

الجمهورية الجزائرية الديمقراطية الشعبية

Ministry of Higher Education and Scientific Research

UNIVERSITY OF SAAD DAHLAB BLIDA
Faculty of Sciences

Department of Mathematics



MASTER's THESIS
In mathematics

Option: Stochastic and Statistical Models

Statistical clustering methods, Application to financial data

Realised by
Hachemane Manel

Jury Members :

President :	Mr. OMAR TAMI	M.C.A	U.S.D.BLIDA 1
Examiner :	Mr. MOUHAMED BOUKHARI	M.A.A	U.S.D.BLIDA 1
Supervisor :	Mr. REDOUANE FRIHI	M.C.B	U.S.D.BLIDA 1

4 July 2024

Dedication

To my dear father and mother, Whom I can never thank enough, for everything

To my beloved siblings: MOUHAMED, OSSEMA

To the one who accompanied me in my career: SALIM

To the person who looks like me, my sister KARIMA

To my closest friends: RAYANE, NESSRINE, HALA, AICHA

To my cousins : AMINA, MERIEM, AMANI, ROUFAYDA, SARAH, SOIMIA

Acknowledgments

Gratitude,

First of all, Elhamdoulillah, gratitude to Allah who granted me the ability to stand here with my dissertation.

I would like to express my deepest gratitude to the following individuals and administration who have played a significant role in the completion of this dissertation:

First and foremost, I am profoundly grateful to my supervisor, Dr FRIHI REDOUANE, whose unwavering guidance, invaluable insights, patience, and constant support have been instrumental in shaping this research. Your expertise and dedication have truly inspired me throughout this process.

I would like to extend my sincere appreciation to the members of my Jury members, the director Dr. TAMI OMAR and the examiner, the teacher, MOHAMED BOUKHARI, for their constructive feedback, thought-provoking discussions, and suggestions that greatly enhanced the quality of this work. Your collective expertise and commitment to academic excellence have been invaluable.

I am grateful to the staff and faculty of Science for providing a conducive academic environment and access to resources necessary for conducting this research. Their support, administrative assistance, and library services have been immensely helpful.

My deepest thanks go to my dear parents, brother, family and my precious friends who have encouraged and supported me throughout my studies. And finally, thanks to all the people who have contributed in any way to the elaboration of this work.

ملخص

المذكورة تركز على طرق التجميع الإحصائية في تحليل البيانات المالية، حيث تستفيد من خوارزميات مثل k-means و التجميع الهرمي ونماذج المزيج الغاوسي لتجميع نقاط البيانات بناءً على أوجه التشابه الإحصائية.

تؤدي هذه التقنيات دورًا أساسيًا في العديد من التطبيقات المالية، بما في ذلك تحسين المحفظة وتقسيم السوق وإدارة المخاطر واكتشاف الشذوذ. على سبيل المثال، في تحسين المحفظة الاستثمارية، يساعد التجميع في تنويع الاستثمارات من خلال تجميع الأصول ذات ملفات تعريف المخاطر والعائدات المتشابهة، وبالتالي تقليل التقلبات الإجمالية للمحفظة

مما يساعد في استراتيجيات التحوط وتخفيف مخاطر السوق. علاوةً على ذلك، تساعد أساليب التجميع في الكشف عن الحالات الشاذة في المعاملات المالية مما يعزز أليات الكشف عن الاحتيال ومنع الأخطاء، وتحسين عمليات اتخاذ القرار، وتحسين الاستراتيجيات المالية من خلال التحليل المنهجي لأنماط البيانات والعلاقات المعقدة.

Abstract

The note focuses on statistical clustering methods in financial data analysis, leveraging algorithms such as k-means, hierarchical clustering, and Gaussian mixture models to cluster data points based on statistical similarities.

These techniques play an essential role in many financial applications, including portfolio optimization, market segmentation, risk management, and anomaly detection. For example, in portfolio optimization, clustering helps diversify investments by grouping assets with similar risk and return profiles, thereby reducing overall portfolio volatility

which aids in hedging strategies and mitigates market risk. Moreover, clustering methods help in detecting anomalies in financial transactions, enhancing fraud detection and error prevention, improving decision-making processes, and optimizing financial strategies by systematically analyzing complex data patterns and relationships.

Résumé

Cette note se concentre sur les méthodes de regroupement statistique dans l'analyse des données financières, en s'appuyant sur des algorithmes tels que les k-moyennes, le regroupement hiérarchique et les modèles de mélange gaussien pour regrouper les points de données sur la base de similitudes statistiques.

Ces techniques jouent un rôle essentiel dans de nombreuses applications financières, notamment l'optimisation de portefeuille, la segmentation du marché, la gestion des risques et la détection des anomalies. Par exemple, dans le cadre de l'optimisation de portefeuille, le regroupement permet de diversifier les investissements en regroupant les actifs présentant des profils de risque et de rendement similaires, réduisant ainsi la volatilité globale du portefeuille.

ce qui facilite les stratégies de couverture et atténue le risque de marché. En outre, les méthodes de regroupement permettent de détecter les anomalies dans les transactions financières, d'améliorer la détection des fraudes et la prévention des erreurs, d'améliorer les processus de prise de décision et d'optimiser les stratégies financières en analysant systématiquement les modèles et les relations de données complexes.

Table of contents

- List of contents 1
- List of Figures 1
- Chapitre 1. An introduction to classification and clustering 1**
 - 1.1 What is Classification? 1
 - 1.1.1 Supervised classification 1
 - 1.1.2 Unsupervised classification 1
 - 1.2 What is a cluster? 2
 - 1.2.1 Example of the use of clustering 2
 - 1.2.2 Numerical methods of classification – Cluster analysis 4
 - 1.3 Statistic Clustering Method 5
 - 1.4 Other Considerations in Clustering 6
 - 1.5 Criteria for correct classification 6
 - 1.5.1 Validity 7
 - 1.5.2 Interpretability 7
 - 1.5.3 Stability 7
 - 1.5.4 Other criteria 7
 - 1.6 Reasons for classifying 7
 - 1.7 Statistic clustering method 8
- Chapitre 2. Detecting clusters with univariate and bivariate plots of Data 12**
 - 2.1 Introduction 12

2.2	Histograms	13
2.3	Density Estimation	13
2.4	Scatterplot matrices	15
2.5	Hierarchical clustering	16
2.5.1	Hierarchy types	16
2.5.2	Distance Functions	16
2.5.3	Agglomerative Clustering	18
2.5.4	Divisive Clustering	20
2.5.5	Splitting Criteria	20
2.6	Quantifying output value an axiomatic approach	20
2.6.1	Distance, Similarity, and their use :	22
2.7	K -means clustering	22
2.7.1	Objective Function	23
2.7.2	An Overview of K -Means Clustering	23
2.7.3	K -means clustering algorithm	24
2.7.4	When will K -means cluster analysis fail	26
2.7.5	K -means clustering advantages and disadvantages	26
2.7.6	Difficult questions posed by K -means	28
2.8	Fuzzy C -Means Clustering	30
2.8.1	Objective Function	30
2.8.2	Membership and Centroid Update Equations	30
2.8.3	Euclidean-Distance-Based Fuzzy C -Means Clustering	31
2.8.4	Fuzzy C -Means Algorithm (FCM)	32
2.9	Gaussian Mixture Models (GMM)	33
2.9.1	Model Definition	33
2.9.2	Mixture of K -Gaussian distributions: (Multi-modal distribution)	34
2.9.3	Gaussian Mixture Model (GMM)	34
2.9.4	Expectation-Maximization (EM) Algorithm	35

2.9.5	Density-Based Spatial Clustering of Applications with Noise (BSCAN)	36
2.9.6	Why DBSCAN?	37
2.9.7	DBSCAN Clustering Algorithm	37
2.9.8	Distance Calculation in DBSCAN Clustering	38
Chapitre 3. Financial Application		41
3.1	Introduction	41
3.2	Simulation	41
3.2.1	Univariate Data	42
3.3	Bivariate Data	44
3.3.1	<i>K</i> -means method	44
3.4	Clustering real data application:	46
3.4.1	Cluster Analysis	46
3.5	Applying Clustering Methods to Cluster Cryptocurrencies	47
3.5.1	The Basics about Cryptocurrency	47
3.5.2	What is cryptocurrency?	47
3.5.3	What are the risks to using cryptocurrency?	47
3.5.4	Key Features of Cryptocurrencies:	48
3.5.5	Examples of Cryptocurrencies:	49
3.5.6	Clustering cryptocurrencies	49
3.5.7	Applying Clustering Methods to Cluster Development Indices	55
3.5.8	Background and Motivation	55
3.6	Importance of Clustering Development Indices	55
3.6.1	Development Indices	56
3.6.2	Objectives of the Study	58
3.6.3	Clustering Index development	58
Bibliography		64
Appendices		65

List of figures

1.1	Examples of supervised learning	2
1.2	A version of the Periodic Chart from Sisler Macri (1963)	4
1.3	K -means clustering	8
1.4	Hierarchical methods	9
1.5	Fuzzy C -means clustering	10
1.6	Gaussian Mixture Models	10
1.7	DBSCAN Clustering	11
2.1	Histogram myelinated fiber size in the lumbosacral ventral root of a kitten of a given age	13
2.2	Coppejans and Sieg (2005) estimate showing individual kernels	14
2.3	Scatterplot of fMRI data enhanced with estimated bivariate density (Li et al. (2011))	15
2.4	Scatterplot Matrix	17
2.5	Dendrogram showing single linkage clustering of simulated data set	17
2.6	Hierarchical clustering dendrogram	18
2.7	Agglomerative approach	19
2.8	Objective Functions and Algorithms	21
2.9	K -means clustering	24
2.10	Top left, the data is shown. Top center, each observation is randomly assigned to a cluster. Top right, the cluster centroids are computed	25

2.11	cluster	27
2.12	Illustration of DBSCAN Clustering Concepts	40
3.1	Histogram Clustering for two data sets	42
3.2	Histogram Clustering for three data sets	42
3.3	Density Clustering	43
3.4	Fuzzy C-means Clustering for Univariate sample	43
3.5	K -means methods for three clusters	44
3.6	K -means methods for three cluster	44
3.7	Gaussian Mixture method	45
3.8	DBSCAN methods	46
3.9	Clustering for three clusters	51
3.10	DBSCAN Clustering	51
3.11	K -means	52
3.12	Data of Price	53
3.13	Data of Volume	53
3.14	Data of Market-Cap	53
3.15	Histogramm of data Price	54
3.16	Histogramm of Volume	54
3.17	Histogramm of data Market-Cap	54
3.18	k -means cluster of the index development for 48 countries with Two clusters.	58
3.19	k -means cluster of the index development for 48 countries with Three clusters.	60

Abbreviations

- **GMM**:Gaussian Mixture Models
- **DBSCAN**:Density-based Spatial Clustering of Applications with Noise
- **BTC** :Bitcoin
- **ETH**:Ethereum
- **XRP**:Ripple 'ripple credits',the X prefix for non-national currencies in the ISO 4217 standard
- **Ltc**:Litecoin
- **ADA**:Cardano
- **FCM**:Fuzzy C-Means Algorithm

General Introduction

Clustering is an unsupervised machine learning technique used to group similar data points into clusters based on their characteristics. It is widely used in fields such as data mining, pattern recognition and image analysis to identify natural groupings in data. Here's an introduction to some of the most common clustering methods.

In marketing, clustering helps determine client categories for targeted campaigns. In scientific research and text analysis, it groups data and documents to identify trends and themes.

In image and video processing, clustering aids in segmenting images for object recognition and clustering similar videos for search and recommendation purposes.

E-commerce uses clustering to suggest products based on user behavior. Common clustering techniques include k-means, which is efficient for spherical clusters but requires specifying the number of clusters; hierarchical clustering, which creates a hierarchy of clusters without needing to specify the number in advance but is computationally expensive; and DBSCAN, a density-based method. Clustering is versatile and valuable for extracting information and improving decision-making, with the choice of technique depending on the data and analysis objectives.

The thesis is structured as follows:

- **Chapter 1:** An introduction to classification and clustering. A detailed review of existing clustering methods, including traditional and fuzzy approaches, and their applications.
- **Chapter 2:** Detecting clusters with univariate and bivariate plots of data. A

description of the datasets, evaluation metrics, and experimental setup used in the study.

- **Chapter 3:** Financial Application The implementation details of the clustering algorithms, along with the results of their application to financial datasets.
- **Chapter 4:** Conclusion and Future Work. A summary of the findings, conclusions drawn from the study, and potential directions for future research.

Chapitre 1

An introduction to classification and clustering

1.1 What is Classification?

The problem of classification involves classifying objects into classes when there is already been classified by experts or by other means.

Classification aims to determine which class new objects belong to and develops automatic algorithms for doing so. Typically this involves assigning new observations to the class whose objects they most closely resemble in some sense.

1.1.1 Supervised classification

In supervised classification, the algorithm is trained using a labeled dataset, where each training example is paired with a corresponding label.

1.1.2 Unsupervised classification

In unsupervised learning, the dataset does not have labeled outputs. The goal is to identify inherent structures in the data, such as grouping similar instances together.

This is often referred to as clustering rather than classification, but it serves a similar purpose of categorizing data.

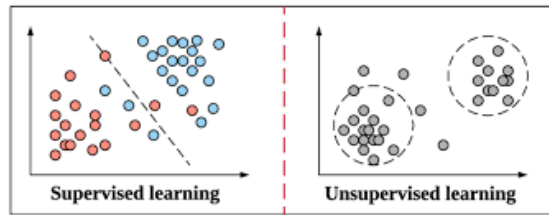


Figure 1.1 – Examples of supervised learning

1.2 What is a cluster?

Clustering this is unsupervised learning it is an exploratory data analysis technique used to summarize information about the data or determine connections between points.

The goal is to discover the inherent structure in the data without predefined labels. K -means , hierarchical clustering and DBSCAN are popular clustering algorithms. An application could be grouping customers based on purchasing behavior to target marketing strategies effectively in data.

1.2.1 Example of the use of clustering

The general problem which cluster analysis addresses appears in many disciplines medicine, bank, marketing, etc.

Here we describe briefly the number of applications of cluster analysis reported in some of this discipline.

1. **Bank:** In banking, clustering is a popular unsupervised learning technique that divides clients into discrete groups according to their financial actions, behavior, or traits. This can assist banks in managing risks, identifying new business prospects, enhancing customer service, and customizing their marketing efforts.

