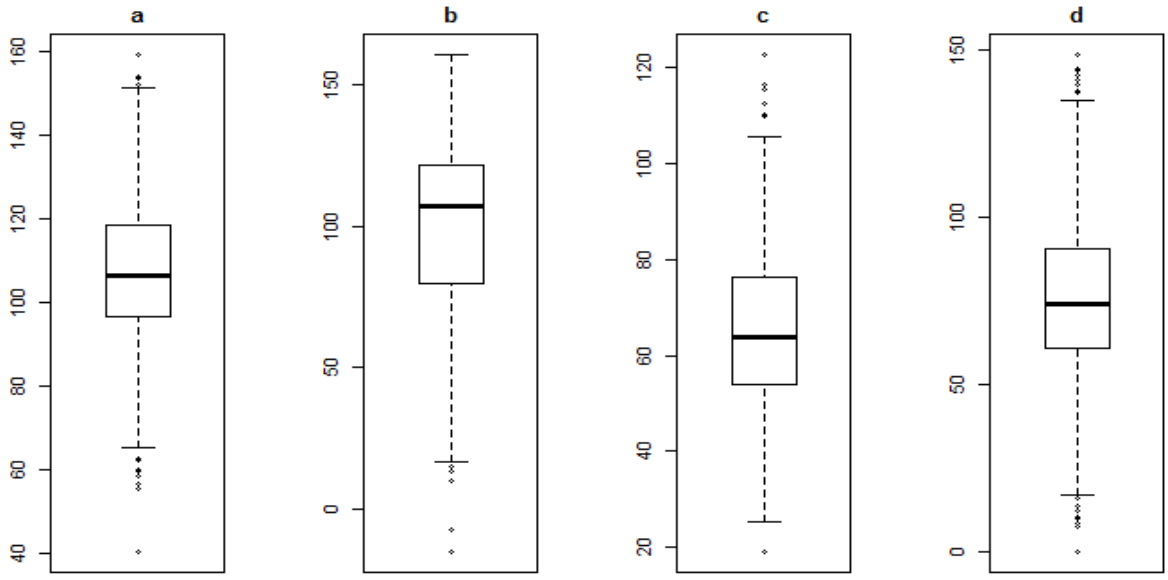

Figure C.1: Box and whisker plots for variables in data set W

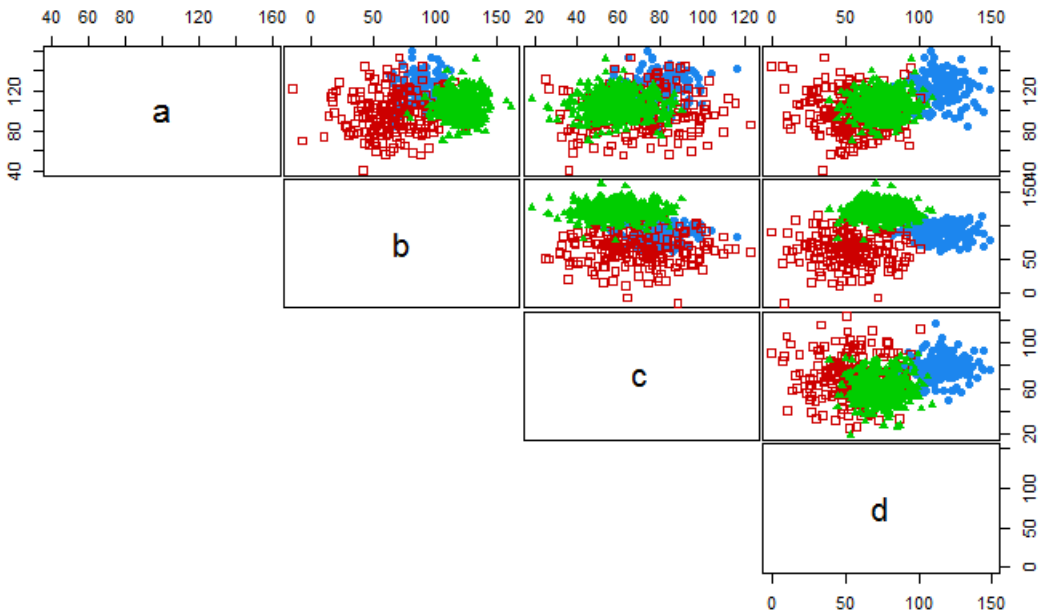

Figure C.2: Pairs plot for W

Table C.1: PCA results for data set W

	PC1	PC2	PC3	PC4
Standard deviation	1.227	1.131	0.803	0.755
Percentage of Variance	37.6	32.0	16.1	14.3
Cumulative Percentage	37.6	69.6	85.8	100.0

Table C.2: Hierarchical clustering results for data set W

Agglomeration	Distance	Standardisation	Rand	ARI	MA	FM	Jaccard
Ward	Euclidean	No standardisation	0.930	0.855	0.855	0.913	0.840
		Z- score	0.921	0.837	0.838	0.905	0.825
		Range	0.934	0.863	0.863	0.919	0.850
		Maximum	0.934	0.863	0.863	0.919	0.850
	Manhattan	No standardisation	0.919	0.831	0.832	0.899	0.816
		Z- score	0.944	0.885	0.885	0.931	0.871
		Range	0.936	0.868	0.868	0.922	0.855
		Maximum	0.948	0.892	0.892	0.935	0.879
	Canberra	No standardisation	0.935	0.866	0.866	0.920	0.852
		Z- score	0.907	0.805	0.806	0.883	0.791
		Range	0.935	0.866	0.866	0.920	0.852
		Maximum	0.935	0.866	0.866	0.920	0.852
	Minkowski	No standardisation	0.930	0.855	0.855	0.913	0.840
		Z- score	0.921	0.837	0.838	0.905	0.825
		Range	0.934	0.863	0.863	0.919	0.850
		Maximum	0.934	0.863	0.863	0.919	0.850
Single	Euclidean	No standardisation	0.402	0.003	0.003	0.632	0.400
		Z- score	0.402	0.003	0.003	0.632	0.400
		Range	0.402	0.003	0.003	0.632	0.400
		Maximum	0.402	0.003	0.003	0.632	0.400
	Manhattan	No standardisation	0.402	0.003	0.003	0.632	0.400
		Z- score	0.402	0.003	0.003	0.632	0.400
		Range	0.402	0.003	0.003	0.632	0.400
		Maximum	0.402	0.003	0.003	0.632	0.400
	Canberra	No standardisation	0.402	0.003	0.003	0.632	0.400
		Z- score	0.399	-0.003	-0.003	0.629	0.398
		Range	0.402	0.003	0.003	0.632	0.400
		Maximum	0.402	0.003	0.003	0.632	0.400
	Minkowski	No standardisation	0.402	0.003	0.003	0.632	0.400
		Z- score	0.402	0.003	0.003	0.632	0.400
		Range	0.402	0.003	0.003	0.632	0.400
		Maximum	0.402	0.003	0.003	0.632	0.400
Complete	Euclidean	No standardisation	0.633	0.258	0.260	0.581	0.408
		Z- score	0.801	0.594	0.595	0.768	0.622
		Range	0.792	0.583	0.584	0.771	0.622
		Maximum	0.714	0.450	0.451	0.729	0.555
	Manhattan	No standardisation	0.674	0.390	0.390	0.718	0.532
		Z- score	0.648	0.300	0.301	0.619	0.443
		Range	0.719	0.456	0.457	0.727	0.555
		Maximum	0.707	0.435	0.435	0.719	0.544
	Canberra	No standardisation	0.415	0.015	0.015	0.630	0.401
		Z- score	0.662	0.363	0.363	0.698	0.513
		Range	0.415	0.015	0.015	0.630	0.401
		Maximum	0.415	0.015	0.015	0.630	0.401
	Minkowski	No standardisation	0.633	0.258	0.260	0.581	0.408
		Z- score	0.801	0.594	0.595	0.768	0.622
		Range	0.792	0.583	0.584	0.771	0.622
		Maximum	0.714	0.450	0.451	0.729	0.555
Average	Euclidean	No standardisation	0.404	0.004	0.004	0.631	0.400
		Z- score	0.402	0.003	0.003	0.632	0.400
		Range	0.402	0.003	0.003	0.632	0.400
		Maximum	0.408	0.008	0.008	0.631	0.400
	Manhattan	No standardisation	0.407	0.007	0.007	0.631	0.400
		Z- score	0.408	0.008	0.008	0.631	0.400
		Range	0.409	0.010	0.010	0.630	0.400
		Maximum	0.408	0.008	0.008	0.631	0.400
	Canberra	No standardisation	0.402	0.003	0.003	0.632	0.400
		Z- score	0.843	0.686	0.687	0.832	0.704
		Range	0.402	0.003	0.003	0.632	0.400
		Maximum	0.402	0.003	0.003	0.632	0.400
	Minkowski	No standardisation	0.404	0.004	0.004	0.631	0.400
		Z- score	0.402	0.003	0.003	0.632	0.400
		Range	0.402	0.003	0.003	0.632	0.400
		Maximum	0.408	0.008	0.008	0.631	0.400