2. **Market research:** Dividing customers into homogeneous groups is one of the basic strategies of marketing. A market researcher may, for example, ask how to group consumers who seek similar benefits from a product so he or she can communicate with them better.

Or a market analyst may be interested in grouping financial characteristics of companies so as to be able to relate them to their stock market performance.

An early specific example of the use of cluster analysis in market research Green and Krieger (1995).

3. **Astronomy:** Astronomers want to know how many distinct classes of, for example, stars there are on the basis of some statistical criterion.

4. **Psychiatry:** Cluster analysis has also been used to find a classification of individuals who attempt suicide, which might form the basis for studies into the causes and treatment of the problem Rassaby Paykel et al. (1980), for example, studied 236 suicide attempters presenting at the main emergency service of a city in the USA.

5. **Archaeology:** In archaeology, the classification of artefacts can help in uncovering their different uses, the periods they were in use and which populations they were used by.

Similarly, the study of fossilized material can help to reveal how prehistoric societies lived. An early example of the cluster analysis of artefacts Hodson (1970).

6. **Physics and Chemistry:** The Periodic Law belongs among the most profound achievements in the discipline, as it links four aspects of the elements: the internal structure of the atoms, their bondage into molecules, their chemical interaction properties, and their physical features.

1a																		7a	0																		
1	H																	1	2																		
2a																		3a	4a	5a	6a	H	He														
3	Li	4	Be																	5	6	7	8	9	10												
11	Na	12	Mg	3b	4b	5b	6b	7b	8b		1b	2b	13	14	15	16	17	18																			
19	K	20	Ca	21	Sc	22	Ti	23	V	24	Cr	25	Mn	26	Fe	27	Co	28	Ni	29	Cu	30	Zn	31	Ga	32	Ge	33	As	34	Se	35	Br	36	Kr		
37	Rb	38	Sr	39	Y	40	Zr	41	Nb	42	Mo	43	Tc	44	Ru	45	Rh	46	Pd	47	Ag	48	Cd	49	In	50	Sn	51	Sb	52	Te	53	I	54	Xe		
55	Cs	56	Ba	57-71	Hf	72	Ta	73	W	74	Re	75	Os	76	Ir	77	Pt	78	Au	79	Hg	80	Tl	81	Pb	82	Bi	83	Po	84	At	85	Rn				
87	Fr	88	Ra	89-103																																	

Figure 1.2 – A version of the Periodic Chart from Sisler Macri (1963)

1.2.2 Numerical methods of classification – Cluster analysis

Numerical techniques for deriving classifications originated largely in the natural sciences such as biology and zoology in an effort to rid taxonomy of its traditionally subjective nature. The aim was to provide objective and stable classifications.

Objective in the sense that the analysis of the same set of organisms by the same sequence of numerical methods produces the same classification; Stable in that the classification remains the same under a wide variety of additions of organisms or of new characteristics describing them .

$$I = \begin{pmatrix} X_{11} & \cdots & X_{1p} \\ X_{21} & \vdots & \cdots \\ \vdots & \vdots & \cdots \\ X_{n1} & \cdots & X_{np} \end{pmatrix}$$

The entry X_{ij} in X gives the value of the j variable on object i . Such a matrix is often termed ‘two-mode’, indicating that the rows and columns correspond to different things. The variables in X may often be a mixture of continuous, ordinal and/or categorical, and often some entries will be missing.

1.3 Statistic Clustering Method

Statistical clustering methods are techniques used to partition data into groups or clusters based on statistical criteria. These methods aim to identify homogeneous subgroups within a dataset, where instances within the same cluster are more similar to each other than to those in other clusters. Here, we briefly outline some key statistical clustering methods.

Key Methods

1. *K*-means Clustering

K-means clustering is a partitioning method that aims to partition n observations into k clusters, where each observation belongs to the cluster with the nearest mean, serving as the cluster's centroid.

Given a dataset $X = \{x_1, x_2, \dots, x_n\}$ where each $x_i \in \mathbb{R}^d$, and an initial set of centroids $\{\mu_1^{(0)}, \mu_2^{(0)}, \dots, \mu_k^{(0)}\}$, the algorithm iteratively minimizes the within-cluster sum of squares:

$$\operatorname{argmin}_{C_1, C_2, \dots, C_k} \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

2. Hierarchical Clustering

Hierarchical clustering builds a hierarchy of clusters, either agglomeratively (bottom-up) or divisively (top-down), based on the pairwise distances between data points. It does not require the number of clusters k to be specified in advance.

3. Gaussian Mixture Models (GMM)

Gaussian Mixture Models represent the distribution of data points as a mixture of several Gaussian distributions. Each Gaussian component represents a cluster, and the model parameters (mean, covariance, and mixing proportions) are estimated using the Expectation-Maximization (EM) algorithm.

4. Density-Based Clustering

Density-based clustering methods, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), identify clusters as dense regions of data points separated by regions of lower density. These methods are effective for datasets with irregular shapes and varying densities.

1.4 Other Considerations in Clustering

- Both K -means and hierarchical clustering will assign each observation to a cluster.
- However, sometimes this might not be appropriate.
- For instance, Suppose that most of the observations truly belong to a small number of (unknown) subgroups, and a small subset of the observations are quite different from each other and from all other observations.
- Then since K -means and hierarchical clustering force every observation into a cluster, the clusters found may be heavily distorted due to the presence of outliers that do not belong to any cluster.
- Mixture models are an attractive approach for accommodating the presence of such outliers.
- These amount to a soft version of K -means clustering, and are described in Hastie Steinley (2006).

1.5 Criteria for correct classification

The main objective of classification techniques is to find a partition in which the objects in a class should be similar . Objects in a class should be similar (to each other), objects in different classes should be different should be different , a good classification should fulfil different criterion :

1.5.1 Validity

It can be defined as :

- Each class in a score must be homogeneous: Objects belonging to the same class must be similar.
- Classes must be isolated from each other: Objects in different classes must be different.
- Classification must be adapted to the data: The classification must be able to explain the variation in the data.

1.5.2 Interpretability

Classes must have a substantive interpretation. Possible to give names to the classes, at best the names must correspond to the types deduced from a certain theory.

1.5.3 Stability

Classes must be stable, which means that small changes to the data and methods in the methods must not change the results.

1.5.4 Other criteria

Sometimes the size and number of classes are used as additional criteria. additional criteria: the number of classes should be as small as possible, and the should not be too small.

1.6 Reasons for classifying

Classifying, or categorizing, is a fundamental process used in various fields to simplify, organize, and make sense of complex data. It enables efficient retrieval, analysis, and understanding of information by grouping similar items together

based on predefined criteria. Classification helps in identifying patterns, making predictions, and drawing meaningful insights from data, which is crucial for decision-making and problem-solving. In business, it aids in market segmentation and customer profiling; in science, it helps in taxonomy and species identification; and in machine learning, it enhances the accuracy and performance of predictive models.

1.7 Statistic clustering method

- (a) ***K*-means clustering:** *K*-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into K clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. *K*-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using *K*-medians and *K*-medoids.

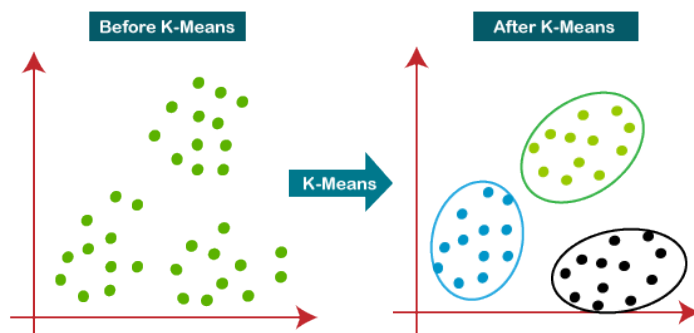


Figure 1.3 – *K*-means clustering

- (b) **Hierarchical clustering:**

In a hierarchical classification the data are not partitioned into a particular number of classes or clusters at a single step. Instead the classifica-

tion consists of a series of partitions, which may run from a single cluster containing all individuals, to n clusters each containing a single individual. Hierarchical clustering techniques may be subdivided into agglomerative methods, which proceed by a series of successive fusions of the n individuals into groups, and divisive methods, which separate the n individuals successively into finer groupings. Both types of hierarchical clustering can be viewed as attempting to find the optimal step, in some defined sense, at each stage in the progressive subdivision or synthesis of the data, and each operates on a proximity matrix of some kind (1.4).

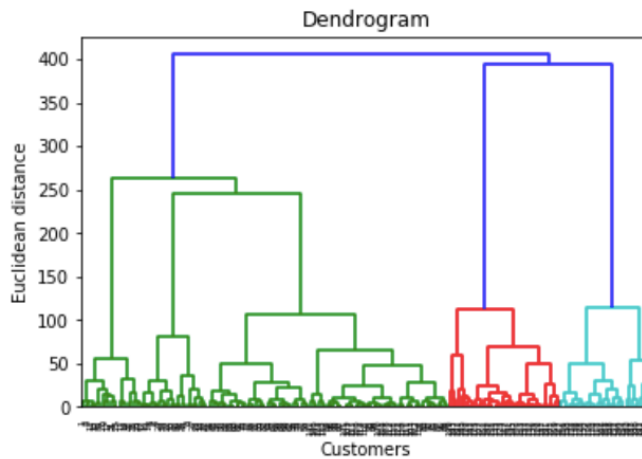


Figure 1.4 – Hierarchical methods

(c) **Fuzzy C -Means Clustering:**

Is a form of clustering in which each data point can belong to more than one cluster. Clustering or cluster analysis involves assigning data points to clusters such that items in the same cluster are as similar as possible, while items belonging to different clusters are as dissimilar as possible. Clusters are identified via similarity measures.

These similarity measures include distance, connectivity, and intensity. Different similarity measures may be chosen based on the data or the application (1.5).

(d) **Gaussian Mixture Models (GMM):**

A Gaussian mixture model (GMM) is useful for modeling data that comes

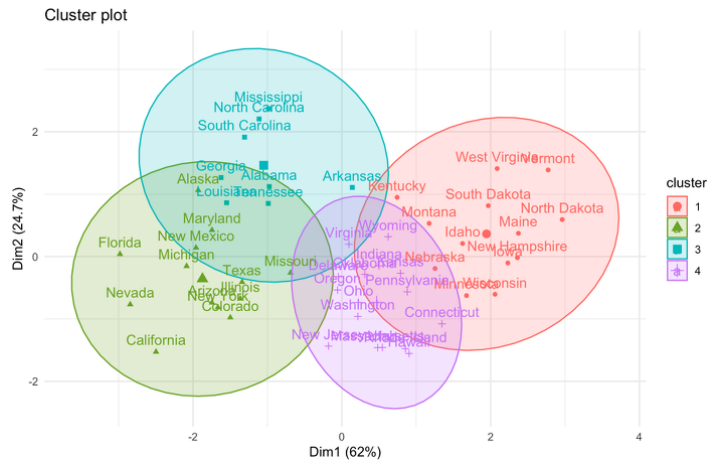


Figure 1.5 – Fuzzy C -means clustering

from one of several groups: the groups might be different from each other, but data points within the same group can be well-modeled by a Gaussian distribution (1.6).

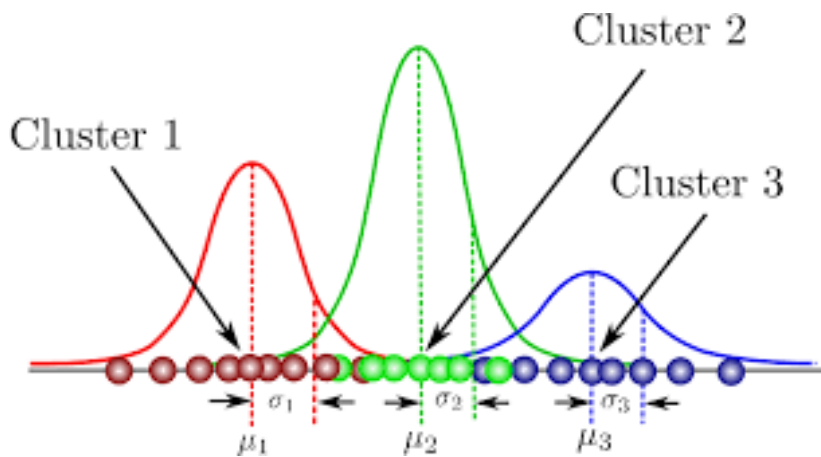


Figure 1.6 – Gaussian Mixture Models

(e) **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**

It is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Sander et al. (1998). It is a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away), DBSCAN is one of the most common, clustering

algorithms.

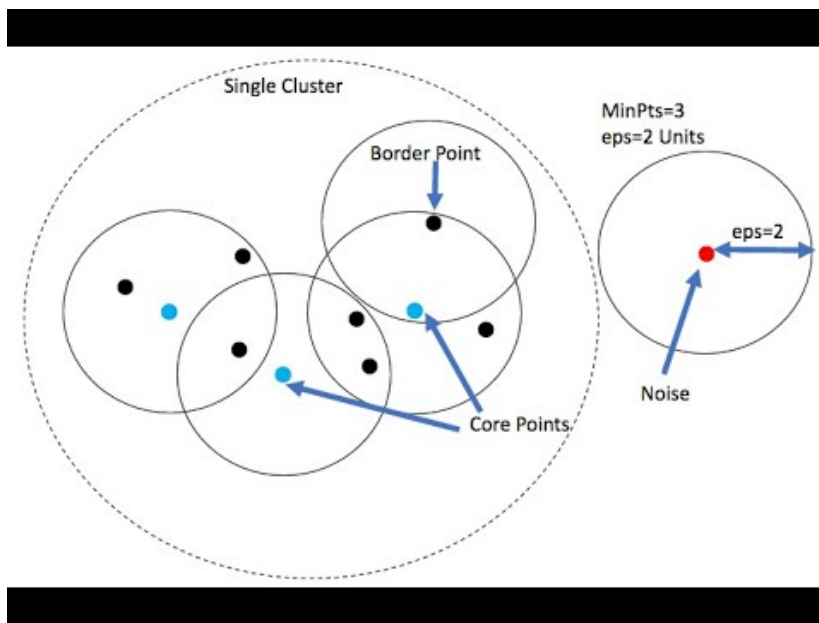


Figure 1.7 – DBSCAN Clustering

Chapitre 2

Detecting clusters with univariate and bivariate plots of Data

2.1 Introduction

It is generally argued that a unimodal distribution corresponds to a homogeneous, unclustered population and, in contrast, that the existence of several distinct modes indicates a heterogeneous, clustered population, with each mode corresponding to a cluster of observations. Although this is well known not to be universally true.

But here we shall not give details of these methods, preferring to concentrate on a rather more informal ‘eye-balling’ approach to the problem of mode detection using suitable one and two-dimensional plots of the data.

2.2 Histograms

The simple histogram is often a useful first step in finding modes in the data, especially, of course, if the data are uni-variate. Figure (2.1), for example, shows a histogram of myelinated fiber size in the lumbosacral ventral root of a kitten of a given age. The distribution is clearly bimodal, suggesting the presence of two relatively distinct groups of observations in the data.

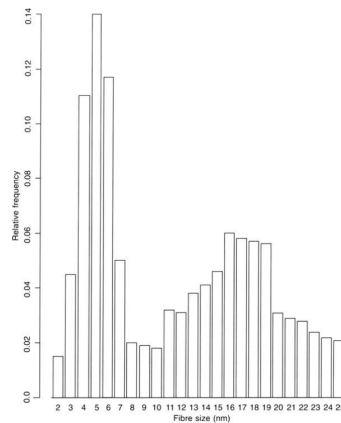


Figure 2.1 – Histogram myelinated fiber size in the lumbosacral ventral root of a kitten of a given age

2.3 Density Estimation

In statistics, probability density estimation or simply density estimation is the construction of an estimate, based on observed data, of an unobservable underlying probability density function. The unobservable density function is thought of as the density according to which a large population is distributed; the data are usually thought of as a random sample from that population. A variety of approaches to density estimation are used, including Parzen windows and a range of data clustering techniques, including vector quantization. The most basic form of density estimation is a rescaled histogram.

From the definition of a probability density, if the random variable X has

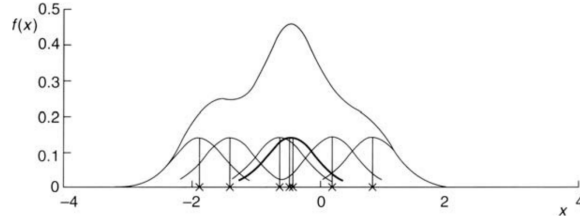


Figure 2.2 – Coppejans and Sieg (2005) estimate showing individual kernels

density f

$$f(x) = \lim_{h \rightarrow 0} \frac{1}{2h} P(x - h < X < x + h) \quad (2.1)$$

Where P is the probability distribution For any given h , a naïve estimator of $P(x - h < X < x + h)$ is the proportion of the observations X_1, X_2, \dots, X_n falling in the interval $(x - h, x + h)$; that is

$$\hat{f}(x) = \frac{1}{2hn} [\text{no. of } X_1, X_2, \dots, X_n \text{ falling in } (x - h, x + h)].$$

If we introduce a weight function W given by

$$W(x) = \begin{cases} \frac{1}{2} & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases},$$

then the naive estimator can be rewritten as

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} W\left(\frac{x - X_i}{h}\right).$$

Unfortunately, this estimator is not a continuous function and is not satisfactory for practical density estimation. It does, however, lead naturally to the kernel estimator defined by

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right),$$

where K is known as the kernel function and h as the bandwidth or smoothing

parameter. The kernel function must satisfy the condition

$$\int_{-\infty}^{\infty} K(x)dx = 1.$$

Usually, but not always, the kernel function will be a symmetric density function, for example, the normal.

2.4 Scatterplot matrices

A scatter plot matrix is table of scatter plots. Each plot is small so that many plots can be fit on a page. When you need to look at several plots, such as at the beginning of a multiple regression analysis, a scatter plot matrix is a very useful tool.

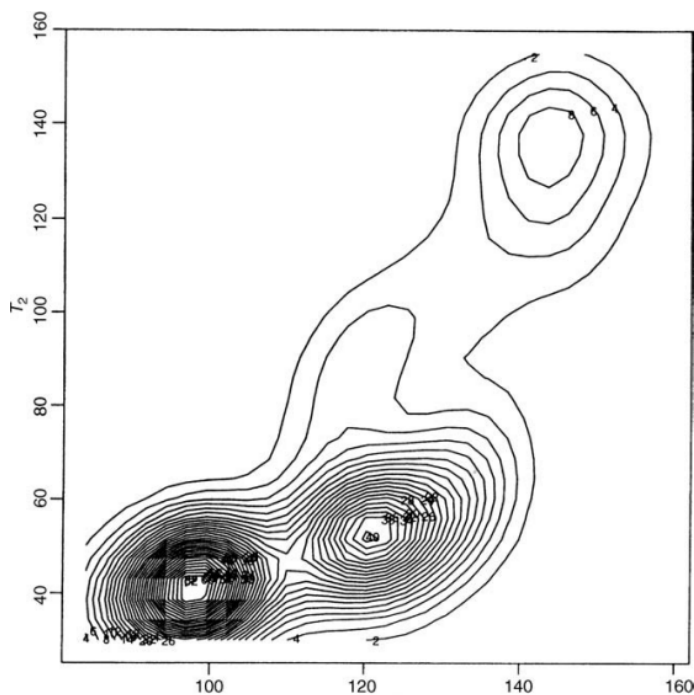


Figure 2.3 – Scatterplot of FMRI data enhanced with estimated bivariate density (Li et al. (2011))

2.5 Hierarchical clustering

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. There are two main types of hierarchical clustering: Agglomerative and Divisive.

2.5.1 Hierarchy types

- (a) Bottom Up - Agglomerative
 - (b) Starts by considering each observation as a cluster of it's own.
 - (c) Clusters are merged as we move up the hierarchy.
5. Top Down - Divisive.
- Starts by considering all observations in one cluster.
 - Clusters are divided as we move down the hierarchy.

2.5.2 Distance Functions

Certain mathematical proprieties are expected of any distance measure, or metric:

- (a) $d(x, y) = 0, \quad \forall x, y$
- (b) $d(x, y) = 0 \iff x = y$
- (c) $d(x, y) \leq d(x, z) + d(z, y), \quad \forall x, y, z$ (triangle inequality)

Euclidean distance $\sqrt{\sum_{i=1}^d (\mathbf{x}_i - \mathbf{y}_i)^2}$.

The most commonly used metric. Not that it weight all features/dimensions “equally”.

- (a) Decompose data objects into several levels of nested partitioning (tree of clusters), called a dendrogram.

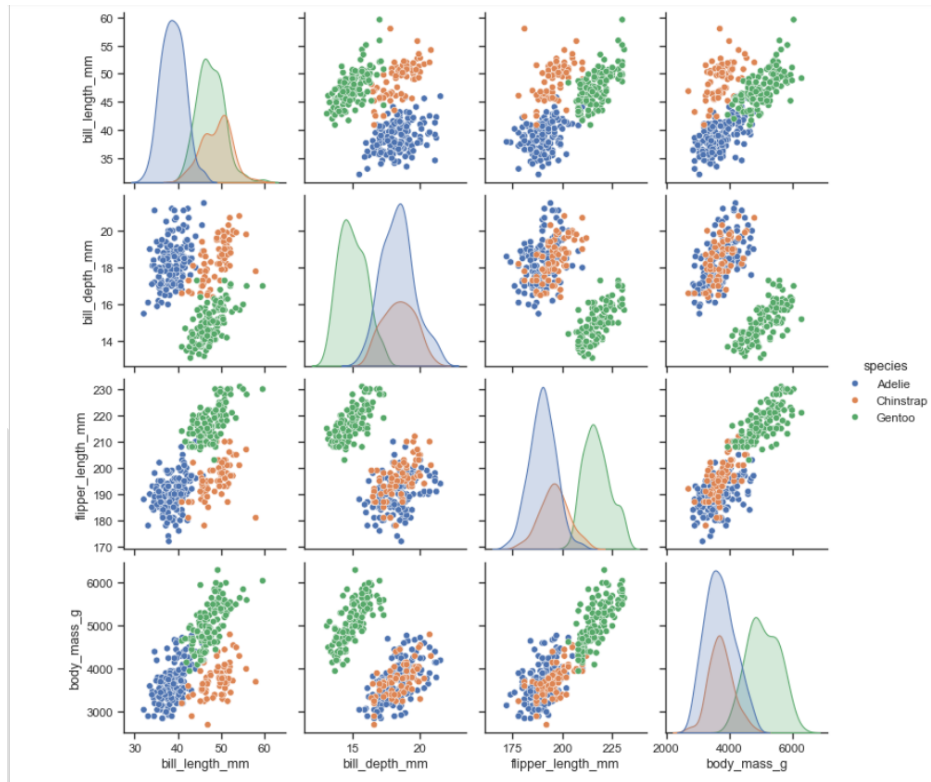


Figure 2.4 – Scatterplot Matrix

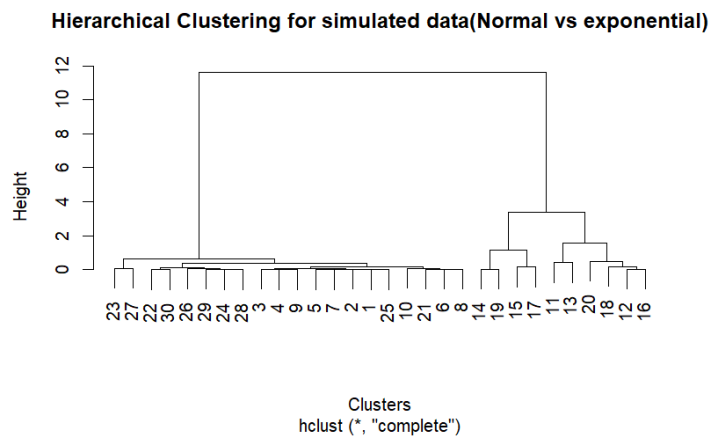


Figure 2.5 – Dendrogram showing single linkage clustering of simulated data set

- (b) A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.
- 6. A tree that shows how clusters are merged/split hierarchically
- 7. Each node on the tree is a cluster; each leaf node is a singleton cluster
- 8. A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster

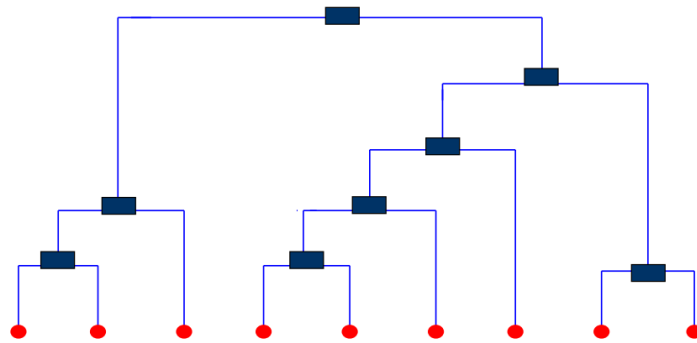


Figure 2.6 – Hierarchical clustering dendrogram

2.5.3 Agglomerative Clustering

Agglomerative clustering is a “bottom-up” approach. Here, we start with individual elements and successively merge them into clusters. The process continues until all elements are in a single cluster.

Formally, let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be the set of elements to be clustered.

The steps are as follows:

- (a) Start with n clusters, each containing one element $\{\{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \dots, \{\mathbf{x}_n\}\}$.
- (b) At each step, merge the two clusters C_i and C_j with the smallest distance $d(C_i, C_j)$, where the distance d can be defined in various ways (e.g., single-linkage, complete-linkage, average-linkage).
- (c) Repeat until there is only one cluster left.

Distance Metrics

Different methods for measuring the distance between clusters lead to different types of agglomerative clustering:

— **Single-linkage clustering:**

$$d(C_i, C_j) = \min\{d(x_p, x_q) : x_p \in C_i, x_q \in C_j\}$$

— **Complete-linkage clustering:**

$$d(C_i, C_j) = \max\{d(x_p, x_q) : x_p \in C_i, x_q \in C_j\}$$

— **Average-linkage clustering:**

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x_p \in C_i} \sum_{x_q \in C_j} d(x_p, x_q)$$

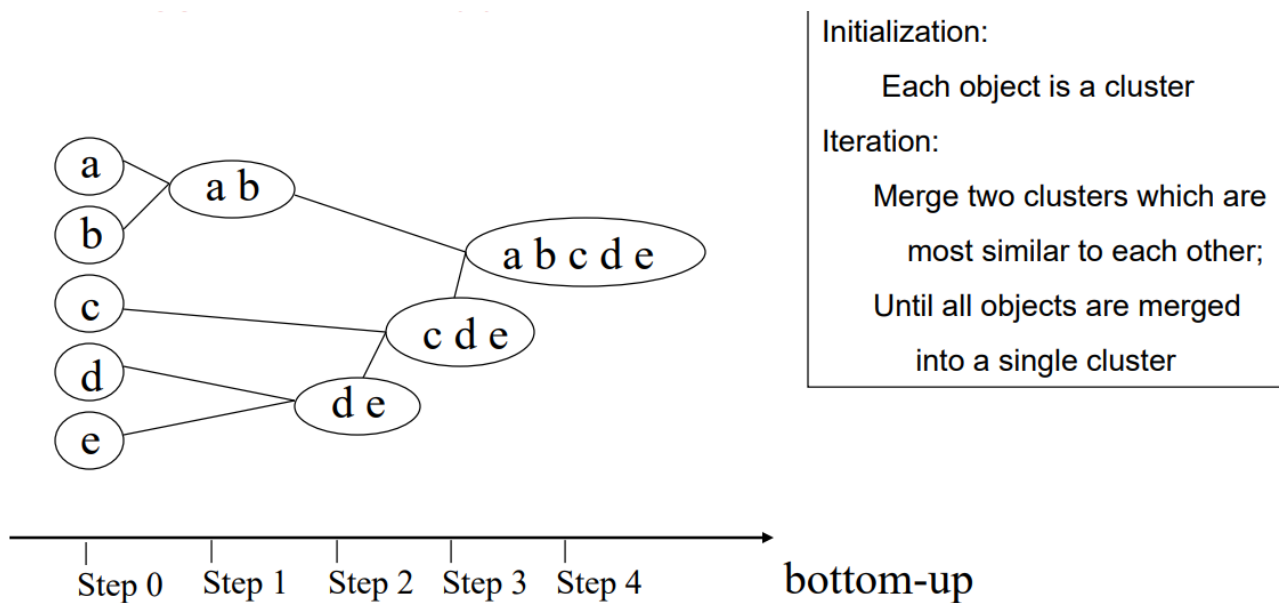


Figure 2.7 – Agglomerative approach

2.5.4 Divisive Clustering

Divisive clustering is a “top-down” approach. Here, we start with all elements in a single cluster and recursively split them into smaller clusters.