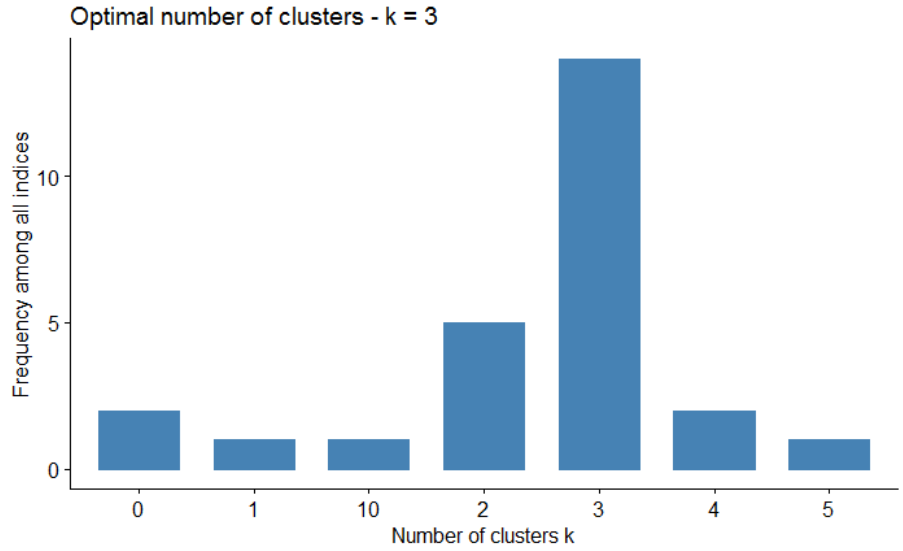

Figure C.3: Optimal number of clusters for W

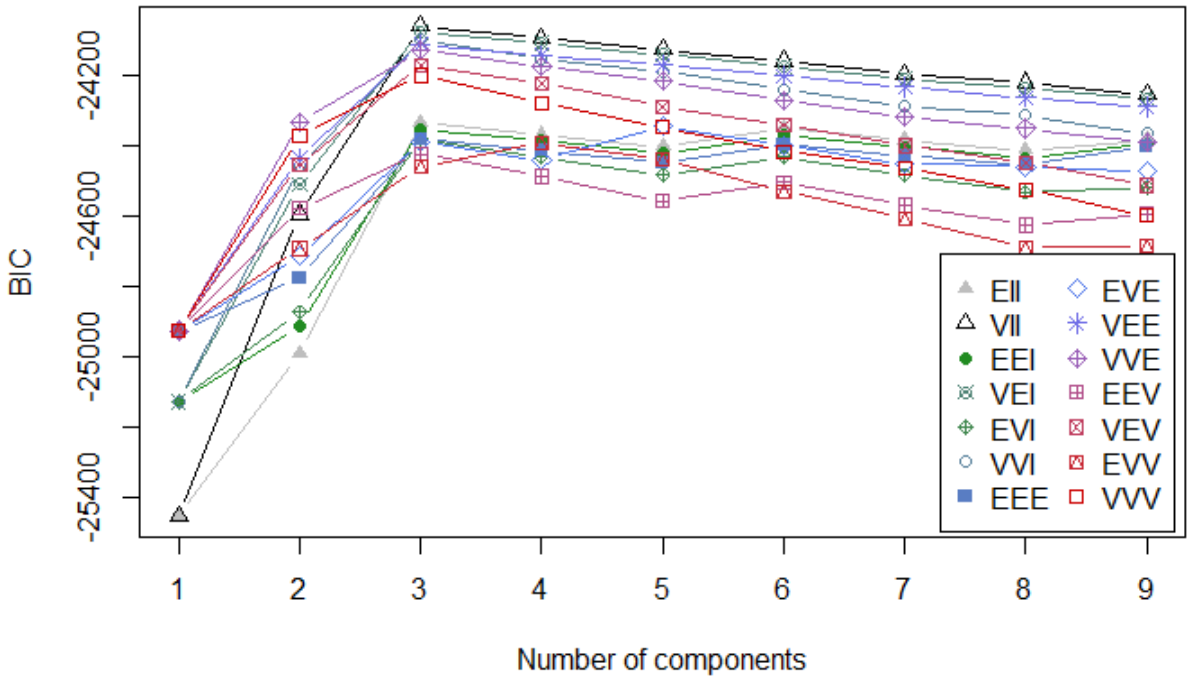

Figure C.4: BIC plot for W

Table C.3: Bootstrap sequential LRTS for W

Clusters	LRTS	Bootstrap p-value
1 vs 2	899	0.01
2 vs 3	547	0.01
3 vs 4	5	0.40

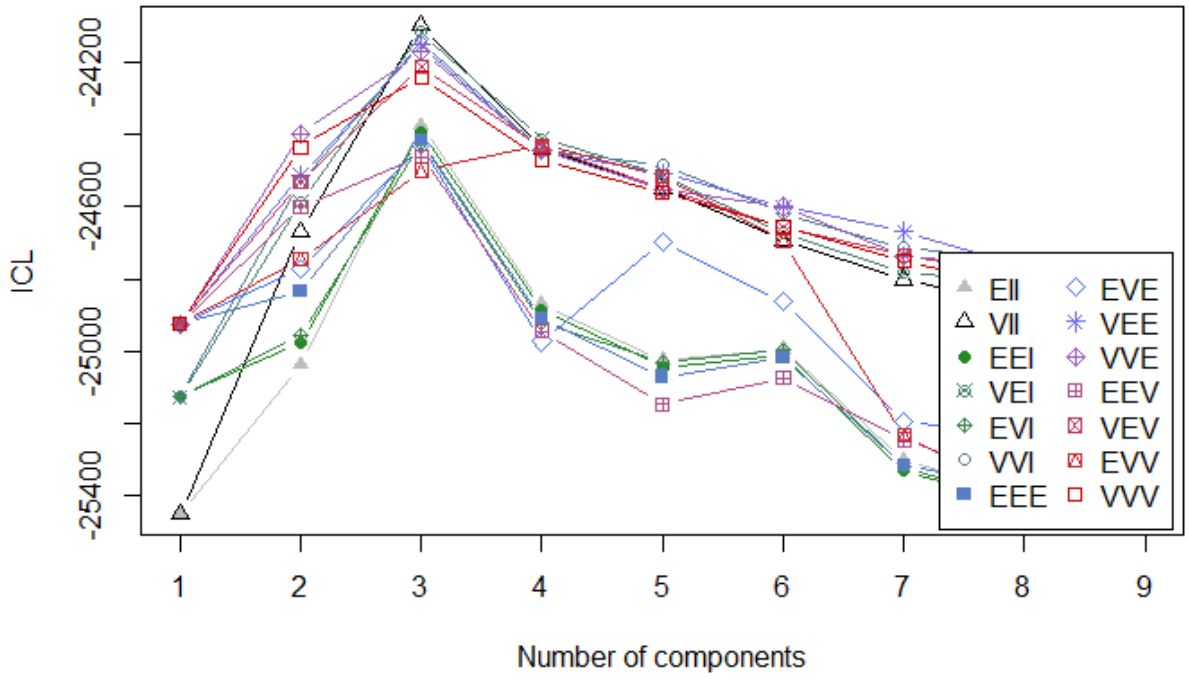

Figure C.5: ICL plot for W

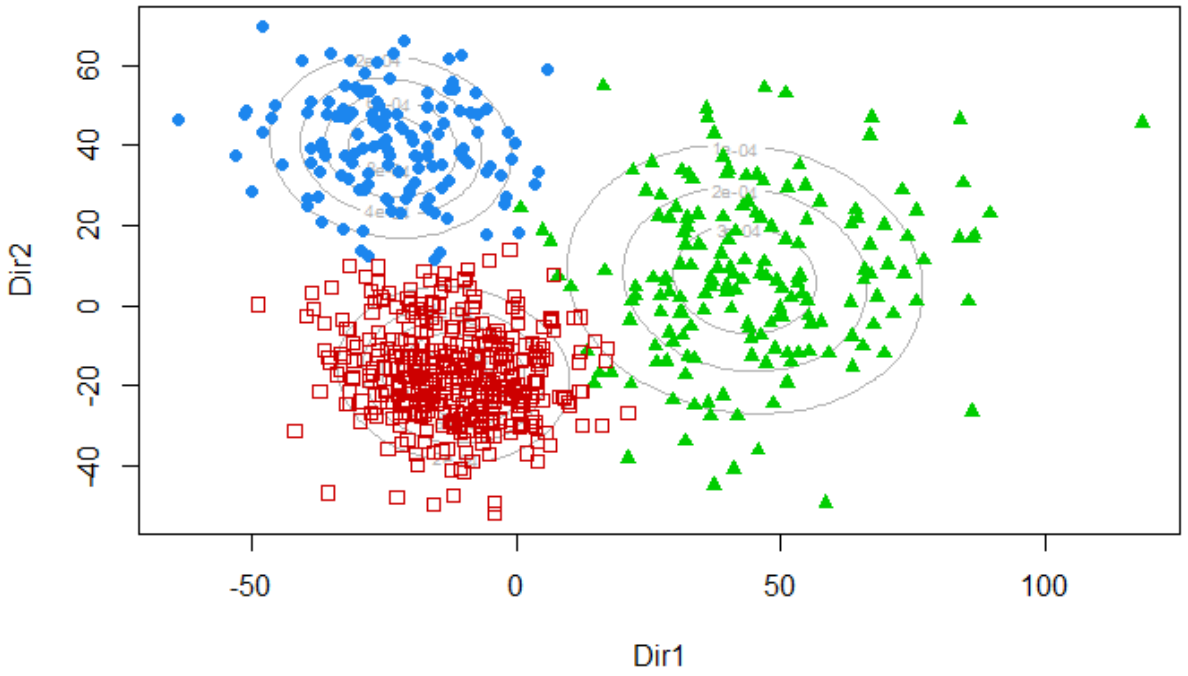

Figure C.6: LC cluster analysis: Contours Plot for 3 cluster solution for W

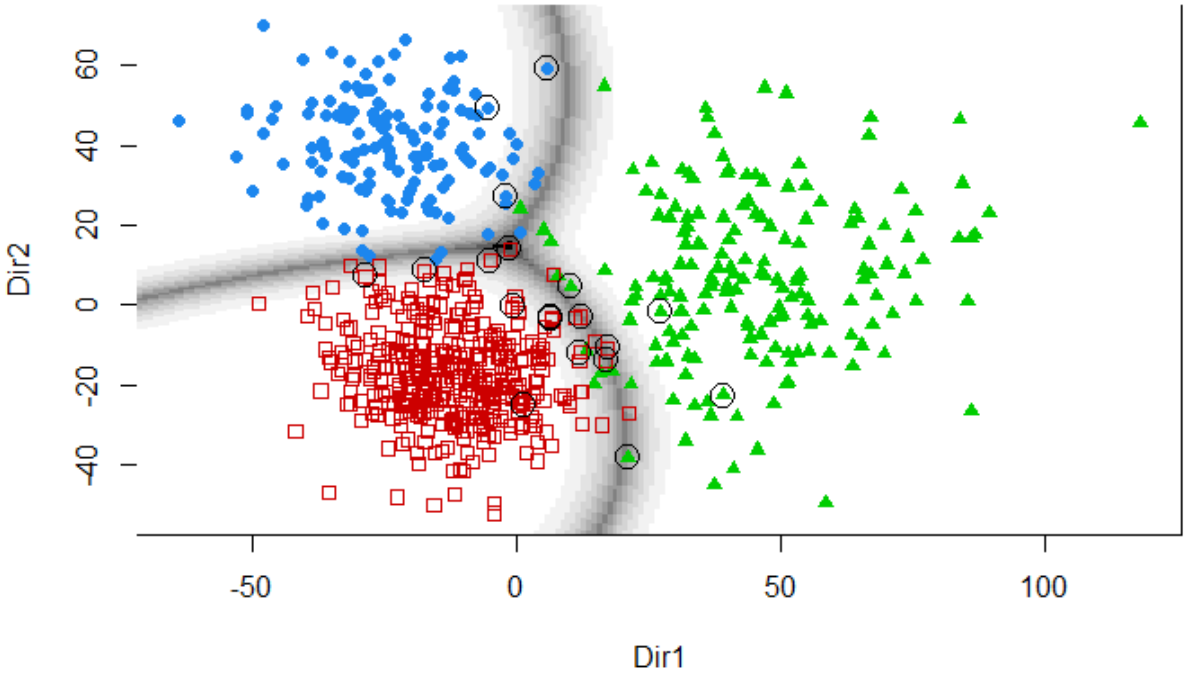

Figure C.7: LC cluster analysis: Uncertainty Plot for 3 cluster solution for W

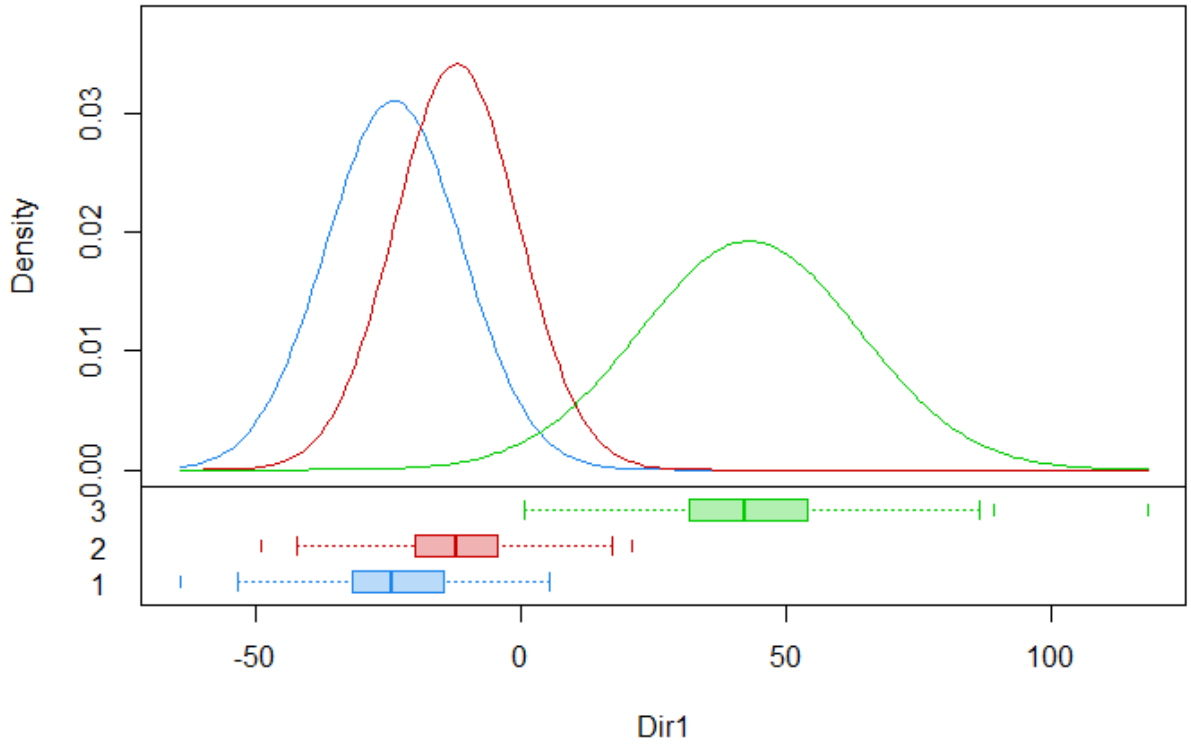

Figure C.8: LC cluster analysis: First dimension density plot for 3 cluster solution for W

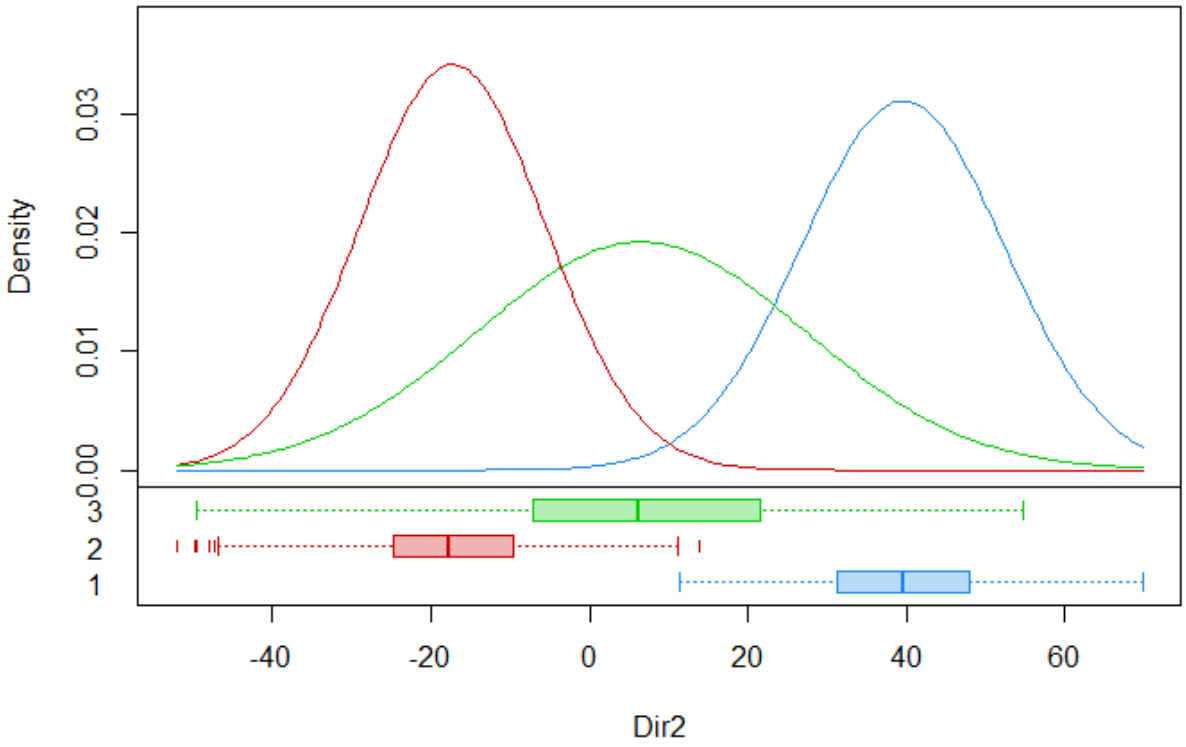

Figure C.9: LC cluster analysis: Second dimension plot for 3 cluster solution for W

Appendix C.2: LC cluster analysis model summary for W: 3 components

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VII (spherical, varying volume) model with 3 components:

Clustering table:
  1   2   3
137 387 176

Mixing probabilities:
   1    2    3
0.20 0.55 0.26

Means:
  [,1] [,2] [,3]
a  124  105   98
b   88  120   63
c   79   59   69
d  115   74   53

Variances:
[,,1]
    a   b   c   d
a 165   0   0   0
b   0 165   0   0
c   0   0 165   0
d   0   0   0 165
[,,2]
    a   b   c   d
a 136   0   0   0
b   0 136   0   0
c   0   0 136   0
d   0   0   0 136
[,,3]
    a   b   c   d
a 431   0   0   0
b   0 431   0   0
c   0   0 431   0
d   0   0   0 431
```

Appendix C.3: BIC and ICL for W

```
Bayesian Information Criterion (BIC):
     EII    VII    EEI    VEI    EVI    VVI    EEE    EVE    VEE    VVE    EEV    VEV
1 -25454 -25454 -25128 -25128 -25128 -25128 -24925 -24925 -24925 -24925 -24925 -24925
2 -24992 -24594 -24914 -24508 -24873 -24458 -24775 -24714 -24436 -24335 -24578 -24454
3 -24336 -24060 -24353 -24077 -24380 -24099 -24381 -24388 -24112 -24128 -24423 -24170
4 -24369 -24094 -24386 -24108 -24432 -24152 -24414 -24441 -24143 -24174 -24487 -24221
5 -24402 -24126 -24420 -24141 -24483 -24190 -24447 -24345 -24170 -24217 -24558 -24288
6 -24351 -24159 -24370 -24174 -24431 -24239 -24396 -24399 -24202 -24270 -24504 -24340
7 -24383 -24196 -24403 -24210 -24484 -24288 -24428 -24452 -24233 -24317 -24569 -24400
8 -24413 -24221 -24435 -24237 -24531 -24314 -24456 -24459 -24264 -24351 -24625 -24450
9 -24384 -24254 -24396 -24265 -24519 -24364 -24405 -24472 -24290 -24392 -24594 -24512
     EVV    VVV
1 -24925 -24925
2 -24692 -24372
3 -24459 -24200
4 -24392 -24279
5 -24437 -24348
6 -24529 -24416
7 -24607 -24466
8 -24689 -24525
9 -24685 -24597

Top 3 models based on the BIC criterion:
 VII,3  VEI,3  VII,4
-24060 -24077 -24094

Integrated Complete-data Likelihood (ICL) criterion:
     EII    VII    EEI    VEI    EVI    VVI    EEE    EVE    VEE    VVE    EEV    VEV
1 -25454 -25454 -25128 -25128 -25128 -25128 -24925 -24925 -24925 -24925 -24925 -24925
2 -25039 -24671 -24977 -24588 -24955 -24532 -24834 -24773 -24513 -24397 -24600 -24529
3 -24375 -24098 -24394 -24116 -24421 -24143 -24416 -24427 -24151 -24170 -24461 -24212
4 -24870 -24437 -24888 -24411 -24923 -24441 -24912 -24974 -24444 -24442 -24946 -24434
5 -25026 -24548 -25044 -24514 -25030 -24485 -25072 -24696 -24506 -24552 -25148 -24550
6 -24997 -24694 -25014 -24673 -24998 -24618 -25019 -24862 -24599 -24600 -25076 -24656
7 -25304 -24802 -25333 -24783 -25319 -24714 -25318 -25196 -24670 -24737 -25246 -24733
8 -25397 -24880 -25424 -24808 -25425 -24753 -25377 -25238 -24767 -24761 -25369 -24791
9 -25356 -24941 -25297 -24831 -25386 -24834 -25306 -25184 -24789 -24817 -25282 -24906
     EVV    VVV
1 -24925 -24925
2 -24746 -24434
3 -24498 -24240
4 -24429 -24470
5 -24514 -24561
6 -24696 -24656
7 -25235 -24751
8 -25415 -24817
9 -24992 -24900

Top 3 models based on the ICL criterion:
 VII,3  VEI,3  VVI,3
-24098 -24116 -24143