Formally, let $X = \{x_1, x_2, \dots, x_n\}$ be the set of elements to be clustered. The steps are as follows:

1. Start with one cluster containing all elements $\{X\}$.
2. At each step, split the cluster C into two clusters C_i and C_j such that a chosen criterion is optimized (e.g., minimizing the within-cluster variance).
3. Repeat until each element is in its own cluster.

2.5.5 Splitting Criteria

Different methods for splitting clusters lead to different types of divisive clustering:

- **Variance Minimization** (e.g., k-means based splitting):

$$\text{Minimize } \sum_k \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

where μ_k is the mean of cluster C_k .

- **Spectral Clustering:**

Use the eigenvectors of the Laplacian matrix of the graph representation of the data to partition the graph.

2.6 Quantifying output value an axiomatic approach

Let us focus on the *similarity* case, where *cost* and *objective* are used interchangeably.

Consider an undirected weighted graph $G = (V, E, w)$, where T is a cluster tree for the graph. We are interested in cost functions for cluster trees that measure the quality of the hierarchical clustering produced by T .

Axiom: A desirable property for the cost function is that a cluster tree T has *minimum cost* if and only if T is a *generating tree* for G .

This means the objective function can determine whether a given tree is generating and thus whether it represents the underlying ground-truth hierarchical clustering. Hence, the objective function acts as a “*guide*” for finding the correct hierarchical classification. Note that there may be multiple trees that are generating for the same graph.

For instance, if $G = (V, E, w)$ is a *clique* with all edges having the same weight, then every tree is a generating tree. In such cases, all generating trees are valid ground-truth hierarchical clusterings.

Building on the recent work of DasguptaKobren et al. (2017), we adopt an approach where we assign a cost to each internal node of the tree T to reflect the quality of the split at that node and restrict the search space for such cost functions.

For an internal node N in the clustering tree T , where A and B ($A \cup B \subset V$) are the leaves of the subtrees rooted at N 's left and right children, respectively, we define the *cost* Γ of the tree T as the sum of the costs at every internal node N :

$$\Gamma(T) = \sum_N \text{cost}(N)$$

The *cost function* $\text{cost}(N)$ for an individual node N is defined as:

$$\text{cost}(N) = \delta(N) = \left(\sum_{x \in A, y \in B} W(x, y) \right) \cdot g(|A|, |B|)$$

where $g(a, b) = a + b$.

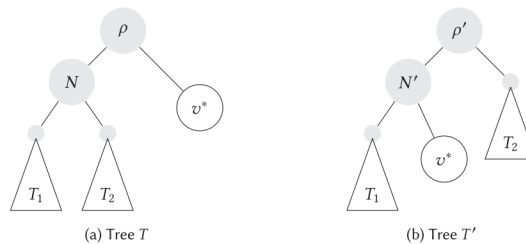


Figure 2.8 – Objective Functions and Algorithms

2.6.1 Distance, Similarity, and their use :

Before clustering, the phase of *data measurement* or *measurement of the observable* is crucial. In this phase, we need to measure the elements and their distances to determine group memberships. There are several key considerations for this process, particularly related to metrics and spatial embeddings.

To group data, we require a method to measure distances between elements. This can be done using *similarity* or *dissimilarity* measures. A distance function must satisfy the following properties:

- **Symmetry:** $d(i, j) = d(j, i)$
- **Positive Definiteness:** $d(i, j) \geq 0$ and $d(i, j) = 0$ if $i = j$
- **Triangular Inequality:** $d(i, j) \leq d(i, k) + d(k, j)$

If the triangular inequality is not satisfied, we have a *dissimilarity* measure. A *similarity* measure can be defined as:

$$s(i, j) = \max_{i, j} \{d(i, j)\} - d(i, j)$$

A traditional way to measure distances in a vector space is using the *Minkowski distance*, which is a family of metrics defined as:

$$d(i, j) = \left(\sum_{k=1}^n |x_i^k - x_j^k|^p \right)^{1/p}$$

where $p \geq 1$ is a parameter that determines the specific type of Minkowski distance, such as the Euclidean distance ($p = 2$) or the Manhattan distance ($p = 1$).

2.7 *K*-means clustering

The *K*-means algorithm developed by McQueen in 1967 (Hofmann et al. (2008)), one of the simplest unsupervised learning algorithms, called the moving center algorithm, assigns each point in a cluster. It assigns each point in a cluster to the nearest centroid. The center is the average of all points in the cluster, and its

coordinates are the arithmetic mean for each dimension separately of all the Points in the cluster, i.e. each cluster is represented by its center of gravity.

2.7.1 Objective Function

The objective of K-means is to minimize the sum of squared distances between each data point and the centroid of its assigned cluster. Mathematically, this is represented as:

$$\min_{\mathbf{C}} \sum_{k=1}^K \sum_{i \in C_k} \|\mathbf{x}_i - \mu_k\|^2$$

where:

- \mathbf{x}_i is the i -th data point.
- μ_k is the centroid of the k -th cluster.
- C_k is the set of indices of data points that belong to cluster k .
- $\|\cdot\|$ denotes the Euclidean norm.

2.7.2 An Overview of K -Means Clustering

Clustering models aim to group data into distinct “clusters” or groups. This can both serve as an interesting view in an analysis, or can serve as a feature in a supervised learning algorithm.

Consider a social setting where there are groups of people having discussions in different circles around a room. When you first look at the room, you just see a group of people. You could mentally start placing points in the center of each group of people and name that point as a unique identifier. You would then be able to refer to each group by a unique name to describe them. This is essentially what K -means clustering does with data.

In the left-hand side of the diagram above, we can see 2 distinct sets of points that are unlabeled and colored as similar data points. Fitting a K -means model to this data (right-hand side) can reveal 2 distinct groups (shown in both distinct circles and

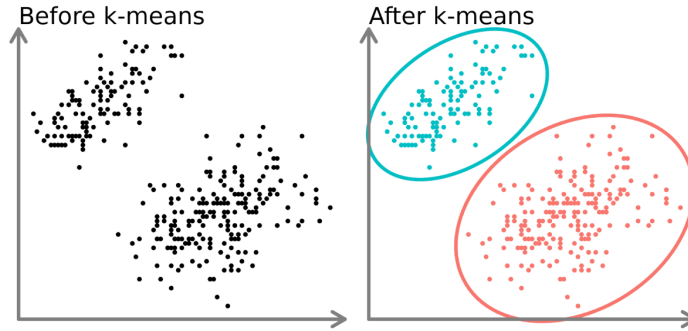


Figure 2.9 – K -means clustering

colors).

In two dimensions, it is easy for humans to split these clusters, but with more dimensions, you need to use a model.

2.7.3 K -means clustering algorithm

1. Randomly assign a number, from 1 to K , to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the clustering algorithm stops changing.
 - For each of the K clusters, compute the centroid. The k cluster centroid is the vector of the p feature averages for the observations in the cluster k .
 - Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

$$\bar{X}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} X_{ij} \quad (2.2)$$

The above algorithm is guaranteed to decrease the value of the objective function at each step.

$$\frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (X_{ij} - X_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^p (X_{ij} - \bar{X}_{kj})^2$$

Where

$$\bar{X}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij} \quad (2.3)$$

is the mean for data point (feature) j in cluster C_k

- In Step 2(a), the cluster means for each data point are the constants that minimize the sum of squared deviations.
- In Step 2(b), reallocating the data points can only improve the the within sum of squares.
- This means that as the algorithm is run, the clustering obtained will continually improve until the result no longer changes and the objective of equation 1 will never increase!
- I When the result no longer changes, we reach a local optimum.

This example is use K -means algorithm to identificate the clusters.

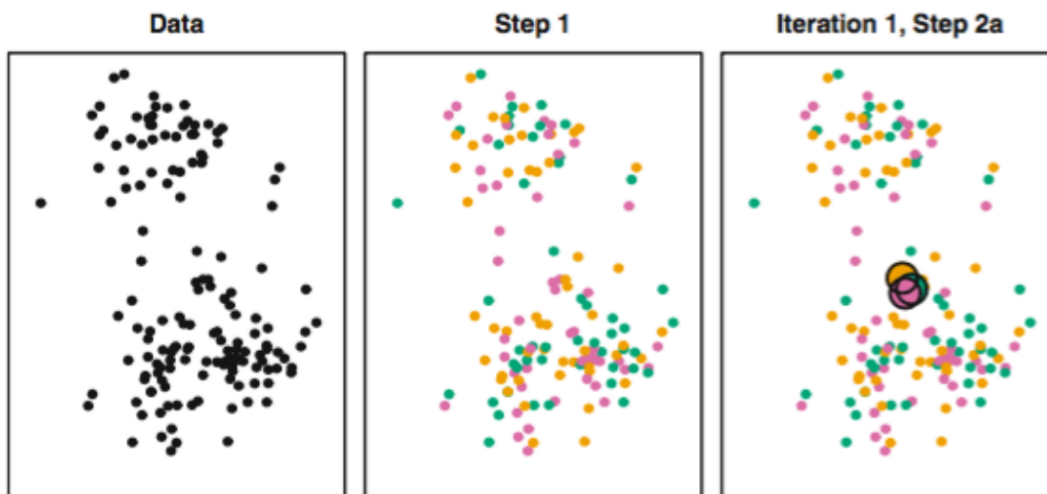


Figure 2.10 – Top left, the data is shown. Top center, each observation is randomly assigned to a cluster. Top right, the cluster centroids are computed

Example Calculation

For a simple example with three points in 2D space and two clusters:

- Data points: $\mathbf{x}_1 = (1, 2)$, $\mathbf{x}_2 = (1, 4)$, $\mathbf{x}_3 = (5, 6)$
- Initialize centroids: $\mu_1 = (1, 2)$, $\mu_2 = (5, 6)$

Assignment

- Distance from \mathbf{x}_1 to μ_1 : $\|\mathbf{x}_1 - \mu_1\|^2 = 0$
- Distance from \mathbf{x}_1 to μ_2 : $\|\mathbf{x}_1 - \mu_2\|^2 = 32$
- Thus, \mathbf{x}_1 is assigned to cluster 1.
- Similarly, compute for \mathbf{x}_2 and \mathbf{x}_3 .

Update

- Recalculate μ_1 as the mean of points in cluster 1.
- Recalculate μ_2 as the mean of points in cluster 2.

Repeat until convergence.

This process ensures that the sum of squared distances between points and their respective centroids is minimized, resulting in compact and well-separated clusters.

2.7.4 When will K -means cluster analysis fail

K -means clustering performs best on data that are spherical. Spherical data are data that group in space in close proximity to each other either.

This can be visualized in 2 or 3 dimensional space more easily.

Data that aren't spherical or should not be spherical do not work well with K -means clustering.

For example, K -means clustering would not do well on the below data as we would not be able to find distinct centroids to cluster the two circles or arcs differently, despite them clearly visually being two distinct circles and arcs that should be labeled as such.

2.7.5 K -means clustering advantages and disadvantages

K -means clustering is very simple and fast algorithm. It can efficiently deal with very large data sets. However there are some weaknesses, including:

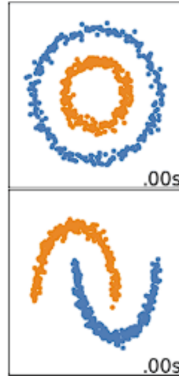


Figure 2.11 – cluster

- It assumes prior knowledge of the data and requires the analyst to choose the appropriate number of cluster (K) in advance.
- The final results obtained is sensitive to the initial random selection of cluster centers.

Why is this a problem? Because, for every different run of the algorithm on the same data set, you may choose a different set of initial centers.

This may lead to different clustering results on different runs of the algorithm.

- It's sensitive to outliers.
- If you rearrange your data, it's very possible that you'll get a different solution every time you change the ordering of your data.

Possible solutions to these weaknesses, include:

1. Solution to issue 1: Compute K -means for a range of k values, for example by varying k between 2 and 10. Then, choose the best k by comparing the clustering results obtained for the different K Values.
2. Solution to issue 2: Compute K-means algorithm several times with different initial cluster centers. The run with the lowest total within cluster sum of square is selected as the final clustering solution .
3. To avoid distortions caused by excessive outliers, It's possible to use PAM algorithm, which is less sensitive to outliers.

2.7.6 Difficult questions posed by K -means

K -means clustering is a widely used unsupervised learning algorithm for partitioning a dataset into k distinct, non-overlapping clusters. Despite its popularity and simplicity, K -means poses several challenging questions and issues that are important to consider.

1. Key Questions

Sensitivity to Initial Cluster Centers

One of the primary challenges with K -means clustering is its sensitivity to the initial selection of cluster centers. The algorithm may converge to different solutions depending on where these centers are initially placed.

Choosing the Optimal Number of Clusters (k)

Determining the optimal number of clusters (k) is another challenging aspect of K -means clustering. Various methods, such as the elbow method, silhouette score, or gap statistic, can be used, but none are foolproof and the choice often requires domain knowledge or additional validation.

Handling Non-Globular Clusters

K -means assumes that clusters are spherical or globular in shape due to its reliance on Euclidean distance. Handling clusters that are non-linear or irregularly shaped can lead to suboptimal results or misclassification.

Impact of Outliers

Outliers can significantly influence K -means clustering results, as the algorithm tries to minimize the sum of squares distances from each point to its assigned

cluster center. Outliers may pull cluster centers away from the main concentration of data points, affecting cluster boundaries.

Robustness to Noise and Scaling

K -means clustering is sensitive to noise and requires careful preprocessing, such as scaling or normalization of features, to produce meaningful clusters. Without proper preprocessing, clusters may not reflect the underlying structure of the data.

2. Mathematical Formulation

The K -means algorithm minimizes the within-cluster sum of squares:

Given a dataset $X = \{x_1, x_2, \dots, x_n\}$, where each $x_i \in \mathbb{R}^d$, the objective function is:

$$\operatorname{argmin}_{C_1, C_2, \dots, C_k} \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

where μ_j is the centroid (mean) of cluster C_j .

3. Conclusion

K -means clustering offers a straightforward approach to partitioning data, but it comes with several challenges that can affect its performance and reliability. Addressing these challenges often requires a combination of algorithmic modifications, preprocessing techniques, and domain-specific knowledge to achieve meaningful cluster assignments.

2.8 Fuzzy C -Means Clustering

Fuzzy C -Means (FCM) clustering is a method of clustering which allows one piece of data to belong to two or more clusters. This method is based on the minimization of the following objective function:

2.8.1 Objective Function

The Fuzzy C -Means objective function is defined as:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|\mathbf{x}_i - \mathbf{c}_j\|^2$$

where:

- N is the number of data points.
- C is the number of clusters.
- m is a real number greater than 1 (typically $m = 2$) that controls the fuzziness of the resulting clusters.
- u_{ij} is the degree of membership of \mathbf{x}_i in the cluster j .
- \mathbf{x}_i is the i -th data point.
- \mathbf{c}_j is the centroid of the j -th cluster.
- $\|\cdot\|$ denotes the Euclidean norm.

2.8.2 Membership and Centroid Update Equations

The membership degrees and the centroids are updated iteratively according to the following equations:

Membership Update

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|\mathbf{x}_i - \mathbf{c}_j\|}{\|\mathbf{x}_i - \mathbf{c}_k\|} \right)^{\frac{2}{m-1}}}$$

Centroid Update

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m x_i}{\sum_{i=1}^N u_{ij}^m}$$

2.8.3 Euclidean-Distance-Based Fuzzy C -Means Clustering

In the FCM algorithm, distance calculations are used to measure the similarity between data points to determine the probability that a data point belongs to a cluster. The traditional FCM algorithm is based on Euclidean distance. While Euclidean distance is optimized to detect spherical structural clusters, studies show that it does not compute accurate clustering with high dimensional data.

Let the Euclidean distance between two vectors $X = (X_1, \dots, X_d)^T$ and $Y = (Y_1, \dots, Y_d)^T$ be :

$$d(x, y) = \sqrt{\sum_{p=1}^d (x_p - y_p)^2}$$

In performing fuzzy c -means clustering, the goal is to minimize the objective function:

$$J(U, C; X, m) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ij}^2$$

subject to :

$$\sum_{i=1}^c U_{ij}, \forall j \in 1, \dots, n \quad (2.4)$$

$$\sum_{j=1}^n u_{ij} > 0, \forall j \in 1, \dots, c \quad (2.5)$$

Where:

m is the degree of fuzziness ($m > 1$),

$X = (x_1, \dots, x_n)$ is a set of data points,

$C = (c_1, \dots, c_c)$ is the set of cluster prototypes ,

$U = (u_{ij})_{cn}$ is the fuzzy partition matrix ,

$d_{ij} = d(c_i, x_j)$.

We apply the Lagrange multipliers method to solve the above optimization problem.

Let $\lambda_j, 0 \leq j \leq n$ be the Lagrange multipliers in accordance with (2,4). Then, the

Lagrangian is

$$\zeta(U, C, \lambda, X, m) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ij}^2 + \sum_{j=1}^n \lambda_j \left(1 - \sum_{i=1}^c u_{ij}\right)$$

Minimizing the membership and the prototype yields the following optimal membership and cluster prototype update formula for the i th prototype and j th data point

2.8.4 Fuzzy C -Means Algorithm (FCM)

Steps in the Algorithm

1. Initialization:

- Initialize the membership matrix $U = [u_{ij}]$ with random values such that the sum of the memberships for each data point is 1:

$$\sum_{j=1}^C u_{ij} = 1, \quad \forall i \in \{1, \dots, N\}$$

2. Repeat until convergence:

- Update the centroids \mathbf{c}_j using the centroid update equation.
- Update the membership matrix U using the membership update equation.
- Check for convergence. The algorithm converges when the change in the membership matrix U falls below a predefined threshold or after a maximum number of iterations.

Convergence Criterion

The convergence of the Fuzzy C -Means algorithm is typically determined by checking whether the change in the membership values is below a certain threshold, or if the maximum number of iterations has been reached.

$$\|U^{(t+1)} - U^{(t)}\| < \epsilon$$

where ϵ is a small positive number (e.g., 10^{-5}).

2.9 Gaussian Mixture Models (GMM)

Objective

The objective of GMM clustering is to maximize the likelihood of the observed data under the model. This is achieved by estimating the parameters of the Gaussian components (means, covariances, and mixing coefficients).

2.9.1 Model Definition

A GMM is defined as a weighted sum of K Gaussian components:

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k)$$

where:

- π_k is the weight (mixing coefficient) of the k -th Gaussian component, with $\sum_{k=1}^K \pi_k = 1$ and $\pi_k \geq 0$.
- $\mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k)$ is the Gaussian distribution with mean μ_k and covariance matrix Σ_k .

The Gaussian distribution for a data point \mathbf{x} given mean μ_k and covariance matrix Σ_k is:

$$\mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k)\right)$$

where d is the dimensionality of the data.

Observation $X = (x_1, x_2, \dots, x_p) \in R^p$ are assumed iid with density $f(X_i) = \sum_{j=1}^k$.

$$f(x_i) = \sum_{j=1}^k \pi_j \varphi_{a_j}, \sum_j (X_j) \tag{2.6}$$

parameters $\pi_i \sum_j$ will be estimated by maximum likelihood

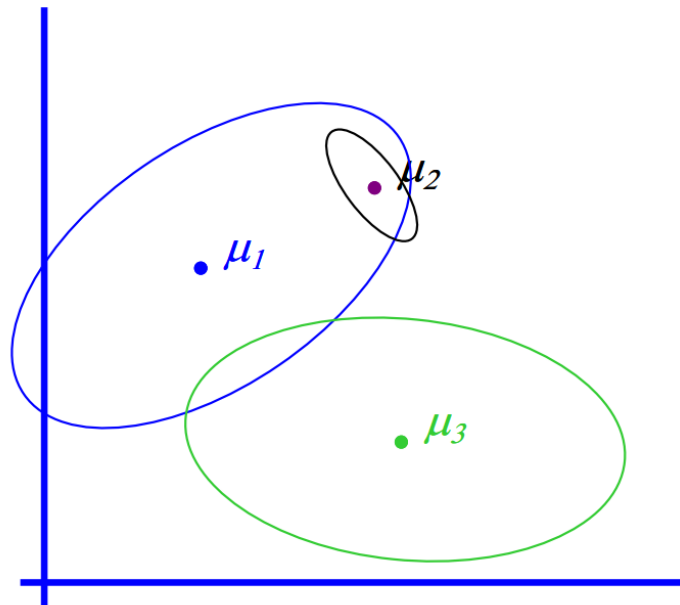
2.9.2 Mixture of K -Gaussian distributions: (Multi-modal distribution)

2.9.3 Gaussian Mixture Model (GMM)

- There are k components.
- Component i has an associated mean vector μ_i .
- Each component generates data from a Gaussian with mean μ_i and covariance matrix Σ_i .
- Each data point is generated according to the following recipe:
 - Pick a component at random: Choose component i with probability $P(y = i)$.
 - $X \sim (\mu_i, \Sigma_i)$.

$$p(x|y = i) \sim \mathcal{N}(\mu_i, \sigma^2 \mathbf{I})$$

$$p(x) = \sum_i p(x|y = i) P(y = i)$$



2.9.4 Expectation-Maximization (EM) Algorithm

The EM algorithm is used to find the maximum likelihood estimates of the parameters.

It iteratively performs the following steps:

Initialization

Initialize the parameters π_k , μ_k , and Σ_k for each Gaussian component.

Expectation Step (E-step)

Calculate the responsibility that each Gaussian component k takes for each data point \mathbf{x}_i :

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x}_i | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_i | \mu_j, \Sigma_j)}$$

where γ_{ik} is the posterior probability that data point \mathbf{x}_i belongs to cluster k .

Maximization Step (M-step)

Update the parameters π_k , μ_k , and Σ_k based on the responsibilities:

$$\begin{aligned} N_k &= \sum_{i=1}^N \gamma_{ik} \\ \pi_k &= \frac{N_k}{N} \\ \mu_k &= \frac{1}{N_k} \sum_{i=1}^N \gamma_{ik} \mathbf{x}_i \\ \Sigma_k &= \frac{1}{N_k} \sum_{i=1}^N \gamma_{ik} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T \end{aligned}$$

where N is the total number of data points, and N_k is the effective number of points assigned to cluster k .

Repeat

Repeat the E-step and M-step until convergence, i.e., until the parameters do not change significantly.

Convergence

The EM algorithm converges when the change in the log-likelihood of the data given the model parameters falls below a predefined threshold or after a maximum number of iterations.

Log-Likelihood

The log-likelihood of the data given the model parameters is:

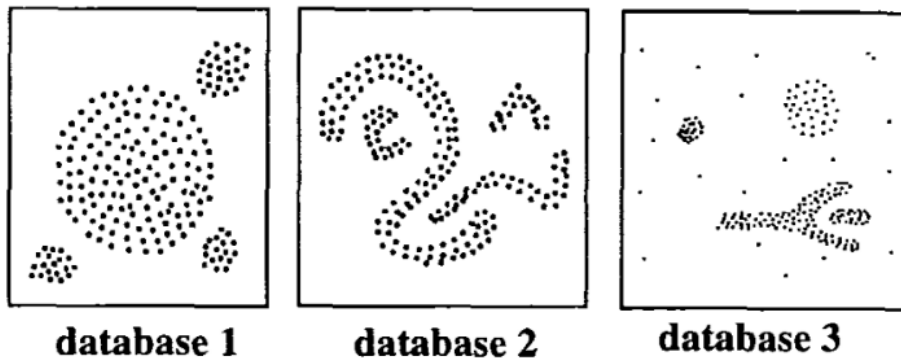
$$\log L(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i | \mu_k, \boldsymbol{\Sigma}_k) \right)$$

Maximizing this log-likelihood function with respect to the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ is the goal of the EM algorithm in GMM clustering.

2.9.5 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

The density-based clustering algorithm is introduced in Ester 1996 Ester (2018), which can be used to identify clusters of any shape in a data set containing noise and outliers. The basic idea behind the density-based clustering approach is derived from a human intuitive clustering method. For instance, by looking at the figure below, one can easily identify four clusters along with several points of noise, because of the different in the density of points. Clusters are dense regions in the data space, separated by regions of lower density of points. The DBSCAN algorithm is based on this intuitive notion of “clusters” and “noise”. The key idea is that for each point of a cluster, the

neighborhood of a given radius has to contain at least a minimum number of points.



2.9.6 Why DBSCAN?

Partitioning methods (K -means, PAM clustering) and hierarchical clustering are suitable for finding spherical-shaped clusters or convex clusters. In other words, they work well only for compact and well separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.

2.9.7 DBSCAN Clustering Algorithm

Let $D = \{x_1, x_2, \dots, x_n\}$ be the set of data points to be clustered.

Definitions

- **Core Point:** A point x_i is a core point if:

$$|N_\epsilon(x_i)| \geq \text{minPts}$$

where $N_\epsilon(x_i)$ is the ϵ -neighborhood of x_i .

- **Directly Density Reachable:** A point x_j is directly density reachable from a core point x_i if:

$$|N_\epsilon(x_j)| \geq \text{minPts} \quad \text{and} \quad x_j \in N_\epsilon(x_i)$$

- **Density Reachable:** x_j is density reachable from x_i if there exists a chain of

points x_1, x_2, \dots, x_k such that $x_1 = x_i, x_k = x_j$, and each x_{m+1} is directly density reachable from x_m .

- **Density Connected (Cluster):** A cluster is a maximal set of density connected points.
- **Noise (Outlier):** Points that do not belong to any cluster are considered noise or outliers.

Parameters

- ϵ : Maximum radius of the neighborhood.
- minPts: Minimum number of points within ϵ distance.

Algorithm

The DBSCAN algorithm iteratively finds all core points and expands clusters from them based on density reachability until all points are either assigned to a cluster or marked as noise.

2.9.8 Distance Calculation in DBSCAN Clustering

Let $D = \{x_1, x_2, \dots, x_n\}$ denote the set of data points in your dataset.

Distance Metric

Let $d(x_i, x_j)$ represent the distance between points x_i and x_j . This distance metric d can be any suitable measure, such as:

- **Euclidean Distance:** For points $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ and $x_j = (x_{j1}, x_{j2}, \dots, x_{jd})$:

$$d_{\text{euclidean}}(x_i, x_j) = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2}$$

— **Manhattan Distance:**

$$d_{\text{manhattan}}(x_i, x_j) = \sum_{k=1}^d |x_{ik} - x_{jk}|$$

- **Other Distance Metrics:** Depending on your data and problem, other metrics like cosine distance, Minkowski distance, or custom-defined distances might be appropriate.

Distance Matrix

Compute the pairwise distances between all points in D to form a distance matrix \mathbf{D} , where $\mathbf{D}_{ij} = d(x_i, x_j)$.

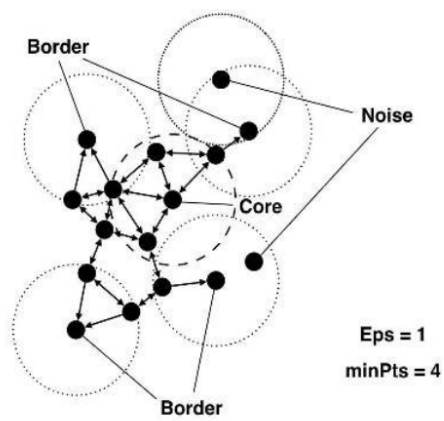
$$\mathbf{D} = \begin{pmatrix} d(x_1, x_1) & d(x_1, x_2) & \cdots & d(x_1, x_n) \\ d(x_2, x_1) & d(x_2, x_2) & \cdots & d(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(x_n, x_1) & d(x_n, x_2) & \cdots & d(x_n, x_n) \end{pmatrix}$$

Example: Euclidean Distance

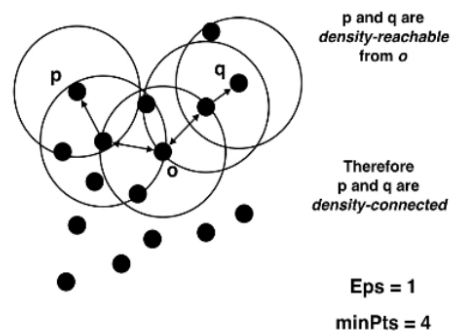
For a concrete example, using the Euclidean distance metric $d_{\text{euclidean}}$, the distance between two points $x_i = (x_{i1}, x_{i2})$ and $x_j = (x_{j1}, x_{j2})$ in a 2-dimensional space is:

$$d_{\text{euclidean}}(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2}$$

This formalization provides a clear and structured approach to calculating distances between points, essential for implementing DBSCAN clustering effectively.