Best ICL values:
             VII,3        VEI,3        VVI,3
ICL          -24097       -24116       -24143
ICL diff          0          -18          -46
```

Appendix D

Results for seeds data

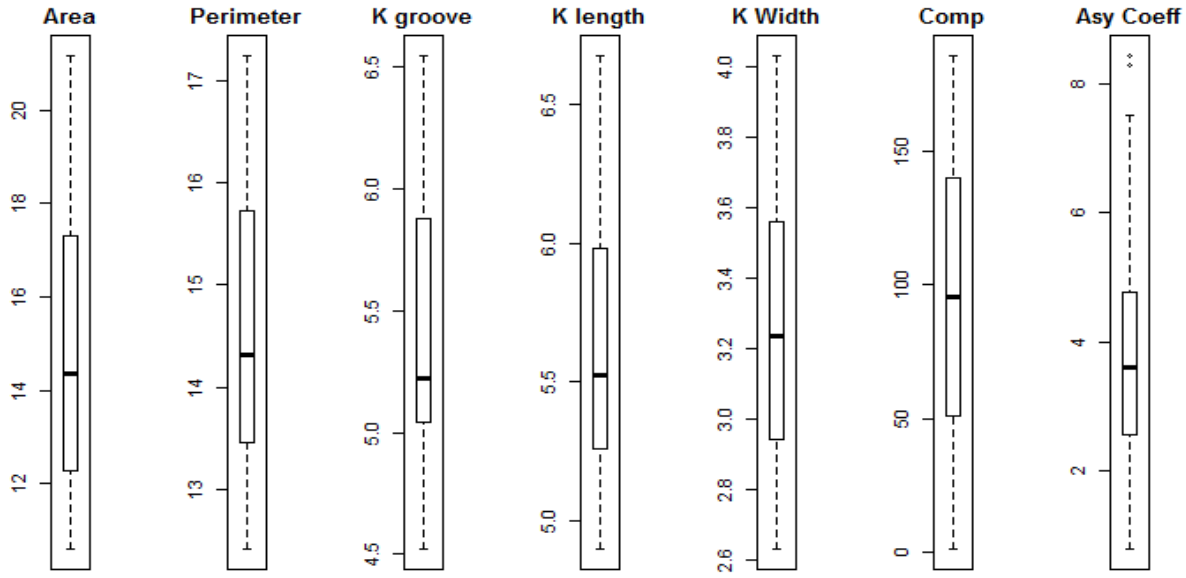

Figure D.1: Box and whisker plots seeds data

Table D.1: PCA results for seeds data

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	2.2407	1.0828	0.8393	0.2686	0.1516	0.0776	0.0311
Proportion of Variance	0.7173	0.1675	0.1006	0.0103	0.0033	0.0009	0.0001
Cumulative Proportion	0.7173	0.8848	0.9854	0.9957	0.9990	0.9999	1.0000

Table D.2: Hierarchical clustering results for seeds data

Agglomeration	Distance	Standardisation	Rand	ARI	MA	FM	Jaccard
Ward	Euclidean	No standardisation	0.645	0.205	0.212	0.473	0.309
		Z- score	0.877	0.724	0.726	0.816	0.689
		Range	0.858	0.680	0.683	0.787	0.648
		Maximum	0.765	0.479	0.484	0.658	0.490
	Manhattan	No standardisation	0.627	0.176	0.183	0.461	0.299
		Z- score	0.867	0.702	0.705	0.803	0.670
		Range	0.867	0.702	0.705	0.802	0.670
		Maximum	0.852	0.667	0.670	0.777	0.636
	Canberra	No standardisation	0.843	0.645	0.648	0.762	0.616
		Z- score	0.804	0.574	0.578	0.727	0.569
		Range	0.843	0.645	0.648	0.762	0.616
		Maximum	0.843	0.645	0.648	0.762	0.616
	Minkowski	No standardisation	0.645	0.205	0.212	0.473	0.309
		Z- score	0.877	0.724	0.726	0.816	0.689
		Range	0.858	0.680	0.683	0.787	0.648
		Maximum	0.765	0.479	0.484	0.658	0.490
Single	Euclidean	No standardisation	0.337	0.000	0.000	0.569	0.328
		Z- score	0.337	0.000	0.000	0.569	0.328
		Range	0.343	0.000	0.000	0.564	0.326
		Maximum	0.337	0.000	0.000	0.569	0.328
	Manhattan	No standardisation	0.337	0.000	0.000	0.569	0.328
		Z- score	0.343	0.000	0.000	0.564	0.326
		Range	0.337	0.000	0.000	0.569	0.328
		Maximum	0.337	0.000	0.000	0.569	0.328
	Canberra	No standardisation	0.337	0.000	0.000	0.569	0.328
		Z- score	0.337	0.000	0.000	0.569	0.328
		Range	0.337	0.000	0.000	0.569	0.328
		Maximum	0.337	0.000	0.000	0.569	0.328
	Minkowski	No standardisation	0.337	0.000	0.000	0.569	0.328
		Z- score	0.337	0.000	0.000	0.569	0.328
		Range	0.343	0.000	0.000	0.564	0.326
		Maximum	0.337	0.000	0.000	0.569	0.328
Complete	Euclidean	No standardisation	0.602	0.126	0.134	0.431	0.274
		Z- score	0.788	0.531	0.536	0.694	0.531
		Range	0.767	0.480	0.484	0.656	0.488
		Maximum	0.700	0.352	0.357	0.586	0.412
	Manhattan	No standardisation	0.639	0.194	0.201	0.466	0.304
		Z- score	0.816	0.588	0.591	0.727	0.571
		Range	0.768	0.483	0.487	0.659	0.491
		Maximum	0.841	0.642	0.645	0.760	0.613
	Canberra	No standardisation	0.754	0.467	0.471	0.659	0.489
		Z- score	0.766	0.504	0.508	0.692	0.523
		Range	0.754	0.467	0.471	0.659	0.489
		Maximum	0.754	0.467	0.471	0.659	0.489
	Minkowski	No standardisation	0.602	0.126	0.134	0.431	0.274
		Z- score	0.788	0.531	0.536	0.694	0.531
		Range	0.767	0.480	0.484	0.656	0.488
		Maximum	0.700	0.352	0.357	0.586	0.412
Average	Euclidean	No standardisation	0.624	0.173	0.180	0.461	0.299
		Z- score	0.742	0.489	0.493	0.713	0.533
		Range	0.744	0.501	0.504	0.727	0.545
		Maximum	0.668	0.347	0.351	0.629	0.441
	Manhattan	No standardisation	0.627	0.181	0.189	0.467	0.304
		Z- score	0.853	0.671	0.674	0.782	0.641
		Range	0.856	0.675	0.678	0.783	0.643
		Maximum	0.717	0.453	0.456	0.700	0.513
	Canberra	No standardisation	0.709	0.421	0.424	0.667	0.484
		Z- score	0.732	0.459	0.462	0.683	0.505
		Range	0.709	0.421	0.424	0.667	0.484
		Maximum	0.709	0.421	0.424	0.667	0.484
	Minkowski	No standardisation	0.624	0.173	0.180	0.461	0.299
		Z- score	0.742	0.489	0.493	0.713	0.533
		Range	0.744	0.501	0.504	0.727	0.545
		Maximum	0.668	0.347	0.351	0.629	0.441

Appendix D.1: Summary and PCA results for S

```
Summary: S

      Area         Perimeter      Compactness    length_kern     width_kern     asy_coeff
 Min.   :10.6   Min.   :12.4   Min.   :  1   Min.   :4.9   Min.   :2.6   Min.   :0.8
 1st Qu.:12.3   1st Qu.:13.4   1st Qu.: 51   1st Qu.:5.3   1st Qu.:2.9   1st Qu.:2.6
 Median :14.4   Median :14.3   Median : 96   Median :5.5   Median :3.2   Median :3.6
 Mean   :14.8   Mean   :14.6   Mean   : 95   Mean   :5.6   Mean   :3.3   Mean   :3.7
 3rd Qu.:17.3   3rd Qu.:15.7   3rd Qu.:140   3rd Qu.:6.0   3rd Qu.:3.6   3rd Qu.:4.8
 Max.   :21.2   Max.   :17.2   Max.   :186   Max.   :6.7   Max.   :4.0   Max.   :8.5
  kern_groove  Label
 Min.   :4.5   1:70
 1st Qu.:5.0   2:70
 Median :5.2   3:70
 Mean   :5.4
 3rd Qu.:5.9
 Max.   :6.5

Variance: S
> var(area)
[1] 8.5
> var(per)
[1] 1.7
> var(comp)
[1] 2766
> var(lengthk)
[1] 0.2
> var(widthk)
[1] 0.14
> var(asyc)
[1] 2.3
> var(kerngr)
[1] 0.24

PCA Results:
Standard deviations (1, .., p=7):
[1] 2.2407 1.0828 0.8393 0.2686 0.1516 0.0776 0.0311

Rotation (n x k) = (7 x 7):
                PC1     PC2     PC3    PC4     PC5     PC6      PC7
Area         -0.445  0.0233 -0.0202  0.207 -0.1716  0.4019  0.753218
Perimeter    -0.442  0.0807  0.0616  0.283 -0.0639  0.5430 -0.644152
Compactness  -0.273 -0.5217 -0.6468 -0.355  0.3124  0.0965 -0.045437
length_kern  -0.424  0.2036  0.2069  0.207  0.7503 -0.3564  0.052816
width_kern   -0.433 -0.1203 -0.2021  0.259 -0.5228 -0.6365 -0.108009
asy_coeff     0.117  0.7230 -0.6720  0.104  0.0325  0.0133  0.000167
kern_groove  -0.388  0.3770  0.2055 -0.794 -0.1778 -0.0436 -0.034794

Importance of components:
                          PC1   PC2   PC3    PC4     PC5     PC6     PC7
Standard deviation      2.241 1.083 0.839 0.2686 0.15158 0.07757 0.03113
Proportion of Variance 0.717 0.168 0.101 0.0103 0.00328 0.00086 0.00014
Cumulative Proportion  0.717 0.885 0.985 0.9957 0.99900 0.99986 1.00000
```

Table D.3: Bootstrap sequential LRTS for S

Clusters	LRTS	Bootstrap p-value
1 vs 2	454.63	0.01
2 vs 3	192.58	0.01
3 vs 4	235.39	0.01
4 vs 5	80.92	0.21

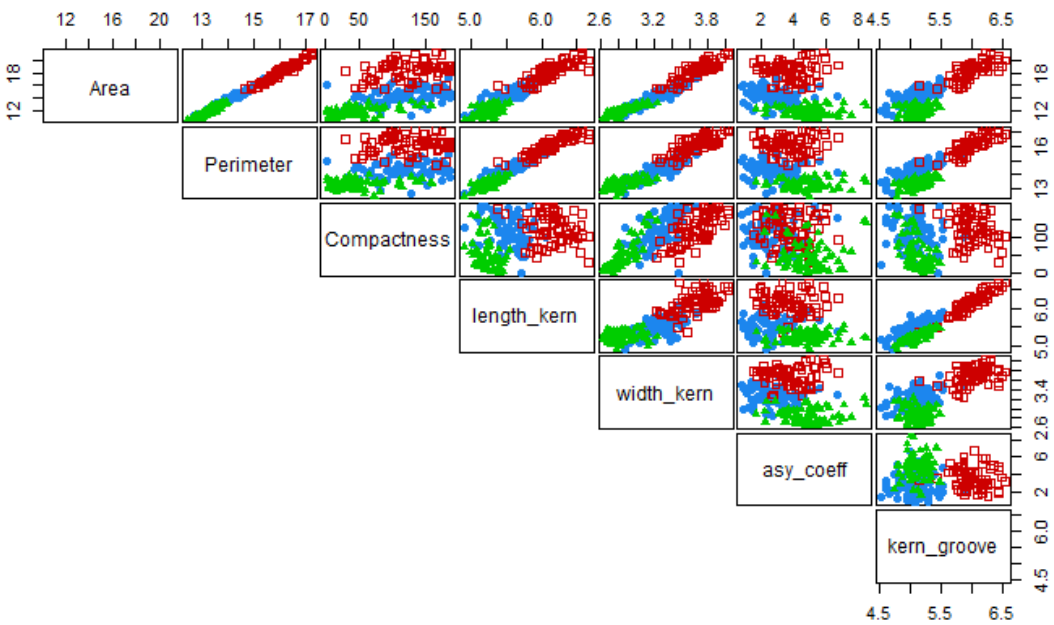

Figure D.2: Pairs plot for seeds data

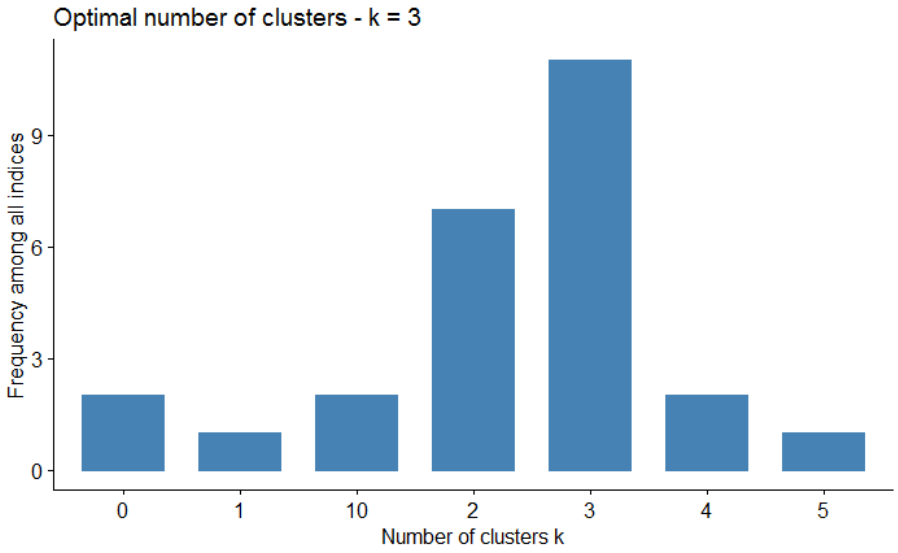

Figure D.3: Optimal number of clusters for S

Appendix D.2: Overlap parameters for seeds data

```
Overlap pameters:
$`OmegaMap`
      [,1]    [,2]    [,3]
[1,] 1.000 0.00519 0.01897
[2,] 0.003 1.00000 0.00011
[3,] 0.014 0.00014 1.00000

$BarOmega
[1] 0.014

$MaxOmega
[1] 0.033

$rcMax
[1] 1 3
```

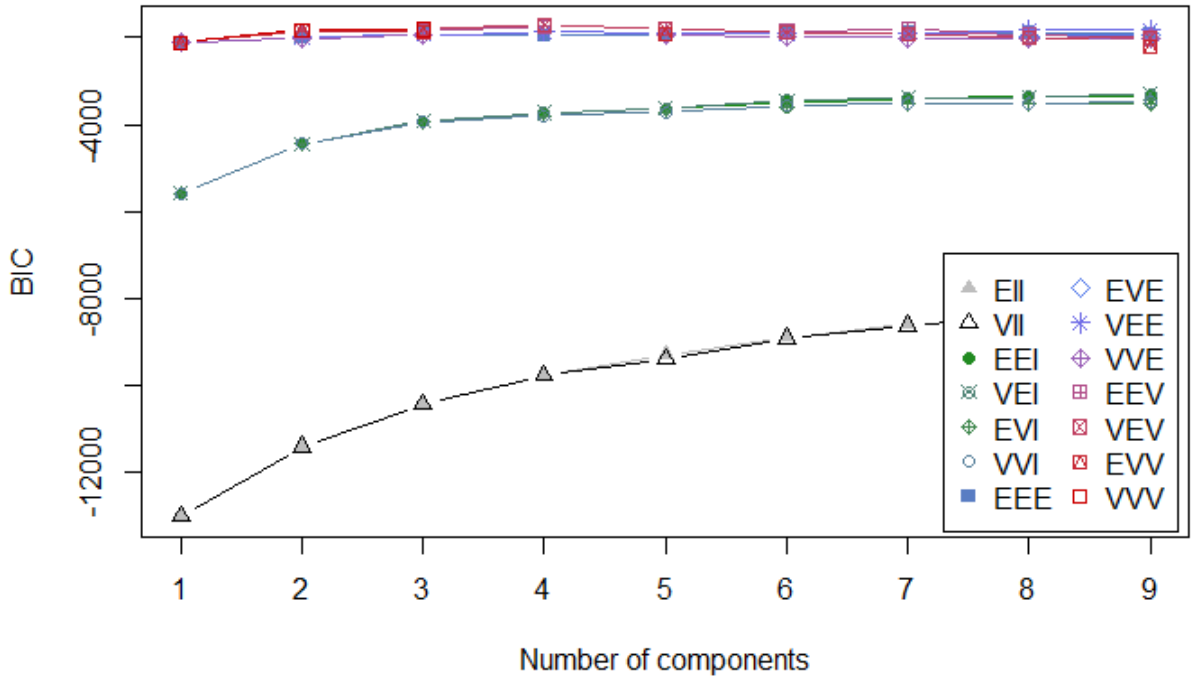

Figure D.4: BIC plot for seeds data

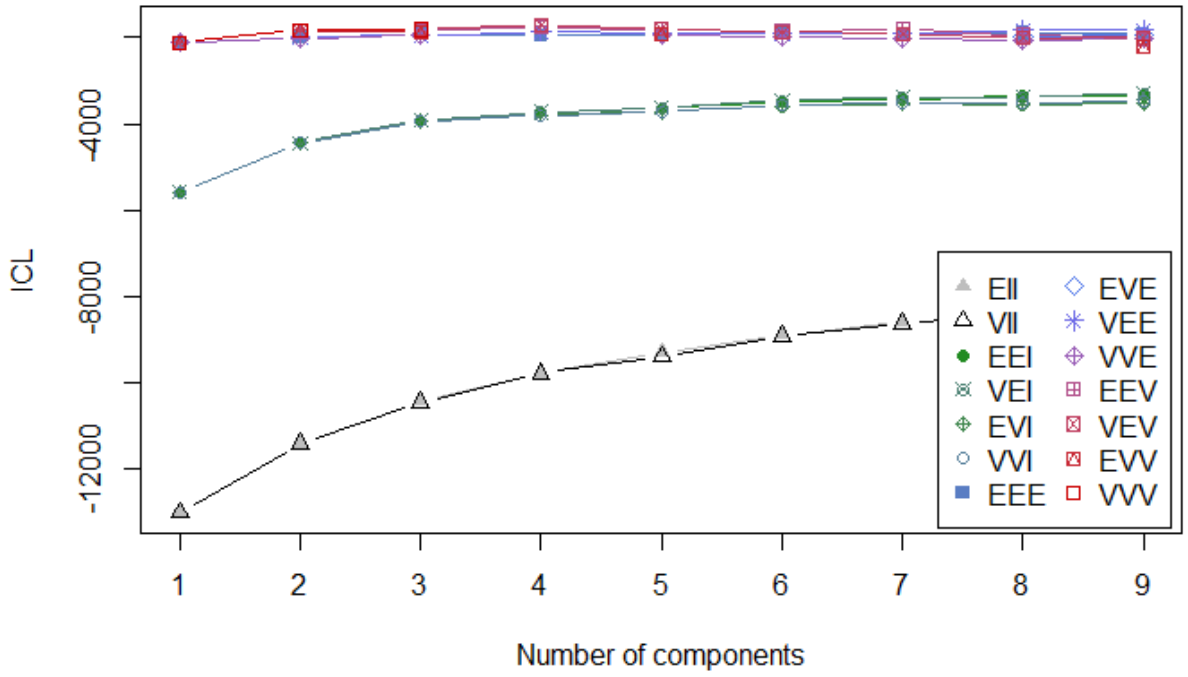

Figure D.5: ICL plot for seeds data

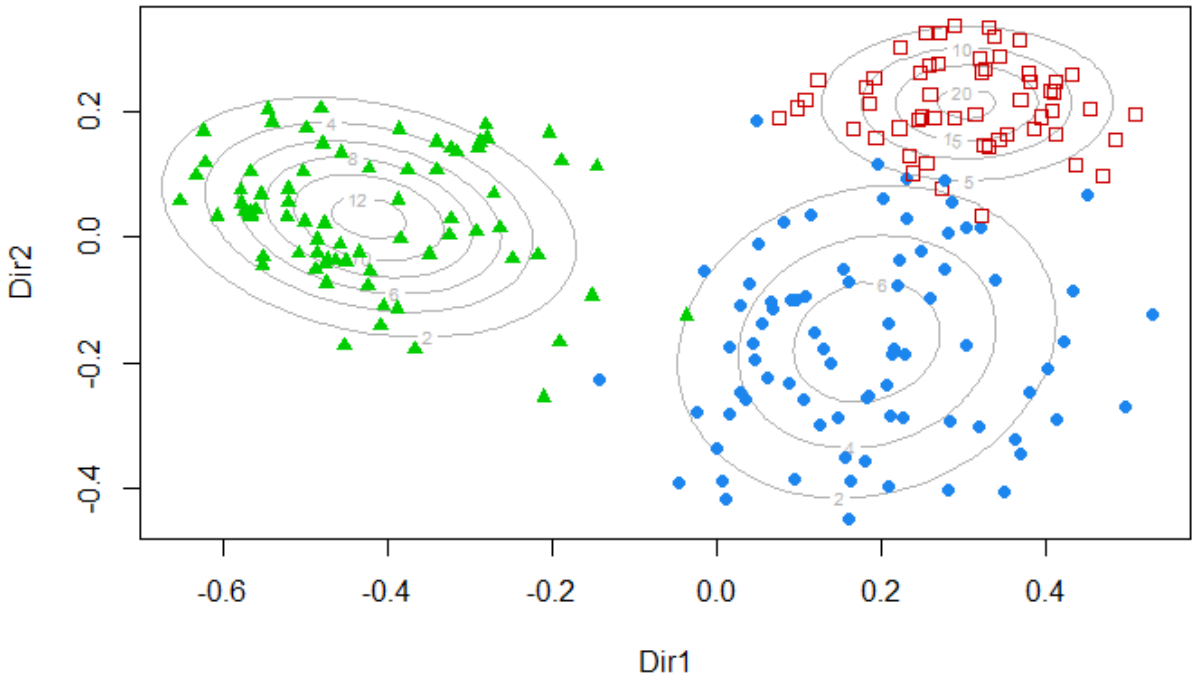

Figure D.6: LC cluster analysis: Contours Plot for 3 cluster solution seeds data

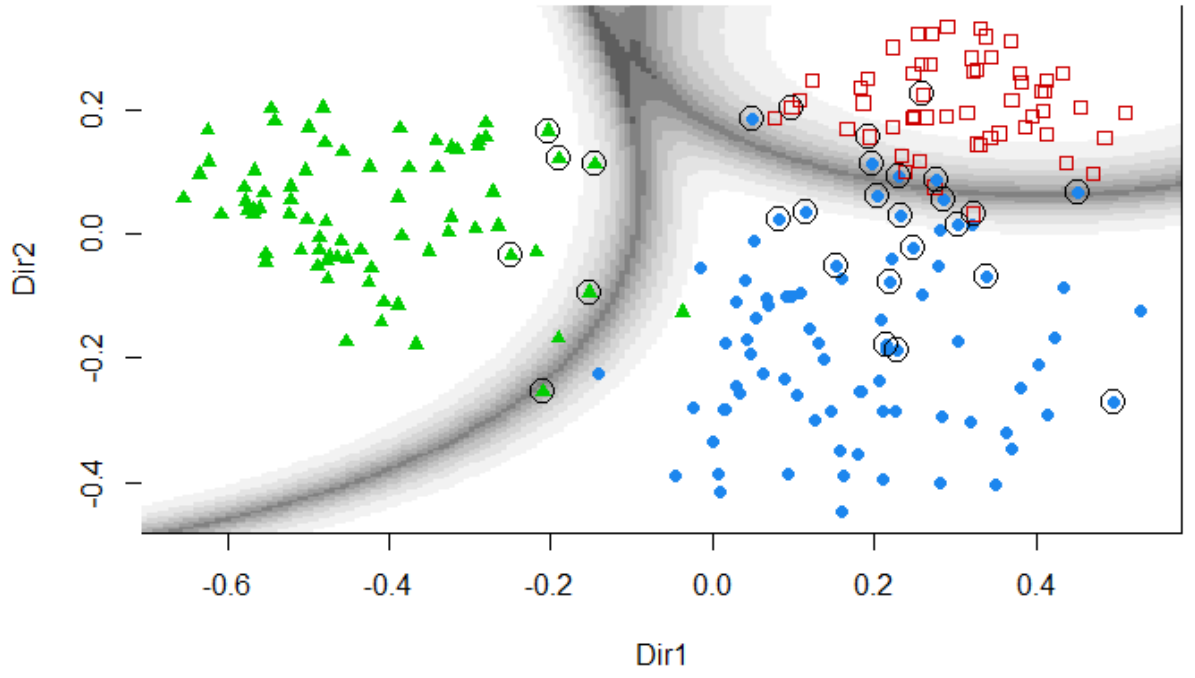

Figure D.7: LC cluster analysis: Uncertainty Plot for 3 cluster solution seeds data

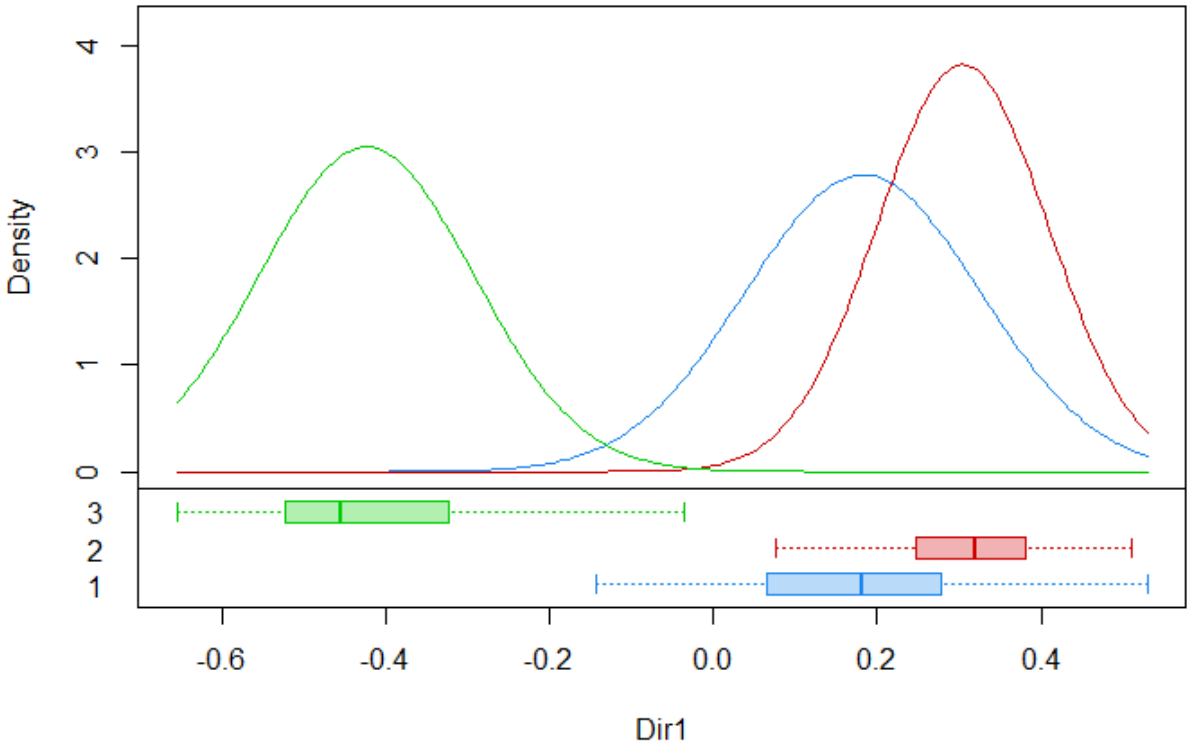

Figure D.8: LC cluster analysis: First dimension density plot for 3 cluster solution seeds data

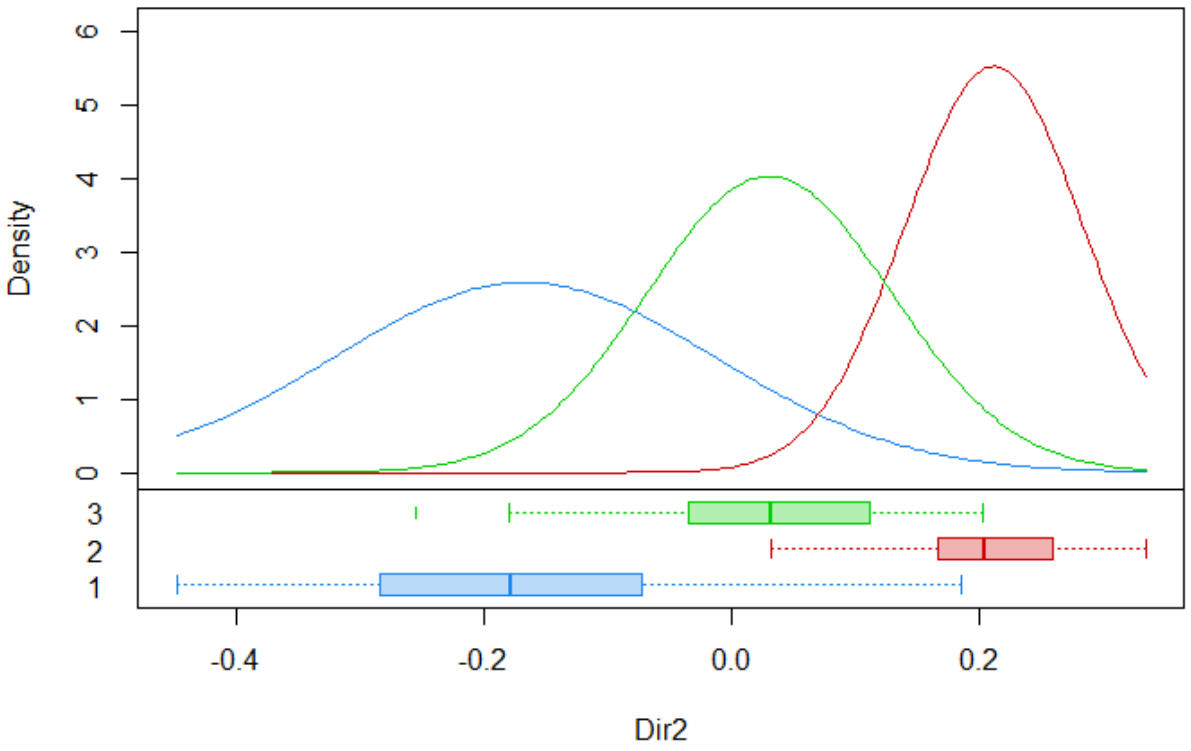

Figure D.9: LC cluster analysis: Second dimension plot for 3 cluster solution seeds data

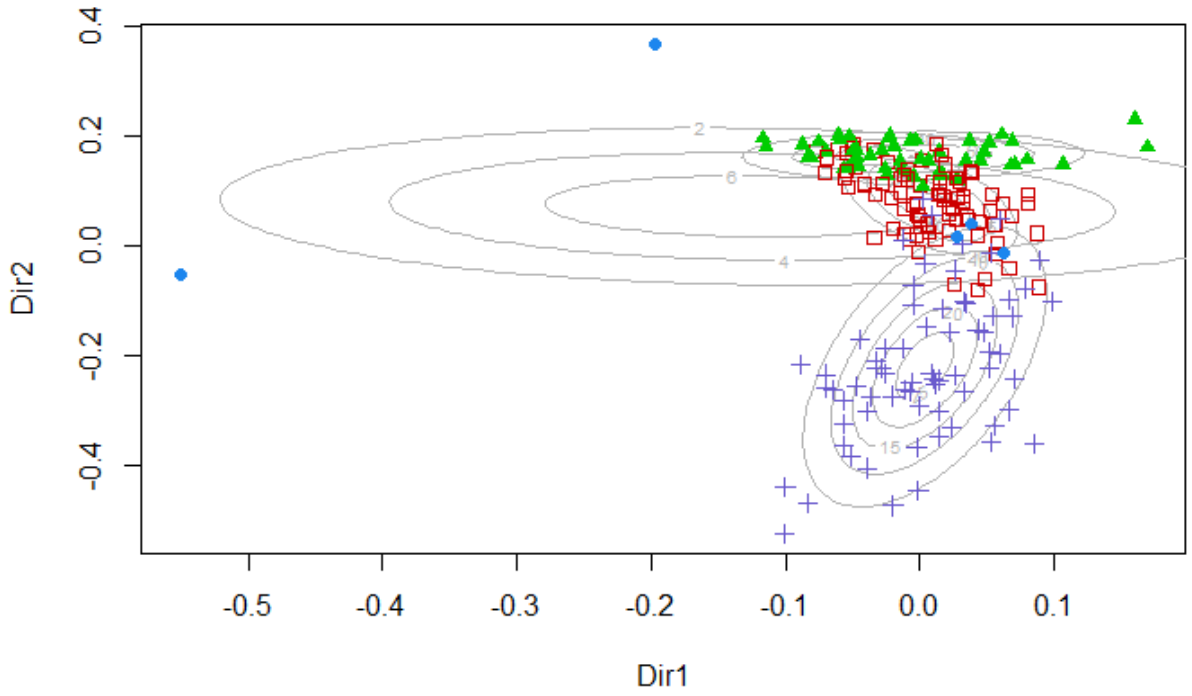

Figure D.10: LC cluster analysis: Contours Plot for 4 cluster solution seeds data

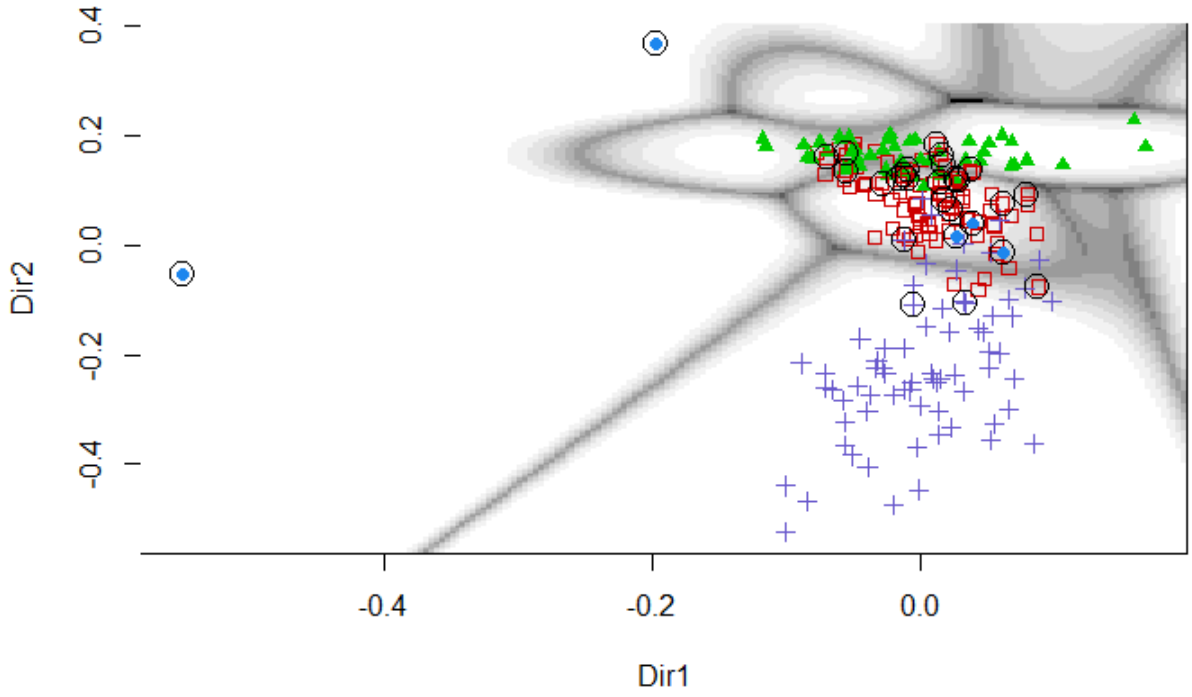

Figure D.11: LC cluster analysis: Uncertainty Plot for 4 cluster solution seeds data

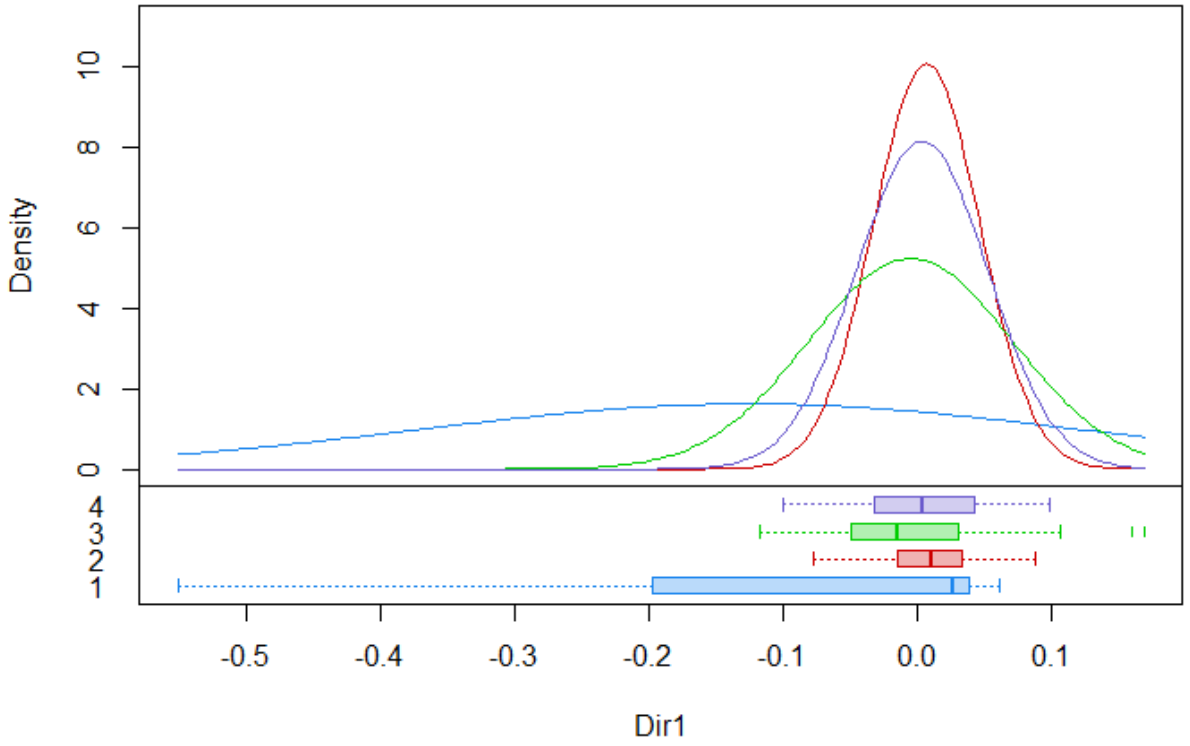

Figure D.12: LC cluster analysis: First dimension density plot for 4 cluster solution seeds data

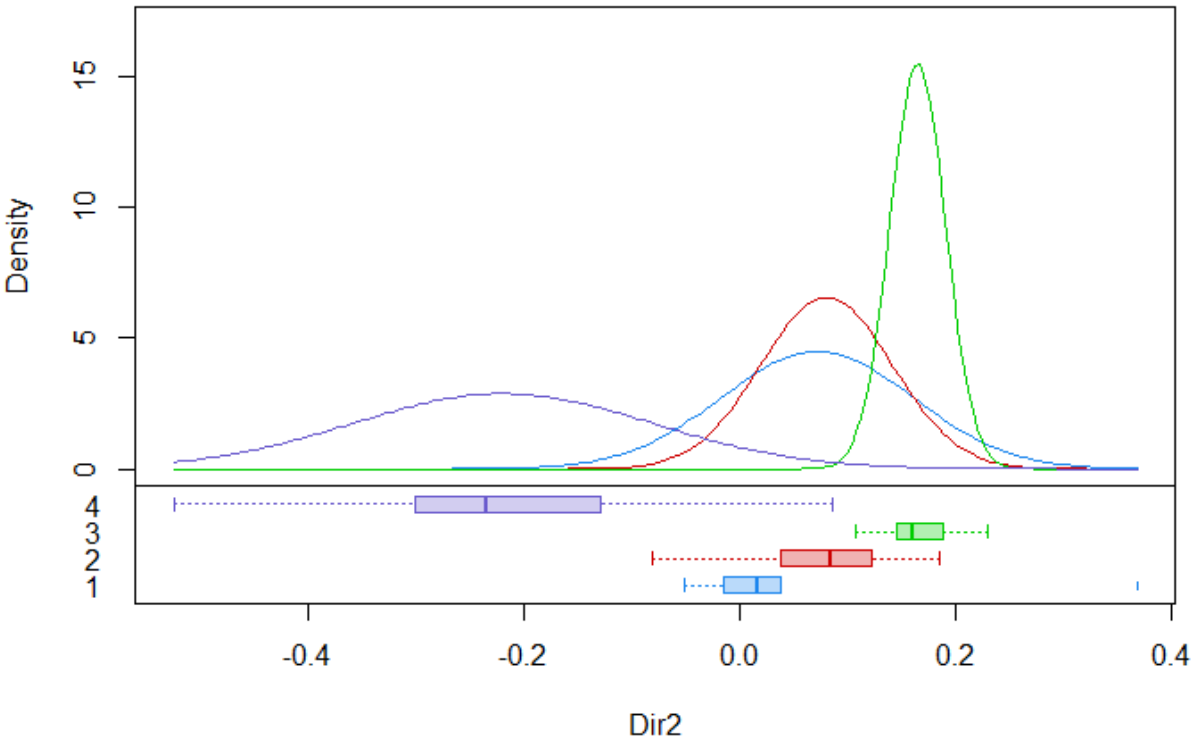

Figure D.13: LC cluster analysis: Second dimension plot for 4 cluster solution seeds data

Appendix D.3: LC cluster analysis model summary: 3 components

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VEV (ellipsoidal, equal shape) model with 3 components:


Clustering table:
 1  2  3
81 55 74

Mixing probabilities:
   1    2    3
0.39 0.26 0.35

Means:
              [,1] [,2]  [,3]
Area          13.9 11.7  18.2
Perimeter     14.1 13.2  16.1
Compactness 110.4 31.8 124.2
length_kern   5.4  5.2   6.1
width_kern    3.2  2.8   3.7
asy_coeff     3.0  4.9   3.6
kern_groove   5.0  5.2   6.0

Variances:
[,,1]
              Area Perimeter Compactness length_kern width_kern asy_coeff kern_groove
Area          1.69     0.809        12.9       0.269      0.217    -0.598       0.172
Perimeter     0.81     0.400         2.1       0.141      0.097    -0.303       0.093
Compactness 12.85     2.073      1677.9      -1.875      3.947     2.554      -2.605
length_kern  0.27     0.141        -1.9       0.057      0.027    -0.103       0.042
width_kern   0.22     0.097         3.9       0.027      0.034    -0.062       0.014
asy_coeff   -0.60    -0.303         2.6      -0.103     -0.062     2.549      -0.019
kern_groove  0.17     0.093        -2.6       0.042      0.014    -0.019       0.052
[,,2]
               Area Perimeter Compactness length_kern width_kern asy_coeff kern_groove
Area         0.5441    0.2555        7.72      0.0736     0.0828    0.0089      0.0647
Perimeter    0.2555    0.1405        0.33      0.0492     0.0287   -0.0053      0.0434
Compactness  7.7215    0.3346      668.21     -1.2733     2.8704    2.8941     -1.2003
length_kern 0.0736    0.0492       -1.27      0.0230     0.0041    0.0068      0.0204
width_kern   0.0828    0.0287        2.87      0.0041     0.0182    0.0170      0.0033
asy_coeff    0.0089   -0.0053        2.89      0.0068     0.0170    1.2266      0.0042
kern_groove 0.0647    0.0434       -1.20      0.0204     0.0033    0.0042      0.0249
[,,3]
               Area Perimeter Compactness length_kern width_kern asy_coeff kern_groove
Area          1.939    0.8222        12.0      0.2969     0.2126   -0.0127       0.269
Perimeter     0.822    0.3635         1.0      0.1386     0.0819   -0.0016       0.127
Compactness  12.019    0.9963      1267.6     -1.8103     3.6980   -2.3546      -2.077
length_kern  0.297    0.1386        -1.8      0.0648     0.0243   -0.0082       0.059
width_kern   0.213    0.0819         3.7      0.0243     0.0296    0.0067       0.020
asy_coeff   -0.013   -0.0016        -2.4     -0.0082     0.0067    1.1997      -0.012
kern_groove  0.269    0.1272        -2.1      0.0591     0.0201   -0.0119       0.064
```