(a)



(b)

Figure 2.12 – Illustration of DBSCAN Clustering Concepts

Chapitre 3

Financial Application

3.1 Introduction

In this chapter, we will explore applications of statistical clustering methods in the context of financial data analysis.

Clustering is a powerful technique that helps in identifying patterns, segmenting data, and uncovering hidden structures within large datasets. Financial data, characterized by its volume, velocity, and variety, presents a perfect domain for the application of these methods

In the first steps, we will simulate data to understand and compare different clustering methods.

In the second step, we will apply clustering methods to a real financial dataset.

This practical approach will demonstrate how clustering can be used to extract meaningful insights from financial data. Simulation studies and for real data are carried out using R software.

3.2 Simulation

In this step, we simulate some samples from random data with different distributions

3.2.1 Univariate Data

- 1. **Histogram Clustering:** We generate a mixture of two normally distributed datasets (3.1) ($\mathcal{N}(0, 1), \mathcal{N}(1, 1)$) and three normally distributed datasets (3.2) ($\mathcal{N}(0, 1), \mathcal{N}(1, 1), \mathcal{N}(2, 1)$) for demonstration purposes.

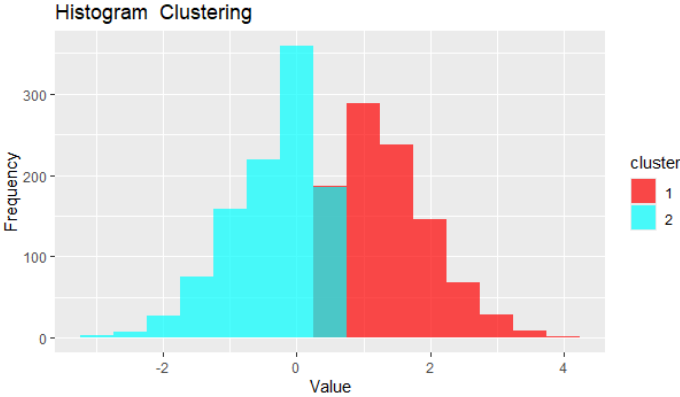


Figure 3.1 – Histogram Clustering for two data sets

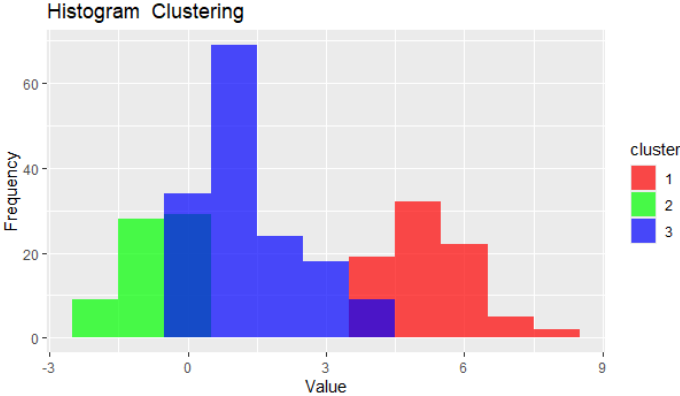


Figure 3.2 – Histogram Clustering for three data sets

2. **GMM Clustering:** We generate a mixture of four normally distributed datasets (3.3) ($\mathcal{N}(0, 2), \mathcal{N}(8, 4), \mathcal{N}(15, 2), \mathcal{N}(25, 3)$) for demonstration purposes.



Figure 3.3 – Density Clustering

3. **Fuzzy C -Means Clustering of Two Mixture Univariate Datasets:** We generate one mixture univariate datasets (data1 and data2) using `rnorm`. Each dataset consists of 50 points from two different normal distributions. We combine these datasets into a single vector data. We specify the number of clusters and the fuzziness parameter (m) as 2 for the Fuzzy C -Means algorithm. We use the `cmeans` function from the `e1071` package to perform Fuzzy C -Means clustering on data.

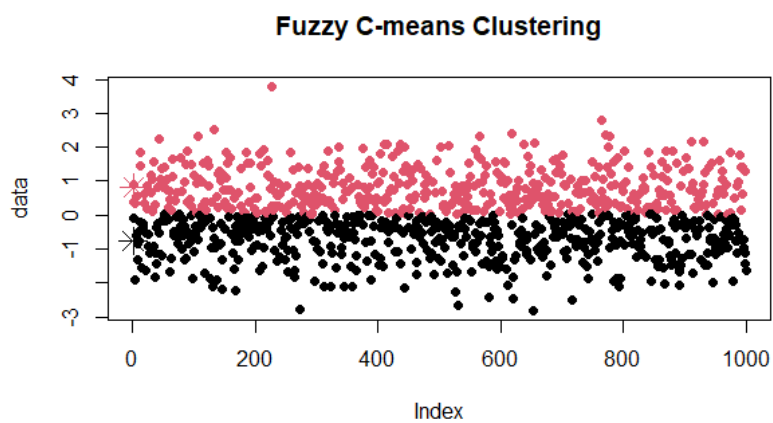


Figure 3.4 – Fuzzy C -means Clustering for Univariate sample

3.3 Bivariate Data

3.3.1 K -means method

We apply the K -means method to cluster simulated three sample data of size 10000 with normal distribution with a means 0, 5, 10 with the same variance respectively. we



Figure 3.5 – K -means methods for three clusters

use the K -means clustering algorithm to group synthetic data points into three distinct clusters, and visualizes the results. In this (3.5), we can see that we have kept the same number of clusters.

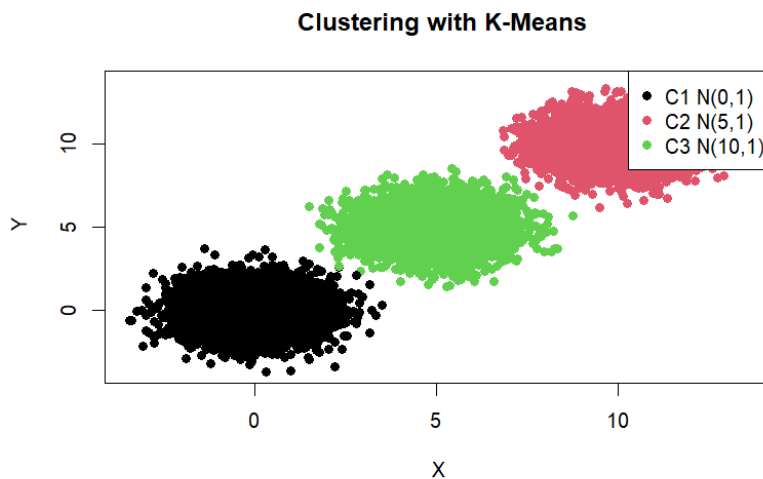


Figure 3.6 – K -means methods for three cluster

In this example we have the same rate

1. **Gaussian Mixture method:**

we use the Gaussian method to clustering a simulated data This block generates

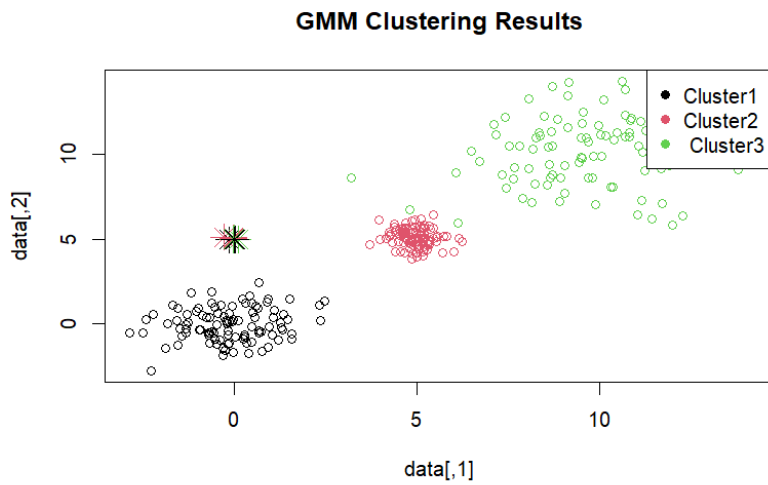


Figure 3.7 – Gaussian Mixture method

a synthetic data set with three distinct clusters. Each cluster is created by generating data from a normal distribution with different means and standard deviations:

- (a) The first cluster has a mean 10 and a standard deviation of 1.
- (b) The second cluster has a mean 10 and a standard deviation of 2.
- (c) The third cluster has a mean 10 and a standard deviation of 0.5.

2. DBSCAN method:

We use the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm for data clustering and cluster display, membership prediction for new data, and geographical data clustering using the DBSCAN algorithm. To evaluate the efficacy of DBSCAN in contrast to alternative techniques, it also contains a performance comparison with other implementations.

Uses the following parameters to apply the DBSCAN algorithm to standardized iris scaled data:

- ($\text{eps} = 0.5$): Maximum radius around a point to define its vicinity.
- ($\text{minPts} = 5$): Minimum number of points in a neighborhood for a point to be considered a core point.

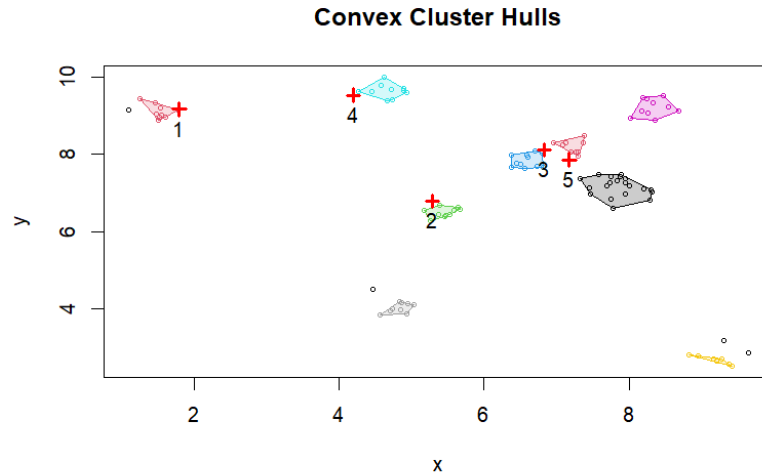


Figure 3.8 – DBSCAN methods

3. **Hierarchical method:** we use the Hierarchical method to clustering a simulated data.

Hierarchical clustering methods are particularly effective for analyzing real data when the relationships between data points can be represented hierarchically. Here's how hierarchical clustering works with real data and its applications.

3.4 Clustering real data application:

3.4.1 Cluster Analysis

Cluster analysis is a statistical approach that classifies items into mutually exclusive groups (clusters) so that members of each group are as similar to one another as possible while remaining as distinct as possible from members of other groups

3.5 Applying Clustering Methods to Cluster Cryptocurrencies

3.5.1 The Basics about Cryptocurrency

Cryptocurrency comes under many names. You have probably read about some of the most popular types of cryptocurrencies such as Bitcoin, Litecoin, and Ethereum. Cryptocurrencies are increasingly popular alternatives for online payments. Before converting real dollars, euros, pounds, or other traditional currencies into (the symbol for Bitcoin, the most popular cryptocurrency), you should understand what cryptocurrencies are, what the risks are in using cryptocurrencies, and how to protect your investment.

3.5.2 What is cryptocurrency?

A cryptocurrency is a digital currency, which is an alternative form of payment created using encryption algorithms. The use of encryption technologies means that cryptocurrencies function both as a currency and as a virtual accounting system. To use cryptocurrencies, you need a cryptocurrency wallet. These wallets can be software that is a cloud-based service or is stored on your computer or on your mobile device. The wallets are the tool through which you store your encryption keys that confirm your identity and link to your cryptocurrency.

3.5.3 What are the risks to using cryptocurrency?

Cryptocurrencies are still relatively new, and the market for these digital currencies is very volatile. Since cryptocurrencies don't need banks or any other third party to regulate them; they tend to be uninsured and are hard to convert into a form of tangible currency (such as US dollars or euros.) In addition, since cryptocurrencies are technology-based intangible assets, they can be hacked like any other intangible

technology asset. Finally, since you store your cryptocurrencies in a digital wallet, if you lose your wallet (or access to it or to wallet backups), you have lost your entire cryptocurrency investment.

3.5.4 Key Features of Cryptocurrencies:

1. **Decentralization:** Cryptocurrencies operate on decentralized networks of computers (nodes) that use consensus mechanisms to validate and record transactions. This decentralization removes the need for intermediaries like banks or payment processors, making transactions more direct and potentially more efficient.
2. **Blockchain Technology:** Most cryptocurrencies use blockchain technology, which is a distributed ledger that records all transactions across a network of computers. Each block in the blockchain contains a set of transactions, and new blocks are added to the chain in a chronological order through a process called mining.
3. **Security:** Cryptocurrencies use cryptographic techniques to secure transactions and control the creation of new units. Public and private keys are used to authenticate transactions and ensure that only the owner of the cryptocurrency can access and transfer their funds.
4. **Anonymity and Transparency:** While cryptocurrency transactions are pseudonymous (meaning they are linked to addresses rather than real-world identities), the blockchain provides a transparent and immutable record of all transactions. This transparency can enhance trust and accountability in the cryptocurrency ecosystem.
5. **Limited Supply:** Many cryptocurrencies have a predetermined maximum supply or a fixed issuance schedule, which helps prevent inflation and maintain the value of the currency over time. For example, Bitcoin has a maximum supply of 21 million coins.