Appendix D.4: LC cluster analysis model summary: 4 components

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VEV (ellipsoidal, equal shape) model with 4 components:


Clustering table:
 1  2  3  4
 5 84 51 70

Mixing probabilities:
    1     2     3     4
0.024 0.410 0.234 0.332

Means:
             [,1]   [,2] [,3]   [,4]
Area         14.8   13.8 11.6   18.4
Perimeter    14.7   14.0 13.2   16.2
Compactness  33.7  110.4 30.0  126.7
length_kern   5.7    5.4  5.2    6.2
width_kern    3.2    3.2  2.8    3.7
asy_coeff     2.6    3.2  4.8    3.6
kern_groove   5.4    5.1  5.1    6.0

Variances:
[,,1]
              Area Perimeter Compactness length_kern width_kern asy_coeff kern_groove
Area         0.615     0.264        0.24     0.10222    0.05817     0.082     0.18534
Perimeter    0.264     0.134        2.97     0.05950    0.01389     0.096     0.11015
Compactness  0.242     2.968      418.71     2.18635   -1.55168     9.301     4.20390
length_kern  0.102     0.060        2.19     0.03026    0.00013     0.075     0.05396
width_kern   0.058     0.014       -1.55     0.00013    0.01302    -0.043     0.00095
asy_coeff    0.082     0.096        9.30     0.07465   -0.04309     0.637     0.10872
kern_groove  0.185     0.110        4.20     0.05396    0.00095     0.109     0.10114
[,,2]
              Area Perimeter Compactness length_kern width_kern asy_coeff kern_groove
Area          1.45     0.691        16.0       0.226     0.1909    -0.562      0.1305
Perimeter     0.69     0.341         4.0       0.120     0.0845    -0.269      0.0748
Compactness  15.98     3.959      1366.2      -1.250     4.2097    -4.412     -2.6254
length_kern   0.23     0.120        -1.3       0.051     0.0226    -0.077      0.0363
width_kern    0.19     0.084         4.2       0.023     0.0304    -0.062      0.0095
asy_coeff    -0.56    -0.269        -4.4      -0.077    -0.0620     2.368      0.0144
kern_groove   0.13     0.075        -2.6       0.036     0.0095     0.014      0.0482
[,,3]
               Area Perimeter Compactness length_kern width_kern asy_coeff kern_groove
Area          0.589     0.288        6.26      0.0796     0.0830   -0.0720      0.0733
Perimeter     0.288     0.167       -0.99      0.0578     0.0281   -0.0396      0.0527
Compactness   6.262    -0.990      739.44     -2.1289     2.9279    1.2986     -1.8792
length_kern   0.080     0.058       -2.13      0.0267     0.0022   -0.0026      0.0249
width_kern    0.083     0.028        2.93      0.0022     0.0183    0.0042      0.0022
asy_coeff    -0.072    -0.040        1.30     -0.0026     0.0042    1.4426     -0.0079
kern_groove   0.073     0.053       -1.88      0.0249     0.0022   -0.0079      0.0300
[,,4]
              Area Perimeter Compactness length_kern width_kern asy_coeff kern_groove
Area         2.060    0.8858         7.6       0.344     0.2129   -0.0360       0.305
Perimeter    0.886    0.3968        -1.2       0.162     0.0826   -0.0068       0.145
Compactness  7.578   -1.1752      1297.0      -2.671     3.2997   -4.0608      -2.814
length_kern  0.344    0.1621        -2.7       0.077     0.0268   -0.0130       0.071
width_kern   0.213    0.0826         3.3       0.027     0.0286    0.0029       0.022
asy_coeff   -0.036   -0.0068        -4.1      -0.013     0.0029    1.2910      -0.014
kern_groove  0.305    0.1450        -2.8       0.071     0.0216   -0.0141       0.073
```

Appendix D.5: BIC and ICL values for seeds data