3.5.5 Examples of Cryptocurrencies:

1. **Bitcoin (BTC):** Bitcoin is the first and most well-known cryptocurrency, created by an unknown person or group of people using the pseudonym Satoshi Nakamoto in 2009.
2. **Ethereum (ETH):** Ethereum is a decentralized platform that enables developers to build and deploy smart contracts and decentralized applications (DApps). It was proposed by Vitalik Buterin in late 2013 and development was crowdfunded in 2014, with the network going live on July 30, 2015.
3. **Ripple (XRP):** Ripple is a digital payment protocol and cryptocurrency that aims to enable fast and low-cost cross-border transactions. It was released in 2012 by Ripple Labs Inc., a technology company based in San Francisco, California.
4. **Litecoin (LTC):** Litecoin is a peer-to-peer cryptocurrency created by Charlie Lee in 2011. It is based on the Bitcoin protocol but with some differences, such as a shorter block generation time and a different hashing algorithm.
5. **Cardano (ADA):** Cardano is a blockchain platform that aims to provide a more secure and scalable infrastructure for the development and execution of smart contracts and DApps. It was founded by Charles Hoskinson, one of the co-founders of Ethereum, and launched in 2017.

These examples represent just a small fraction of the thousands of cryptocurrencies that exist today. Each cryptocurrency has its own unique features, use cases, and communities, contributing to the diverse and dynamic landscape of the cryptocurrency market.

3.5.6 Clustering cryptocurrencies

Applying clustering methods to cluster cryptocurrencies involves similar steps to clustering oil prices. Here's how you can do it:

1. **Data Collection:** Gather historical data for various cryptocurrencies, including dollar exchange rate, trading volume, market capitalization, and other relevant metrics.
2. **Feature Selection:** Decide which features to use for clustering. Consider incorporating price data, trading volume, market capitalization, price volatility, correlations with other cryptocurrencies or financial instruments, and technical indicators.
3. **Data Preprocessing:** Normalize or scale the features as necessary to ensure they are on a similar scale and comparable.
4. **Choose Clustering Algorithm:** Select a clustering algorithm suitable for your data and objectives, such as K -means, hierarchical clustering, DBSCAN, or Gaussian mixture models.
5. **Determine Number of Clusters:** Decide on the number of clusters to use, using techniques such as the elbow method or silhouette score.
6. **Apply Clustering Algorithm:** Apply the chosen clustering algorithm to the preprocessed cryptocurrency data.
7. **Interpret Results:** Analyze the clusters formed by the algorithm, looking for patterns or insights in each cluster.
8. **Evaluate and Refine:** Evaluate the quality of the clustering results and refine the analysis as needed by adjusting parameters or trying different algorithms.
9. **Visualization:** Visualize the clustering results using scatter plots, heatmaps, or other techniques to make them interpretable.
10. **Monitoring:** Continuously monitor cryptocurrency data and update the clustering analysis periodically to identify emerging trends or patterns in the cryptocurrency market.

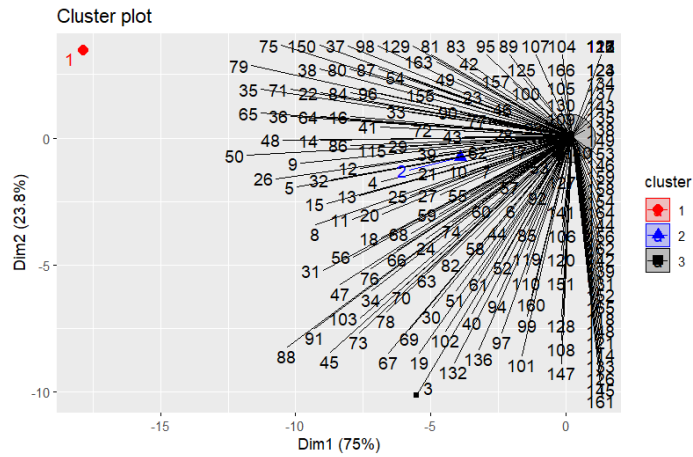


Figure 3.9 – Clustering for three clusters

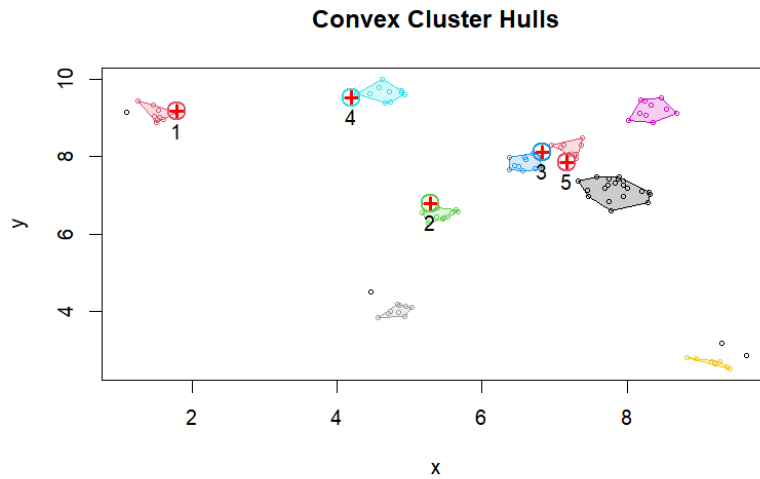


Figure 3.10 – DBSCAN Clustering

This figure shows a 2D scatter plot with clusters represented by different colors. The clusters are enclosed by convex hulls, and each cluster has a label indicating its number. Red crosses denote the centroids of each cluster. The plot is titled “Convex Cluster Hulls” and demonstrates the spatial distribution of clusters in the $x - y$ plane.

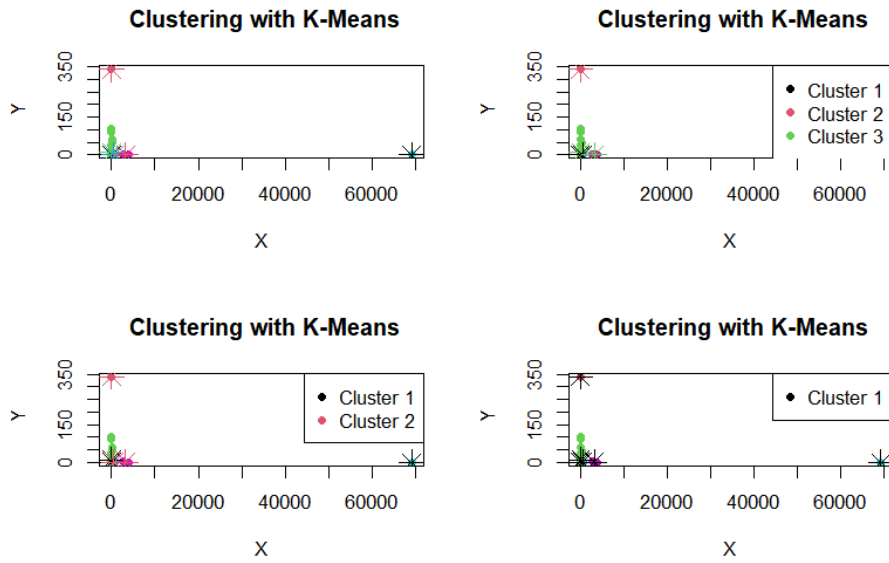


Figure 3.11 – K -means

This figure consists of four subplots showing the results of clustering using the K -Means algorithm.

Each subchart is labeled “Clustering with K -Means” and displays clusters in a two-dimensional scatter plot with X and Y axes.

The top left subplot shows clustering in 4 clusters, while the top right subplot shows clustering in 3 clusters, The lower left subgraph shows the clustering in 2 clusters with different initialization, The lower right bottom subgraph shows the grouping in one group.

Different clusters are indicated by different colors and symbols.

The X -axis represents a range of values up to approximately 70,000, while the Y -axis ranges from 0 to 350.

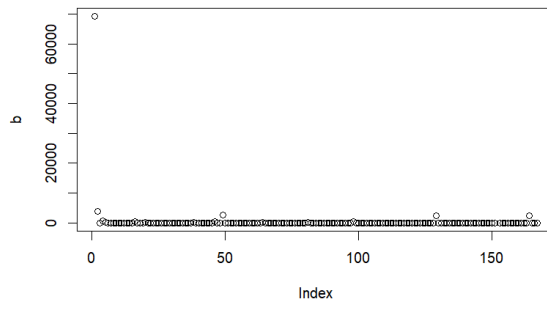


Figure 3.12 – Data of Price

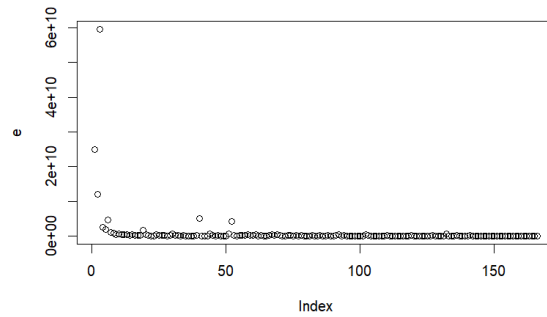


Figure 3.13 – Data of Volume

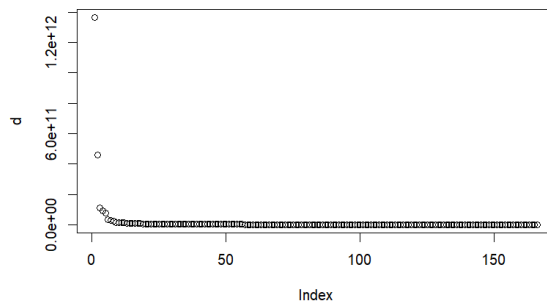


Figure 3.14 – Data of Market-Cap

These scatter plots show the distribution of the variable “e,d,b” across different indices. It is clear from the graphs that there are a few outliers where the variable “e,d,b” reaches very high values, while the majority of the data points are clustered near zero. This indicates that the e variable has a highly skewed distribution, with a long tail extending towards higher values.

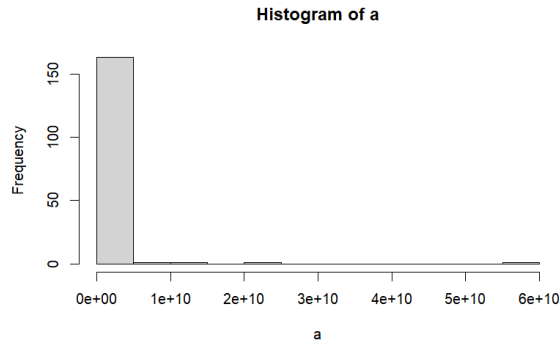


Figure 3.15 – Histogramm of data Price

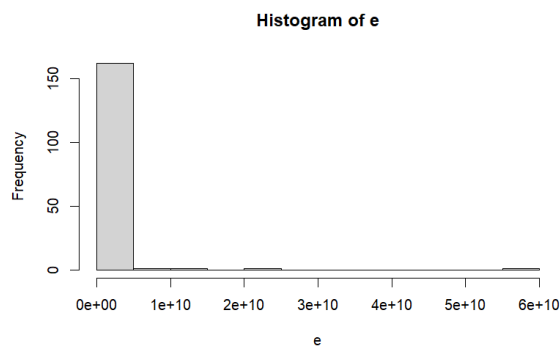


Figure 3.16 – Histogramm of Volume

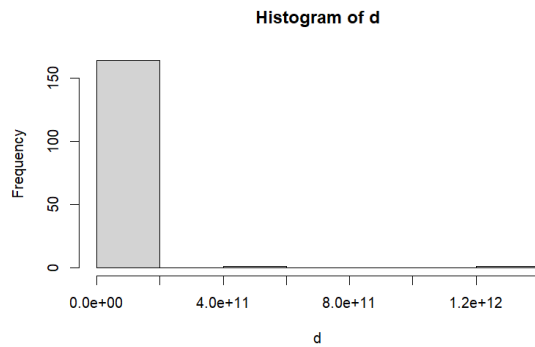


Figure 3.17 – Histogramm of data Market-Cap

The histogram shown represents the distribution of cryptocurrencies (Market-Cap, Volume, Price). As observed, the majority of the data points are concentrated at the lower end of the range (near zero), with very few occurrences of higher values. This indicates a highly skewed distribution. Most of the frequency counts are in the first bin, suggesting that 'Market-Cap, Volume, Price' has a large number of very small

values and a few extremely large values.

3.5.7 Applying Clustering Methods to Cluster Development Indices

3.5.8 Background and Motivation

The development and economic progress of countries around the world are measured using various indices that reflect key aspects such as health, education, and income. These indices provide a comprehensive view of a country's socio-economic status and are crucial for policy-making, resource allocation, and international comparisons. However, the sheer volume and complexity of the data can make it challenging to identify patterns and insights that can inform effective decision-making. This study aims to employ clustering techniques to analyze the development indices of 48 countries, thereby uncovering underlying patterns and grouping countries with similar development profiles.

3.6 Importance of Clustering Development Indices

Clustering is an unsupervised machine learning technique that groups similar data points based on their characteristics. When applied to development indices, clustering can reveal natural groupings of countries with similar socio-economic profiles. These groupings can:

- **Facilitate Comparative Analysis:** By clustering countries with similar development indices, policymakers and researchers can perform more targeted comparative analyses.
- **Identify Development Patterns:** Clustering can uncover hidden patterns and trends in development data that might not be apparent through traditional analysis methods.

- **Assist in Policy Formulation:** Understanding which countries share similar development challenges and strengths can help in formulating more effective and tailored policy interventions.
- **Enhance Resource Allocation:** Clustering can guide international organizations and governments in allocating resources more efficiently by targeting clusters of countries with similar needs.

3.6.1 Development Indices

Development indices of countries are quantitative measures that capture various aspects of a nation's socio-economic progress and well-being. These indices are designed to provide a comprehensive overview of a country's development status by aggregating data across multiple dimensions, such as economic performance, health, education, and living standards.

They serve as critical tools for policymakers, researchers, and international organizations to assess and compare the development levels of different countries, track progress over time, and identify areas requiring intervention and improvement.

1. **Human Development Index (HDI):** The Human Development Index, is a composite statistical index designed to assess the rate of human development of the world's countries. The HDI was initially based on three criteria: GDP per capita, life expectancy at birth and educational attainment of children aged 17 and over.
2. **PIB per Inhabitant:** Gross domestic product per capita, or per capita (GDP per capita or per capita) is an indicator of the level of economic activity. It is the value of GDP divided by the number of inhabitants in a country. It is more effective than GDP in measuring a country's development, but it is only an average, and therefore cannot account for inequalities of income and wealth within a population.
3. **IPM:** The Multidimensional Poverty Index (MPI)¹, also known as the

Multidimensional Poverty Index, is a statistical index assessing poverty in developing countries, created by a department at Oxford University in 2010.