```
Bayesian Information Criterion (BIC):
     EII     VII    EEI    VEI    EVI    VVI    EEE    EVE    VEE    VVE    EEV    VEV    EVV    VVV
1 -13004 -13004 -5587 -5587 -5587 -5587 -2122 -2122 -2122 -2122 -2122 -2122 -2122 -2122
2 -11391 -11392 -4441 -4442 -4447 -4449 -2013 -2021 -2002 -2026 -1837 -1828 -1843 -1832
3 -10426 -10431 -3915 -3917 -3937 -3939 -1930 -1928 -1935 -1940 -1849 -1796 -1878 -1820
4  -9767  -9768 -3726 -3724 -3793 -3796 -1946     NA -1870     NA -1745 -1721     NA     NA
5  -9278  -9383 -3628 -3626 -3713 -3721 -1936     NA -1892 -1942 -1802 -1800 -1918     NA
6  -8891  -8909 -3478 -3453 -3582 -3560 -1891     NA -1905 -1971 -1860 -1872     NA     NA
7  -8584  -8612 -3412 -3386 -3525 -3530 -1894     NA -1896 -2004 -1806 -1915     NA     NA
8  -8412  -8439 -3341 -3370 -3535 -3528 -1886 -1963 -1831 -2037 -1942 -2010     NA     NA
9  -8170  -8231 -3344 -3300 -3527 -3484 -1908 -1931 -1827 -2003 -2058 -2002 -2219     NA

Top 3 models based on the BIC criterion:
VEV,4 EEV,4 VEV,3
-1721 -1745 -1796

Integrated Complete-data Likelihood (ICL) criterion:
     EII     VII    EEI    VEI    EVI    VVI    EEE    EVE    VEE    VVE    EEV    VEV    EVV    VVV
1 -13004 -13004 -5587 -5587 -5587 -5587 -2122 -2122 -2122 -2122 -2122 -2122 -2122 -2122
2 -11396 -11396 -4443 -4444 -4449 -4450 -2016 -2025 -2005 -2030 -1839 -1831 -1844 -1834
3 -10432 -10437 -3920 -3923 -3940 -3942 -1939 -1937 -1945 -1947 -1854 -1803 -1883 -1827
4  -9773  -9773 -3737 -3733 -3802 -3805 -1960     NA -1877     NA -1753 -1728     NA     NA
5  -9286  -9389 -3644 -3636 -3728 -3731 -1959     NA -1901 -1951 -1814 -1807 -1929     NA
6  -8899  -8916 -3491 -3462 -3593 -3566 -1908     NA -1917 -1986 -1866 -1878     NA     NA
7  -8591  -8618 -3430 -3398 -3538 -3538 -1914     NA -1912 -2022 -1811 -1921     NA     NA
8  -8420  -8444 -3361 -3388 -3548 -3535 -1907 -1986 -1844 -2060 -1948 -2015     NA     NA
9  -8178  -8239 -3360 -3311 -3539 -3492 -1930 -1943 -1840 -2020 -2066 -2008 -2226     NA

Top 3 models based on the ICL criterion:
VEV,4 EEV,4 VEV,3
-1728 -1753 -1803
```

Appendix E

R Code for P

```
POORLY SEPARATED CLUSTERS (P)

```{r message=FALSE, warning=FALSE, include=FALSE}
#Required packages.
library(cluster)
#Package for standardisation methods.
library(clusterSim)
#Package for K-means
library(stats)
library(NbClust)
library(factoextra)
library(fpc)
#K-means with different distance measures.
library(amap)
library(tidyverse, warn.conflicts = FALSE)
library(hrbrthemes)
#MixSim package for simulating Gaussian mixtures.
library(MASS)
library(MixSim)
#For finite mixture modelling.
library(mclust)
#For plotting PCA plot
library(devtools)
library(ggplot2)
library(ggbiplot)
library(plyr)
library(dplyr)
library(clues)
```
#Poorly Separated Clusters
```{r}
#Setting a seed for replication of results.
#Creating a mixture component with specified parameters.
set.seed(1995)
Q=MixSim(BarOmega = 0.17,MaxOmega=0.2,K=3, p=5,sph=TRUE,hom=FALSE, PiLow = 0.2,
int = c(30.0,130),resN=1000)
#Simulating the data from the mixture.
C=simdataset(n=500,Pi=Q$Pi,Mu=Q$Mu,S=Q$S)
```
```

```
```{r}
#Extracting the variables from the data set.
a=C$X[,1]
b=C$X[,2]
c=C$X[,3]
d=C$X[,4]
e=C$X[,5]
class=C$id
mix1=cbind(class,a,b,c,d,e)
```
```{r}
#Converting mix3 to a data frame and the class to a factor.
mix1=as.data.frame(mix1)
mix1[, 1] <- as.factor(mix1[, 1])
str(mix1)
summary(mix1)
var(mix1$a)
var(mix1$b)
var(mix1$c)
var(mix1$d)
var(mix1$e)
```
```{r}
#Column for the  class is removed, the model is fitted on the remaining columns.
class=mix1$class
X<-mix1[,-1]
```
#Box Plots
```{r}
#Box plots for all the variables.
par(mfrow=c(1,5))
boxplot(a, main="a",ylim=c(-30,180))
boxplot(b, main="b",ylim=c(-30,180))
boxplot(c, main="c",ylim=c(-30,180))
boxplot(d, main="d",ylim=c(-30,180))
boxplot(e, main="e",ylim=c(-30,180))
```
#Histograms
```{r}
#Histograms
par(mfrow=c(1,3))
hist(a, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.02),breaks=8)# prob=TRUE
for probabilities not counts
lines(density(a), col="red", lwd=2) # add a density estimate with defaults

hist(b, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.02),breaks=8)# prob=TRUE
for probabilities not counts
lines(density(b), col="red", lwd=2) # add a density estimate with defaults

hist(c, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.02),breaks=8)# prob=TRUE
for probabilities not counts
lines(density(c), col="red", lwd=2) # add a density estimate with defaults
par(mfrow=c(1,2))
hist(d, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.02),breaks=8)# prob=TRUE
for probabilities not counts
lines(density(d), col="red", lwd=2) # add a density estimate with defaults
```
```

```
hist(e, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.017),breaks=8)# prob=TRUE
for probabilities not counts
lines(density(e), col="red", lwd=2) # add a density estimate with defaults
```
#Principal Components Analysis
```{r}
# apply PCA - scale. = TRUE is highly
# advisable, but default is FALSE.
ir.pca <- prcomp(X, center = TRUE, scale. = TRUE)
```
```{r}
# print method
print(ir.pca)
```
```{r}
# plot method
plot(ir.pca, type = "l", main=NULL)
```
```{r}
# summary method
summary(ir.pca)
```
```{r}
g <- ggbiplot(ir.pca, obs.scale = 1, var.scale = 1, groups = class, ellipse = TRUE,
circle = FALSE)
g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal',legend.position = 'top')
print(g)
```
```{r}
clPairs(X,class,lower.panel = NULL)
#Pairs plot for variable showing different classes.
```
##Distance Matrix (Assesing whether is clusterable).
```{r}
#Standadising the data using cluster sim package.
X1= data.Normalization (X,type="n8",normalization="column")
#Setting Euclidean distance, stand is for standadising data. Set to false as data
has been standardised.
dist.eucl= get_dist(X1,method = "euclidean", stand=FALSE)
#Visualising the distance matrix.
fviz_dist(dist.eucl,show_labels=FALSE,order = TRUE)+
labs(title = "Distance Matrix: Euclidean Distance")
#lower value means the items are similar. Red squares indicate clusters.
```
#Hierarchical Clustering
```{r}
#Standadising the data using cluster sim package.
X2= data.Normalization (X,type="n8",normalization="column")

# Dissimilarity matrix. Distance Method
d <- dist(X2, method = "manhattan")

# Hierarchical clustering using different Linkages
hc1 <- hclust(d, method = "ward.D" )

plot(hc1, labels=FALSE) # display dendogram
groups <- cutree(hc1, k=3) # cut tree into k clusters
```

```
# draw dendogram with red borders around the k clusters
rect.hclust(hc1, k=3, border="red")

classT=as.vector(mix1$class)
adjustedRand(classT, groups)
```
```
#Determining the number of clusters to use for K-means
```{r message=FALSE, warning=FALSE}
#Standadising the data using cluster sim package.
X3= data.Normalization (X,type="n0",normalization="column")

## Elbow Method Using FactoExtra
fviz_nbclust(X3, FUNcluster = kmeans, method = "wss")

## Silhouette

# Silhouette method in FactoExtra
fviz_nbclust(X3, kmeans, method = "silhouette") + theme_classic()

## Gap Statistic

### GAP STAT Using Facto Extra
fviz_nbclust(X3, kmeans, nstart = 25, method = "gap_stat", nboot = 500) +
#nboot is # of bootstrap sample - must be included
labs(subtitle = "Gap Statistic Method")

#NbClust uses 30 different indices for choosing the best number of clusters.
nb <- NbClust(X3, distance = "euclidean", min.nc = 2,
max.nc = 10, method = "kmeans")
fviz_nbclust(nb)
```
```
#K-means analysis with 2 or 3 clusters
```{r}
X4= data.Normalization (X,type="n0",normalization="column")
#Fitting a 2 cluster solution.
C2=Kmeans(X4,2, iter.max = 500, nstart = 200,method = "euclidean")
#Fitting a 3 cluster solution
C3=Kmeans(X4,3, iter.max = 500, nstart = 200,method = "euclidean")
#Rand index for 2 cluster solution.
# Compute cluster stats
label1 <- as.numeric(C$id)
clust_stats2 <- cluster.stats(d=dist(X4),label1, C2$cluster)
# Corrected Rand index
adjustedRand(classT, C2$cluster)
#Variation of information
clust_stats2$vi
#Rand index for 3 cluster solution.
# Compute cluster stats
clust_stats3 <- cluster.stats(d=dist(X4),label1, C3$cluster)
# Corrected Rand index
adjustedRand(classT, C3$cluster)
#Variation of information
clust_stats3$vi
```
```
#Visualising K-Means Results.
## 2 Cluster solution.
```{r}
fviz_cluster(C2, data = X4,
```
```

```
ellipse.type = "norm", # Concentration ellipse
star.plot = FALSE, # Add segments from centroids to items
repel = TRUE, # Avoid label overplotting (slow)
ggtheme = theme_minimal()
)
```
3 Cluster solution.
```{r}
fviz_cluster(C3, data = X4,
ellipse.type = "norm", # Concentration ellipse
star.plot = FALSE, # Add segments from centroids to items
repel = TRUE, # Avoid label overplotting (slow)
ggtheme = theme_minimal()
)
```
#Finite Mixture Modelling.

```{r}
#Using EM to calculate BIC values for different clusters and parameterizations.
To be used for selecting the best model.
BIC<-mclustBIC(X)
#Plotting the BIC values.
plot(BIC)
#Summary of the BIC values giving the top 3 BIC values.
summary(BIC)
```
3 cluster solution.
```{r}
options(digits=2)
#Optimal model according to the BIC values
mod1<-Mclust(X,x=BIC)
#Summary of the best model.
summary(mod1, parameters=TRUE)
```
Dimension reduction for model 3 cluster model
```{r}
#Dimension reduction model tovisualise the clustering structure and geometric
characteristics.
dr1 <- MclustDR(mod1)
#Summary of the dimension reduction.
summary(dr1)
#Plot of the model in 2 dimensions showing the contours.
plot(dr1, what = "contour")
plot(dr1, what = "boundaries", ngrid = 200)
miscl <- classError(class, mod1$classification)$misclassified
points(dr1$dir[miscl,], pch = 1, cex = 2)
#Plots of the densities in the first and second dimension. They show which
dimension separates which clusters very well.
plot(dr1, what = "density", dimens = 1)
plot(dr1, what = "density", dimens = 2)
plot(dr1, what = "evalues")
#Pairs plot for the dimensions. They show the dimensions that are important
in separating the clusters.
plot(dr1, what = "pairs",lower.panel = NULL)
```
##Fitting a 4 cluster solution model.
```{r}
#Summary of the 4 cluster solution models.
```

```
summary(BIC,data=X, G=4)
```
```{r}
#Optimal model according to the BIC values
mod2<-Mclust(X,x=BIC,G=4)
#Summary of the best model.
summary(mod2)
summary(mod2, parameters=TRUE)
```
```{r}
#Dimension reduction model tovisualise the clustering structure and geometric
characteristics.
dr2 <- MclustDR(mod2,lambda = 1)
#Summary of the dimension reduction.
summary(dr2)
#Plot of the model in 2 dimensions showing the contours.
plot(dr2, what = "contour")
plot(dr2, what = "boundaries", ngrid = 200)
miscl <- classError(class, mod1$classification)$misclassified
points(dr2$dir[miscl,], pch = 1, cex = 2)
#Plots of the densities in the first and second dimension.
They show which dimension separates which clusters very well.
plot(dr2, what = "density", dimens = 1)
plot(dr2, what = "density", dimens = 2)
plot(dr2, what = "evalues")
#Pairs plot for the dimensions. They show the dimensions that are important
in separating the clusters.
plot(dr2, what = "pairs")
```
```{r}
#Adjusted Rand Index
adjustedRand(classT, mod1$classification)
adjustedRand(classT, mod2$classification)
```
```{r}
#Computing and displaying the BIC values for all covariance stuctures.
BIC <- mclustBIC(X)
BIC
```
```{r}
#Integrated complete data likelihood
ICL<-mclustICL(X)
ICL
summary(ICL)
```
```{r}
#Plot of the ICL for all parameterisation.
plot(ICL)
```
```{r}
#Bootstrap sequential LRT for the number of mixture components
LRT1 <- mclustBootstrapLRT(X, modelName = "VII",  nboot = 99)
LRT2 <- mclustBootstrapLRT(X, modelName = "VEI",  nboot = 99)
LRT1
LRT2
```

Appendix F

R Code for M

MODERATELY SEPARATED CLUSTERS (M)

```{r message=FALSE, warning=FALSE, include=FALSE}
#Required packages.
library(cluster)
#Package for standardisation methods.
library(clusterSim)
#Package for K-means
library(stats)
library(NbClust)
library(factoextra)
library(fpc)
#K-means with different distance measures.
library(amap)
library(tidyverse, warn.conflicts = FALSE)
library(hrbrthemes)
#MixSim package for simulating Gaussian mixtures.
library(MASS)
library(MixSim)
#For finite mixture modelling.
library(mclust)
#For plotting PCA plot
library(devtools)
library(ggplot2)
library(ggbiplot)
library(plyr)
library(dplyr)
library(clues)
```
#Moderately Separated Clusters
```{r}
#Setting a seed for replication of results.
#Creating a mixture component with specified parameters.
set.seed(2008)
Q=MixSim(BarOmega = 0.07,MaxOmega=0.09,K=4, p=4,sph=TRUE,hom=FALSE, PiLow = 0.1,
int = c(30.0,130),resN=1000)
#Simulating the data from the mixture.
A=simdataset(n=600,Pi=Q$Pi,Mu=Q$Mu,S=Q$S)
```

```
#Extracting the variables from the data set.
a=A$X[,1]
b=A$X[,2]
c=A$X[,3]
d=A$X[,4]
class=A$id
mix1=cbind(class,a,b,c,d)
#Converting mix1 to a data frame and the class to a factor.
mix1=as.data.frame(mix1)
mix1[, 1] <- as.factor(mix1[, 1])
class=mix1$class
X<-mix1[,-1]
summary(mix1)
# The column for the  class is removed and the model is fitted on the remaining columns.
var(mix1$a)
var(mix1$b)
var(mix1$c)
var(mix1$d)
```
```
#Box Plots
```{r}
#Box plots for all the variables.
par(mfrow=c(1,4))
boxplot(a, main="a",ylim=c(-30,170))
boxplot(b, main="b",ylim=c(-30,170))
boxplot(c, main="c",ylim=c(-30,170))
boxplot(d, main="d",ylim=c(-30,170))
```
```
#Histograms
```{r}
#Histograms
par(mfrow=c(1,4))
hist(a, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.025),breaks=8)
# prob=TRUE for probabilities not counts
lines(density(a), col="red", lwd=2) # add a density estimate with defaults

hist(b, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.025),breaks=8)
# prob=TRUE for probabilities not counts
lines(density(b), col="red", lwd=2) # add a density estimate with defaults

hist(c, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.025),breaks=8)
# prob=TRUE for probabilities not counts
lines(density(c), col="red", lwd=2) # add a density estimate with defaults

hist(d, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.026),breaks=8)
# prob=TRUE for probabilities not counts
lines(density(d), col="red", lwd=2) # add a density estimate with defaults
```
```
#Principal Components Analysis

```{r}
# apply PCA - scale. = TRUE is highly
# advisable, but default is FALSE.
ir.pca <- prcomp(X, center = TRUE, scale. = TRUE)
```
```
```{r}
# print method
print(ir.pca)
```

```
```{r}
# plot method
plot(ir.pca, type = "l", main= NULL)
```
```{r}
# summary method
summary(ir.pca)
```
```{r}
g <- ggbiplot(ir.pca, obs.scale = 1, var.scale = 1, groups = class, ellipse = TRUE,
circle = FALSE)
g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal',legend.position = 'top')
print(g)
```
```{r}
clPairs(X,class,lower.panel = NULL)
#Pairs plot for variable showing different classes.
```
##Distance Matrix (Assesing whether is clusterable).
```{r}
#Standadising the data using cluster sim package.
X1= data.Normalization (X,type="n8",normalization="column")
#Setting Euclidean distance, stand is for standadising data.
Set to false as data has been standardised.
dist.eucl= get_dist(X1,method = "manhattan", stand=FALSE)
#Visualising the distance matrix. Red indicates high similarity and blue low similarity.
fviz_dist(dist.eucl,show_labels=FALSE,order = TRUE)+ labs(title = "Distance Matrix")
#lower value means the items are similar. Red squares indicate clusters.
```
#Hierarchical Clustering
```{r}
#Standadising the data using cluster sim package.
X2= data.Normalization (X,type="n8",normalization="column")
# Dissimilarity matrix. Distance Method
d <- dist(X2, method = "manhattan")
# Hierarchical clustering using different Linkages
hc1 <- hclust(d, method = "ward.D" )
plot(hc1, labels=FALSE) # display dendogram
groups <- cutree(hc1, k=4) # cut tree into k clusters
# draw dendogram with red borders around the k clusters
rect.hclust(hc1, k=4, border="red")
classT=as.vector(mix1$class)
adjustedRand(classT, groups)
```
#Determining the number of clusters to use for K-means
```{r message=FALSE, warning=FALSE}
#Standadising the data using cluster sim package.
X3= data.Normalization (X,type="n0",normalization="column")

## Elbow Method Using FactoExtra
fviz_nbclust(X3, FUNcluster = kmeans, method = "wss")

## Silhouette

# Silhouette method in FactoExtra
fviz_nbclust(X3, kmeans, method = "silhouette") + theme_classic()
```
```

```
## Gap Statistic

### GAP STAT Using Facto Extra
fviz_nbclust(X3, kmeans, nstart = 25, method = "gap_stat", nboot = 500) +
#nboot is # of bootstrap sample - must be included
labs(subtitle = "Gap Statistic Method")

#NbClust uses 30 different indices for choosing the best number of clusters.
nb <- NbClust(X3, distance = "euclidean", min.nc = 2,
max.nc = 10, method = "kmeans")
fviz_nbclust(nb)
```

```
#K-means analysis with 2,3 or 4 clusters
```{r}
X4= data.Normalization (X,type="n0",normalization="column")
#Fitting a 2 cluster solution.
C2=Kmeans(X4,2, iter.max = 500, nstart = 200,method = "euclidean")
#Fitting a 3 cluster solution
C3=Kmeans(X4,3, iter.max = 500, nstart = 200,method = "euclidean")
#Fitting a 4 cluster solution
C4=Kmeans(X4,4, iter.max = 500, nstart = 200,method = "euclidean")

#Rand index for 2 cluster solution.
# Compute cluster stats
label1 <- as.numeric(A$id)
clust_stats2 <- cluster.stats(d=dist(X4),label1, C2$cluster)
# Corrected Rand index
adjustedRand(classT, C2$cluster)
#Variation of information
clust_stats2$vi

#Rand index for 3 cluster solution.
# Compute cluster stats
clust_stats3 <- cluster.stats(d=dist(X4),label1, C3$cluster)
# Corrected Rand index
adjustedRand(classT, C3$cluster)
#Variation of information
clust_stats3$vi

#Rand index for 4 cluster solution.
# Compute cluster stats
clust_stats4 <- cluster.stats(d=dist(X4),label1, C4$cluster)
# Corrected Rand index
adjustedRand(classT, C4$cluster)
#Variation of information
clust_stats4$vi
```

```
#Visualising K-Means Results.
## 2 Cluster solution.
```{r}
fviz_cluster(C2, data = X4,
ellipse.type = "norm", # Concentration ellipse
star.plot = FALSE, # Add segments from centroids to items
repel = TRUE, # Avoid label overplotting (slow)
ggtheme = theme_minimal(),
```

```
geom = "point", show.clust.cent="TRUE")
```
3 Cluster solution.
```{r}
fviz_cluster(C3, data = X4,
ellipse.type = "norm", # Concentration ellipse
star.plot = FALSE, # Add segments from centroids to items
repel = TRUE, # Avoid label overplotting (slow)
ggtheme = theme_minimal(),
geom = "point", show.clust.cent="TRUE")
```
4 Cluster solution.
```{r}
fviz_cluster(C4, data = X4,
ellipse.type = "norm", # Concentration ellipse
star.plot = FALSE, # Add segments from centroids to items
repel = TRUE, # Avoid label overplotting (slow)
ggtheme = theme_minimal(),
geom = "point", show.clust.cent="TRUE")
```
#Finite Mixture Modelling.

```{r}
#Using EM to calculate BIC values for different clusters and parameterizations.
BIC<-mclustBIC(X)
#Plotting the BIC values.
plot(BIC)
#Summary of the BIC values giving the top 3 BIC values.
summary(BIC)
```
5 cluster solution.
```{r}
options(digits = 3)
#Optimal model according to the BIC values
mod1<-Mclust(X,x=BIC)
#Summary of the best model.
summary(mod1, parameters=TRUE)
```
Dimension reduction for model 5 cluster model
```{r}
#To visualise the clustering structure and geometric characteristics.
dr1 <- MclustDR(mod1)
#Summary of the dimension reduction.
summary(dr1)
#Plot of the model in 2 dimensions showing the contours.
plot(dr1, what = "contour")
plot(dr1, what = "boundaries", ngrid = 200)
miscl <- classError(class, mod1$classification)$misclassified
points(dr1$dir[miscl,], pch = 1, cex = 2)
#Plots of the densities in the first and second dimension.
plot(dr1, what = "density", dimens = 1)
plot(dr1, what = "density", dimens = 2)
plot(dr1, what = "evalues")
#Pairs plot for the dimensions.
plot(dr1, what = "pairs",lower.panel = NULL)
```
##Fitting a 4 cluster solution model.
```{r}


```

```
#Optimal model according to the BIC values
mod2<-Mclust(X,x=BIC,G=4)
#Summary of the best model.
summary(mod2)
summary(mod2, parameters=TRUE)
```
```{r}
#To visualise the clustering structure and geometric characteristics.
dr2 <- MclustDR(mod2,lambda = 1)
#Summary of the dimension reduction.
summary(dr2)
#Plot of the model in 2 dimensions showing the contours.
plot(dr2, what = "contour")
plot(dr2, what = "boundaries", ngrid = 200)
miscl <- classError(class, mod1$classification)$misclassified
points(dr2$dir[miscl,], pch = 1, cex = 2)
#Plots of the densities in the first and second dimension.
plot(dr2, what = "density", dimens = 1)
plot(dr2, what = "density", dimens = 2)
plot(dr2, what = "evalues")
#Pairs plot for the dimensions.
plot(dr2, what = "pairs")
```
```{r}
#6 Cluster model according to the BIC values
mod3<-Mclust(X,x=BIC,G=6)
```
```{r}
#Adjusted Rand Index
adjustedRand(classT, mod1$classification) #5 cluster model
adjustedRand(classT, mod2$classification) #4 cluster model
adjustedRand(classT, mod3$classification) #6 cluster model
```
```{r}
#Computing and displaying the BIC values for all covariance stuctures.
BIC <- mclustBIC(X)
BIC
```
```{r}
#Integrated complete data likelihood.
ICL<-mclustICL(X)
ICL
summary(ICL)
```
```{r}
#Plot of the ICL for all parameterisation.
plot(ICL)
```
```{r}
#Bootstrap sequential LRT for the number of mixture components
LRT1 <- mclustBootstrapLRT(X, modelName = "VII",  nboot = 99)
LRT2 <- mclustBootstrapLRT(X, modelName = "VEI",  nboot = 99)
LRT1
LRT2
```

Appendix G

R Code for W

WELL SEPARATED CLUSTERS (W)

```{r message=FALSE, warning=FALSE, include=FALSE}
#Required packages.
library(cluster)
#Package for standardisation methods.
library(clusterSim)
#Package for K-means
library(stats)
library(NbClust)
library(factoextra)
library(fpc)
#K-means with different distance measures.
library(amap)
library(tidyverse, warn.conflicts = FALSE)
library(mclust)
library(hrbrthemes)
#MixSim package for simulating Gaussian mixtures.
library(MASS)
library(MixSim)
#For finite mixture modelling.
library(mclust)
#For plotting PCA plot
library(devtools)
library(ggplot2)
library(ggbiplot)
library(plyr)
library(dplyr)
library(clues)
```

##Well Separated Clusters

```{r}
#Setting a seed for replication of results.
#Creating a mixture component with specified parameters.
set.seed(2013)
Q=MixSim(BarOmega = 0.03,MaxOmega=0.045,K=3, p=4,sph=TRUE,hom=FALSE, PiLow = 0.2,
int = c(30.0,130),resN=1000)
#Simulating the data from the mixture.
```

```
A=simdataset(n=700,Pi=Q$Pi,Mu=Q$Mu,S=Q$S)
#Extracting the variables from the data set.
a=A$X[,1]
b=A$X[,2]
c=A$X[,3]
d=A$X[,4]
class=A$id
mix1=cbind(class,a,b,c,d)
#Converting mix1 to a data frame and the class to a factor.
mix1=as.data.frame(mix1)
mix1[, 1] <- as.factor(mix1[, 1])
class=mix1$class
# The column for the  class is removed and the model is fitted on the remaining columns.
X<-mix1[,-1]
summary(mix1)
var(mix1$a)
var(mix1$b)
var(mix1$c)
var(mix1$d)
```
```
##Box Plots
```{r}
#Box plots for all the variables.
par(mfrow=c(1,4))
boxplot(a, main="a",ylim=c(-20,170))
boxplot(b, main="b",ylim=c(-20,170))
boxplot(c, main="c",ylim=c(-20,170))
boxplot(d, main="d",ylim=c(-20,170))
```
```
##Histograms
```{r}
#Histograms
par(mfrow=c(1,4))
hist(a, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.025),breaks=8)
# prob=TRUE for probabilities not counts
lines(density(a), col="red", lwd=2) # add a density estimate with defaults

hist(b, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.02),breaks=8)
# prob=TRUE for probabilities not counts
lines(density(b), col="red", lwd=2) # add a density estimate with defaults

hist(c, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.025),breaks=8)
# prob=TRUE for probabilities not counts
lines(density(c), col="red", lwd=2) # add a density estimate with defaults

hist(d, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.02),breaks=8)
# prob=TRUE for probabilities not counts
lines(density(d), col="red", lwd=2) # add a density estimate with defaults
```
```
##Principal Components Analysis
```{r}
# apply PCA - scale. = TRUE is highly
# advisable, but default is FALSE.
ir.pca <- prcomp(X, center = TRUE, scale. = TRUE)
```
```
```{r}
# print method
print(ir.pca)
```

```
```{r}
# plot method
plot(ir.pca, type = "l",main=NULL)
```
```{r}
# summary method
summary(ir.pca)
```
```{r}
g <- ggbiplot(ir.pca, obs.scale = 1, var.scale = 1, groups = class, ellipse = TRUE
, circle = FALSE)
g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal',legend.position = 'top')
print(g)
```
#Pairs Plot
```{r}
#pairs plot where observations in a class have different colour and symbol.
clPairs(X,class,lower.panel = NULL)
```
##Distance Matrix (Assesing whether is clusterable).
```{r}
#Standadising the data using cluster sim package.
X1= data.Normalization (X,type="n8",normalization="column")
#Setting Euclidean distance, stand is for standadising data.
Set to false as data has been standardised.
dist.eucl= get_dist(X1,method = "manhattan", stand=FALSE)
#Visualising the distance matrix. Red indicates high similarity and blue low similarity.
fviz_dist(dist.eucl,show_labels=FALSE,order = TRUE)+
labs(title = "Distance Matrix: Euclidean Distance")
#lower value means the items are similar. Red squares indicate clusters.
```
#Hierarchical Clustering
```{r}
#Standadising the data using cluster sim package.
X2= data.Normalization (X,type="n8",normalization="column")
# Dissimilarity matrix. Distance Method
d <- dist(X2, method = "manhattan")
# Hierarchical clustering using different Linkages
hc1 <- hclust(d, method = "ward.D" )
plot(hc1, labels=FALSE) # display dendogram
groups <- cutree(hc1, k=3) # cut tree into k clusters
# draw dendogram with red borders around the k clusters
rect.hclust(hc1, k=3, border="red")
classT=as.vector(mix1$class)
adjustedRand(classT, groups)
```
#Determining the number of clusters to use for K-means
```{r message=FALSE, warning=FALSE}
#Standadising the data using cluster sim package.
X3= data.Normalization (X,type="n8",normalization="column")
## Elbow Method Using FactoExtra
fviz_nbclust(X3, FUNcluster = kmeans, method = "wss")
## Silhouette
# Silhouette method in FactoExtra
fviz_nbclust(X3, kmeans, method = "silhouette") + theme_classic()
## Gap Statistic
```
```

```
### GAP STAT Using Facto Extra
fviz_nbclust(X3, kmeans, nstart = 25, method = "gap_stat", nboot = 500) +
#nboot is # of bootstrap sample - must be included
labs(subtitle = "Gap Statistic Method")
#NbClust uses 30 different indices for choosing the best number of clusters.
nb <- NbClust(X3, distance = "euclidean", min.nc = 2,
max.nc = 10, method = "kmeans")
fviz_nbclust(nb)
```
#K-means analysis with 3 clusters
```{r}
X4= data.Normalization (X,type="n8",normalization="column")
#Fitting a 3 cluster solution
C3=Kmeans(X4,3, iter.max = 500, nstart = 200,method = "euclidean")

#Rand index for 3 cluster solution.
# Compute cluster stats
label1 <- as.numeric(A$id)
clust_stats3 <- cluster.stats(d=dist(X4),label1, C3$cluster)
# Corrected Rand index
adjustedRand(classT, C3$cluster)
#Variation of information
clust_stats3$vi
```
#Visualising K-Means Results.

3 Cluster solution.
```{r}
fviz_cluster(C3, data = X4,
ellipse.type = "norm", # Concentration ellipse
star.plot = FALSE, # Add segments from centroids to items
repel = TRUE, # Avoid label overplotting (slow)
ggtheme = theme_minimal(),
geom = "point", show.clust.cent="TRUE")
```
#Finite Mixture Modelling.
```{r}
#Using EM to calculate BIC values for different clusters and parameterizations.
BIC<-mclustBIC(X)
#Plotting the BIC values.
plot(BIC)
#Summary of the BIC values giving the top 3 BIC values.
summary(BIC)
```
```{r}
#Optimal model according to the BIC values
mod1<-Mclust(X,x=BIC)
#Summary of the best model.
summary(mod1, parameters=TRUE)
```
Dimension reduction for model 3 cluster model
```{r}
#To visualise the clustering structure and geometric characteristics.
dr1 <- MclustDR(mod1)
#Summary of the dimension reduction.
summary(dr1)
#Plot of the model in 2 dimensions showing the contours.
plot(dr1, what = "contour")
```

```
plot(dr1, what = "boundaries", ngrid = 200)
miscl <- classError(class, mod1$classification)$misclassified
points(dr1$dir[miscl,], pch = 1, cex = 2)
#Plots of the densities in the first and second dimension.
plot(dr1, what = "density", dimens = 1)
plot(dr1, what = "density", dimens = 2)
plot(dr1, what = "evalues")
#Pairs plot for the dimensions.
plot(dr1, what = "pairs",lower.panel = NULL)
```

##Fitting a 4 cluster solution model.

```{r}
#Summary of the 4 cluster solution models.
summary(BIC,data=X, G=4)
```
```{r}
#Optimal model according to the BIC values
mod2<-Mclust(X,x=BIC,G=4)
#Summary of the best model.
summary(mod2)
summary(mod2, parameters=TRUE)
```
```{r}
#To visualise the clustering structure and geometric characteristics.
dr2 <- MclustDR(mod2,lambda = 1)
#Summary of the dimension reduction.
summary(dr2)
#Plot of the model in 2 dimensions showing the contours.
plot(dr2, what = "contour")
plot(dr2, what = "boundaries", ngrid = 200)
miscl <- classError(class, mod1$classification)$misclassified
points(dr2$dir[miscl,], pch = 1, cex = 2)
#Plots of the densities in the first and second dimension.
plot(dr2, what = "density", dimens = 1)
plot(dr2, what = "density", dimens = 2)
plot(dr2, what = "evalues")
#Pairs plot for the dimensions.
plot(dr2, what = "pairs")
```
```{r}
#Adjusted Rand Index
adjustedRand(classT, mod1$classification)
adjustedRand(classT, mod2$classification)
```
```{r}
#Computing and displaying the BIC values for all covariance stuctures.
BIC <- mclustBIC(X)
BIC
```
```{r}
#Integrated complete data likelihood.
ICL<-mclustICL(X)
ICL
summary(ICL)
```
```{r}
```

```
#Plot of the ICL for all parameterisation.
plot(ICL)
```
```{r}
#Bootstrap sequential LRT for the number of mixture components
LRT1 <- mclustBootstrapLRT(X, modelName = "VII",  nboot = 99)
LRT2 <- mclustBootstrapLRT(X, modelName = "VEI",  nboot = 99)
LRT1
LRT2
```

Appendix H

R Code for S

SEEDS DATA

```{r message=FALSE, warning=FALSE, include=FALSE}
#Required packages.
library(cluster)
#Package for standardisation methods.
library(clusterSim)
#Package for K-means
library(stats)
library(NbClust)
library(factoextra)
library(fpc)
#K-means with different distance measures.
library(amap)
library(tidyverse, warn.conflicts = FALSE)
library(hrbrthemes)
#For simulating data and estimating cluster overlap.
library(MixSim)
#For finite mixture modelling.
library(mclust)
#For plotting PCA plot
library(devtools)
library(ggplot2)
library(ggbiplot)
library(plyr)
library(dplyr)
library(clues)
```
#Exploratory Data Analysis
```{r}
#Import data into R and covert Label to factor.
Seeds <- read.csv("C:/Users/tmurisa.ALA/Google Drive/Wits Mathematical Statistics
/Research Report Cluster Analysis/Data Sets/seeds.csv", sep=";")
#Setting the variable to numeric.
Seeds$Compactness=as.numeric(Seeds$Compactness)
#Set the variable label to a factor variable.
Seeds$Label=as.factor(Seeds$Label)
#Structure of the seeds data set.
```

```
#Removing the column of labels from the Seeds data set.
X <- (Seeds[,-8])
str(Seeds)
```
```{r}
#Rename variable for easily manipulation.
area<-Seeds$Area
per<-Seeds$Perimeter
comp<-Seeds$Compactness
lengthk<-Seeds$length_kern
widthk<-Seeds$width_kern
asyc<-Seeds$asy_coeff
kerngr<-Seeds$kern_groove
class<-Seeds$Label
```
```{r}
#Summary of all the variables.
summary(Seeds)
var(area)
var(per)
var(comp)
var(lengthk)
var(widthk)
var(asyc)
var(kerngr)
```
#Box plots

```{r}
#Box plots for all the variables.
par(mfrow=c(1,7))
boxplot(area,main="Area")
boxplot(per, main="Perimeter")
boxplot(kerngr, main="K groove")
boxplot(lengthk, main="K length")
boxplot(widthk, main="K Width")
boxplot(comp, main="Comp")
boxplot(asyc, main="Asy Coeff")
```
#Histograms
```{r}
#Histograms
par(mfrow=c(1,4))
hist(area, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.2), xlab = "Area",breaks=8)
# prob=TRUE for probabilities not counts
lines(density(area), col="red", lwd=2) # add a density estimate with defaults
hist(per, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.4), xlab = "Perimeter",breaks=8)
# prob=TRUE for probabilities not counts
lines(density(per), col="red", lwd=2) # add a density estimate with defaults
hist(kerngr, prob=TRUE, col="white", main = NULL, xlab = "Kernel groove length",breaks=8)
# prob=TRUE for probabilities not counts
lines(density(kerngr), col="red", lwd=2) # add a density estimate with defaults
hist(lengthk, prob=TRUE, col="white", main = NULL, ylim = c(0, 1.2),
xlab = "Kernel length",breaks=8)# prob=TRUE for probabilities not counts
lines(density(lengthk), col="red", lwd=2) # add a density estimate with defaults
par(mfrow=c(1,3))
hist(widthk, prob=TRUE, col="white", main = NULL, ylim = c(0, 1.0),
xlab = "Kernel width",breaks=8)# prob=TRUE for probabilities not counts
```

```
lines(density(widthk), col="red", lwd=2) # add a density estimate with defaults
hist(asyc, prob=TRUE, col="white", main = NULL, ylim = c(0, 0.25),
xlab="Asymmetry coefficient",breaks=8)# prob=TRUE for probabilities not counts
lines(density(asyc), col="red", lwd=2) # add a density estimate with defaults
hist(comp, prob=TRUE, col="white", main = NULL, xlab = "Compactness",breaks=8)
# prob=TRUE for probabilities not counts
lines(density(comp), col="red", lwd=2) # add a density estimate with defaults
```
```
#Principal Components Analysis
```{r}
# apply PCA - scale. = TRUE is highly
# advisable, but default is FALSE.
ir.pca <- prcomp(X, center = TRUE, scale. = TRUE)
```
```{r}
options(digits=2)
# print method
print(ir.pca)
```
```{r}
# plot method
plot(ir.pca, type = "l",main=NULL)
```
```{r}
options(digits=2)
# summary method
summary(ir.pca)
```
```{r}
g <- ggbiplot(ir.pca, obs.scale = 1, var.scale = 1, groups = class, ellipse = TRUE, circle = FALSE)
g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal',legend.position = 'top')
print(g)
```
#Pairs Plot
```{r}
#pairs plot where observations in a class have different colour and symbol.
clPairs(X,Seeds$Label,lower.panel = NULL)
```
##Distance Matrix (Assesing whether is clusterable).
```{r}
#Standadising the data using cluster sim package.
X1= data.Normalization (X,type="n1",normalization="column")
#Setting Euclidean distance, stand is for standadising data.
#Set to false as data has been standardised.
dist.eucl= get_dist(X1,method = "euclidean", stand=FALSE)
#Visualising the distance matrix. Red indicates high similarity and blue low similarity.
fviz_dist(dist.eucl,show_labels=FALSE,order = TRUE)+ labs(title = "Distance Matrix")
#lower value means the items are similar. Red squares indicate clusters.
```
#Hierarchical Clustering
```{r}
#Standadising the data using cluster sim package.
X2= data.Normalization (X,type="n1",normalization="column")

# Dissimilarity matrix. Distance Method
d <- dist(X2, method = "euclidean")
```
```

```
# Hierarchical clustering using different Linkages
hc1 <- hclust(d, method = "ward.D" )

plot(hc1, labels=FALSE) # display dendogram
groups <- cutree(hc1, k=3) # cut tree into k clusters
# draw dendogram with red borders around the k clusters
rect.hclust(hc1, k=3, border="red")

classT=as.vector(Seeds$Label)
adjustedRand(classT, groups)
```
```
#Determining the number of clusters to use for K-means
```{r message=FALSE, warning=FALSE}
#Standadising the data using cluster sim package.
X3= data.Normalization (X,type="n1",normalization="column")

## Elbow Method Using FactoExtra
fviz_nbclust(X3, FUNcluster = kmeans, method = "wss")

## Silhouette

# Silhouette method in FactoExtra
fviz_nbclust(X3, kmeans, method = "silhouette") + theme_classic()

## Gap Statistic

### GAP STAT Using Facto Extra
fviz_nbclust(X3, kmeans, nstart = 25, method = "gap_stat", nboot = 500) +
#nboot is # of bootstrap sample - must be included
labs(subtitle = "Gap Statistic Method")

#NbClust uses 30 different indices for choosing the best number of clusters.
nb <- NbClust(X3, distance = "euclidean", min.nc = 2,
max.nc = 10, method = "kmeans")
fviz_nbclust(nb)
```
```
#K-means analysis with 2 or 3 clusters
```{r}
X4= data.Normalization (X,type="n1",normalization="column")
#Fitting a 2 cluster solution.
C2=Kmeans(X4,2, iter.max = 500, nstart = 200,method = "euclidean")
#Fitting a 3 cluster solution
C3=Kmeans(X4,3, iter.max = 500, nstart = 200,method = "euclidean")

#Rand index for 2 cluster solution.
# Compute cluster stats
label1 <- as.numeric(Seeds$Label)
clust_stats2 <- cluster.stats(d=dist(X4),label1, C2$cluster)
# Corrected Rand index
adjustedRand(classT, C2$cluster)
#Variation of information
clust_stats2$vi

#Rand index for 3 cluster solution.
# Compute cluster stats
clust_stats3 <- cluster.stats(d=dist(X4),label1, C3$cluster)
# Corrected Rand index
adjustedRand(classT, C3$cluster)
```

```
#Variation of information
clust_stats3$vi
```
```
#Visualising K-Means Results.
## 2 Cluster solution.
```{r}
fviz_cluster(C2, data = X4,
ellipse.type = "norm", # Concentration ellipse
star.plot = FALSE, # Add segments from centroids to items
repel = TRUE, # Avoid label overplotting (slow)
ggtheme = theme_minimal()
)
```
3 Cluster solution.
```{r}
fviz_cluster(C3, data = X4,
ellipse.type = "norm", # Concentration ellipse
star.plot = FALSE, # Add segments from centroids to items
repel = TRUE, # Avoid label overplotting (slow)
ggtheme = theme_minimal()
)
```
#Finite Mixture Modelling.
Estimating cluster overlap
```{r}
#Calculate overlap of the seeds data set using MixSim
p <- ncol(Seeds) - 1
id <- as.integer(Seeds[, 8])
K <- max(id)
Pi <- prop.table(tabulate(id))
Mu <- t(sapply(1:K, function(k){ colMeans(Seeds[id == k, -8]) }))
S <- sapply(1:K, function(k){ var(Seeds[id == k, -8]) })
dim(S) <- c(p, p, K)
overlap(Pi = Pi, Mu = Mu, S = S)
pdplot(Pi = Pi, Mu = Mu, S = S)
```

```{r}
#Using EM to calculate BIC values for different clusters and parameterizations.
BIC<-mclustBIC(X)
#Plotting the BIC values.
plot(BIC)
#Summary of the BIC values giving the top 3 BIC values.
summary(BIC)
```
#The 4 cluster solution.
```{r}
#Optimal model according to the BIC values, 4 cluster solution.
mod1<-Mclust(X,x=BIC)
#Summary of the best model.
summary(mod1)
summary(mod1, parameters=TRUE)
```

#Plots for 4 cluster solution.
```{r}
#To visualise the clustering structure and geometric characteristics.
dr1 <- MclustDR(mod1)
```
```

```
#Summary of the dimension reduction.
summary(dr1)
#Plot of the model in 2 dimensions showing the contours.
plot(dr1, what = "contour")
#Plot of the uncertainty boundaries. Circles around missclassified points.
plot(dr1, what = "boundaries", ngrid = 200)
miscl <- classError(class, mod1$classification)$misclassified
points(dr1$dir[miscl,], pch = 1, cex = 2)
#Plots of the densities in the first and second dimension.
plot(dr1, what = "density", dimens = 1)
plot(dr1, what = "density", dimens = 2)
plot(dr1, what = "evalues")
#Pairs plot for the dimensions.
plot(dr1, what = "pairs",lower.panel = NULL)
```
```
Fitting a 3 cluster solution model.
```{r}
#Summary of the 3 cluster solution models.
summary(BIC,data=X, G=3)
```
```{r}
#Optimal model according to the BIC values
mod2<-Mclust(X,x=BIC,G=3)
#Summary of the best model.
summary(mod2)
summary(mod2, parameters=TRUE)
```
```
Plots for 3 cluster model.
```{r}
#To visualise the clustering structure and geometric characteristics.
dr2 <- MclustDR(mod2,lambda = 1)
#Summary of the dimension reduction.
summary(dr2)
#Plot of the model in 2 dimensions showing the contours.
plot(dr2, what = "contour")
#Plot of the uncertainty boundaries. Circles around missclassified points.
plot(dr2, what = "boundaries", ngrid = 200)
miscl <- classError(class, mod1$classification)$misclassified
points(dr2$dir[miscl,], pch = 1, cex = 2)
#Plots of the densities in the first and second dimension.
plot(dr2, what = "density", dimens = 1)
plot(dr2, what = "density", dimens = 2)
plot(dr2, what = "evalues")
#Pairs plot for the dimensions.
plot(dr2, what = "pairs")
```
```
```{r}
#Adjusted Rand Index
adjustedRand(classT, mod1$classification)
adjustedRand(classT, mod2$classification)
```
```
```{r}
#Computing and displaying the BIC values for all covariance stuctures.
BIC <- mclustBIC(X)
BIC
```
```
```{r}
#Integrated complete data likelihood.
```
```

```
The higher the value the better the model.
ICL<-mclustICL(X)
ICL
summary(ICL)
```
```{r}
#Plot of the ICL for all parameterisation.
plot(ICL)
```
```{r}
#Bootstrap sequential LRT for the number of mixture components
LRT1 <- mclustBootstrapLRT(X, modelName = "VEV",  nboot = 99)
LRT1
```