4. **Gini Index:** The Gini coefficient, or Gini index, is a statistical measure of the distribution of a variable (income, wealth, etc.) within a population. Primarily, it measures the degree of income inequality in a country¹.
5. **IPH (HPI):** (Human Poverty Index) The Human Poverty Index or HPI is a composite index, created by the UNDP (United Nations Development Program), whose aim is to measure the level of poverty within a country.
6. **IDG:** IDG, Infrastructure de Données Géographiques, in English SDI for Spatial Data Infrastructure) is an organization based on sharing agreements, coordination between its members and IT systems that integrate a set of services (catalogs, servers, software, data, applications, web pages, etc.) used to manage geographic information (maps, orthophotoplans, satellite images, etc.).
7. **Life Quality Index:** The Life Quality Index (LQI) is a calibrated compound social indicator of human welfare that reflects the expected length of life in good health and enhancement of the quality of life through access to income. The Life Quality Index combines two primary social indicators: the expectancy of healthy life at birth, E, and the real gross domestic product per person, G, corrected for purchasing power parity as appropriate. Both are widely available and accurate statistics.
8. **IPC** The Integrated Food Security Phase Classification (IPC) is a set of analytical tools and processes used to analyze and classify the severity of food insecurity according to international scientific standards.
9. **ICG:** The International Crisis Group, also known as Crisis Group, is an international non-profit NGO founded in 1995, whose mission is to prevent and help resolve deadly conflicts through independent field research, analysis and recommendations.

3.6.2 Objectives of the Study

The primary objectives of this study are to gather and preprocess development indices data for 48 countries, identifying relevant features such as GDP per capita, literacy rate, life expectancy, and access to healthcare. Various clustering algorithms, including K-means, hierarchical clustering, and DBSCAN, will be applied to group the countries based on these indices. The resulting clusters will be analyzed to identify common characteristics and differences among the countries. Visualization techniques will be employed to present the clustering results in an interpretable manner. Finally, the implications of these results for international development policies and resource allocation will be discussed.

3.6.3 Clustering Index development

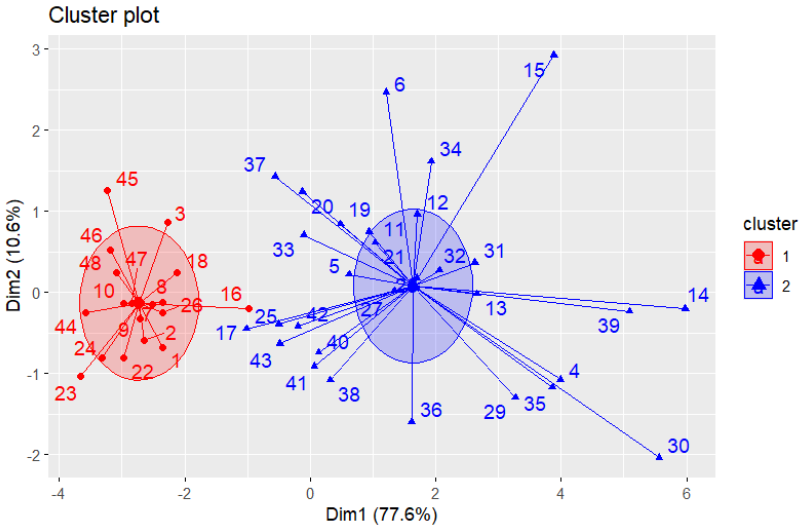


Figure 3.18 – *k*-means cluster of the index development for 48 countries with Two clusters.

This figure is a cluster plot showing the results of a clustering algorithm in 2D space. There are two clusters:

Cluster 1 in red and Cluster 2 in blue. Points represent the clusters, enclosed by ellipses, with the *X*-axis (Dim1) accounting for 77.6% of the variance and the *Y*-axis (Dim2) for 10.6%.

Arrows indicate the direction and distance of points from the cluster centroids. The

plot is titled “Cluster plot,” and the clusters are distinguished by color and shape (see, 3.18)

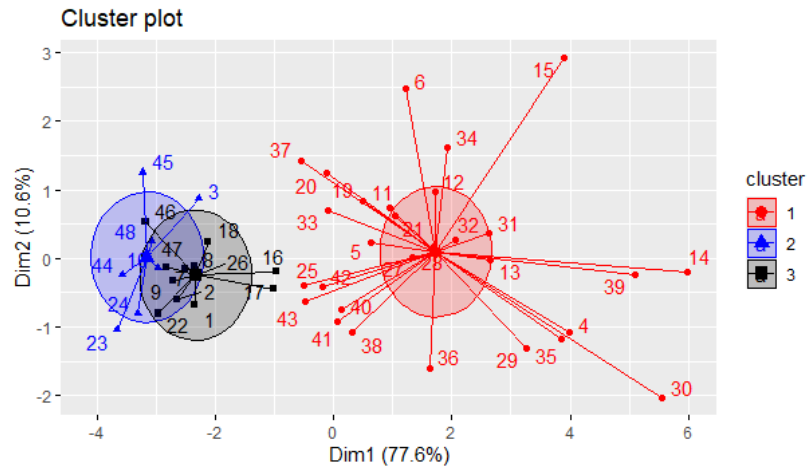


Figure 3.19 – *k*-means cluster of the index development for 48 countries with Three clusters.

This figure is a cluster plot showing the results of a clustering algorithm in 2D space. There are three clusters: Cluster 1 in red, Cluster 2 in black and Cluster 3 in blue. Points represent the clusters, enclosed by ellipses, with the *X*-axis (Dim1) accounting for 77.6% of the variance and the *Y*-axis (Dim2) for 10.6%. Arrows indicate the direction and distance of points from the cluster centroids. The plot is titled “Cluster plot,” and the clusters are distinguished by color and shape (see, 3.19).

In conclusion, we can say that clustering for two clusters is better than clustering with three clusters, because in the first case, the two clusters are well separated.

Summary of Findings

A detailed review of traditional clustering methods, including K-means, hierarchical clustering, DBSCAN, and Gaussian Mixture Models, Fuzzy clustering methods, particularly Fuzzy C-means (FCM), were introduced and their advantages over traditional methods were highlighted. FCM's ability to handle overlapping clusters and provide a more nuanced representation of data was emphasized was conducted. Their strengths, limitations, and suitable application scenarios were discussed.

The performance of various clustering algorithms was implemented and evaluated on both synthetic and real-world datasets. The impact of different parameter settings and initialization strategies on clustering outcomes was investigated.

Potential improvements and hybrid approaches were proposed to enhance the robustness and applicability of clustering methods. These included combining multiple clustering techniques and optimizing parameter selection.

Techniques for visualizing and interpreting clustering results were explored, demonstrating the importance of clear and effective data presentation in understanding clustering outcomes.

The practical applications of clustering methods in various domains, such as bioinformatics, market segmentation, image processing, and social network analysis, were discussed. Case studies and examples illustrated the versatility and impact of clustering techniques in real-world scenarios.

Contributions to the Field

This thesis has made several contributions to the field of data clustering:

- Provided a comprehensive review and comparison of traditional and fuzzy clustering methods.
- Demonstrated the practical application and effectiveness of Fuzzy C-means clustering in handling complex datasets.
- Highlighted the importance of visualization and interpretation in the clustering process.

Future Research Directions

While significant progress has been made, there are several areas for future research:

- **Scalability and Efficiency:** Further research is needed to develop scalable and efficient clustering algorithms capable of handling large datasets.
- **Dynamic Clustering:** Exploring dynamic clustering methods that can adapt to changes in data over time and handle streaming data effectively.
- **Integration with Other Techniques:** Investigating the integration of clustering with other machine learning and data mining techniques, such as classification and regression, to enhance overall data analysis capabilities.
- **Application-Specific Adaptations:** Developing clustering methods tailored to specific applications and domains, addressing unique challenges and requirements.

In conclusion, this thesis has advanced the understanding and application of clustering methods in data analysis.

This study contributes to the ongoing development and refinement of clustering techniques, paving the way for more effective and insightful data analysis in various fields.

Bibliography

- Coppejans, M. and Sieg, H. (2005). Kernel estimation of average derivatives and differences. *Journal of Business & Economic Statistics*, 23(2):211–225.
- Ester, M. (2018). Density-based clustering. *Data Clustering*, pages 111–127.
- Green, P. E. and Krieger, A. M. (1995). A comparison of alternative approaches to cluster-based market segmentation. *Market Research Society. Journal.*, 37(3):1–19.
- Hodson, F. R. (1970). Cluster analysis and archaeology: some new developments and applications. *World archaeology*, 1(3):299–320.
- Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning.
- Kobren, A., Monath, N., Krishnamurthy, A., and McCallum, A. (2017). A hierarchical algorithm for extreme clustering. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 255–264.
- Li, H., Correa, N. M., Rodriguez, P. A., Calhoun, V. D., and Adali, T. (2011). Application of independent component analysis with adaptive density model to complex-valued fmri data. *IEEE Transactions on biomedical engineering*, 58(10):2794–2803.
- Macri, A. R. (1963). *A comparison of the effectiveness of two teaching methods on the competence of college students to understand atomic structure in a one semestre course in general physical science*. New York University.

- Paykel, E. S., Emms, E., Fletcher, J., and Rassaby, E. (1980). Life events and social support in puerperal depression. *The British Journal of Psychiatry*, 136(4):339–346.
- Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gbscan and its applications. *Data mining and knowledge discovery*, 2:169–194.
- Steinley, D. (2006). K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, 59(1):1–34.

Appendices

Appendix 1: Histogram Clustering

```
1 # Load necessary libraries
2 library(ggplot2)
3
4 # Generate some example mixture univariate data
5 #set.seed(123)
6 data1 <- rnorm(100, mean = 0, sd = 1)
7 data2 <- rnorm(100, mean = 1, sd = 1)
8 data3 <- rnorm(100, mean = 5, sd = 1)
9
10 data <- c(data1, data2, data3)
11
12 # Create a histogram of the data
13 hist_data <- hist(data, breaks = "Sturges", plot = FALSE)
14
15 # Extract the midpoints of the histogram bins
16 bin_midpoints <- hist_data$mids
17
18 # Apply k-means clustering to the bin midpoints
19 #set.seed(123)
```

```

20 num_clusters <- 3 # Change this to the number of clusters you
    want
21 kmeans_result <- kmeans(bin_midpoints, centers = num_clusters)
22
23 # Assign each data point to the nearest cluster based on the
    bin it falls into
24 bin_assignments <- cut(data, breaks = hist_data$breaks, labels
    = FALSE, include.lowest = TRUE)
25 data_clusters <- kmeans_result$cluster[bin_assignments]
26
27 # View the first few data points and their corresponding
    clusters
28 head(data)
29 head(data_clusters)
30
31 # Plot data points colored by their cluster assignment
32 ggplot(data.frame(x = data, cluster = as.factor(data_clusters)))
    , aes(x, fill = cluster)) +
33   geom_histogram(binwidth = diff(hist_data$breaks)[1], alpha =
    0.7, position = "identity") +
34   labs(title = "Histogram Clustering ", x = "Value", y = "
    Frequency") +
35   scale_fill_manual(values = rainbow(num_clusters)) # Adding
    colors to clusters

```

Appendix 2: Cryptocurrencies Clustering

```

1
2 MJ <- Cryptocurrencies$'Market Cap'

```

```

3 V <- as.numeric(Cryptocurrencies$Price)
4 D <- Cryptocurrencies$'Volume(24h)'
5
6 (classement <- data.frame( MJ,V, D))
7
8 head(classement)
9 colnames(classement)
10 #Number of Columns.
11 ncol(classement)
12 #Number of Rows.
13 nrow(classement)
14 #Check if there are missing values.
15 sum(is.na(classement))
16 duplicates <- classement%>%duplicated()
17 #Displays how many duplicates are present in a table. If a
    value is not a duplicate, it is placed in 'FALSE'. If the
    value is a duplicate, it is placed in 'TRUE'.
18 duplicates_count <- duplicates%>%table()
19 duplicates_count
20 #Scales/standardizes the data.
21 Cryptocurrencie <- scale(classement)
22 #Views the scaled data.
23 head(Cryptocurrencie)
24
25 #Complete Linkage Clustering Method
26 hcluster_com <- hclust(dist(Cryptocurrencie), method = "
    complete")
27 plot(hcluster_com, main = "Complete Linkage Dendrogram")
28
29 #Average Linkage Clustering Method

```



```

30 hcluster_ave <- hclust(dist(Cryptocurrencie), method = "average
      ")
31 plot(hcluster_ave, main = "Average Linkage Dendrogram")
32
33 km.res <- kmeans(classement, 3, nstart = 20)
34 km.res
35 df_member <- cbind(classement, cluster = km.res$cluster)
36 head(df_member, 166)
37
38 library(factoextra)
39 library(cluster)
40 fviz_cluster(km.res, data = classement,
41               palette=c("red", "blue", "black", "darkgreen"),
42               ellipse.type = "euclid",
43               star.plot = T,
44               repel = T,
45               ggtheme = theme())

```

Appendix 3: Development Indices Clustering

```

1
2 data1 <- data.frame(
3   # Pays = c("France", "Allemagne", " tats -Unis", "Inde", "
      Chine", "Br sil", "Japon", "Royaume-Uni", "Canada", "
      Australie",
4   #           "Russie", "Mexique", "Indon sie", "Nig ria", "
      Afrique du Sud", "Italie", "Espagne", "Cor e du Sud", "
      Argentine",

```

```

5 # "Arabie Saoudite", "Turquie", "Pays-Bas", "Suisse
  ", "Su de", "Pologne", "Belgique", "Tha lande", "Iran", "
  Egypte",
6 # "Pakistan", "Philippines", "Vietnam", "Malaisie",
  "Colombie", "Bangladesh", "Ukraine", "Chili", "Kazakhstan",
  "Kenya",
7 # "Roumanie", "Hongrie", "Gr ce", "Portugal", "
  Irlande", "Singapour", "Hong Kong", "Nouvelle-Z lande", "
  Emirats Arabes Unis"),
8 PIBPH = c(41464, 46245, 62641, 2100, 10410, 8897, 40113,
  42330, 45045, 56329,
9           11430, 9360, 4120, 2222, 5996, 34240,
           30176, 34277, 10400,
10          21123, 12507, 52835, 81900, 55601,
           15730, 43139, 7279, 5983, 3560,
11          1430, 3120, 3450, 11515, 6067, 1900,
           3610, 14810, 11239, 1854,
12          12630, 15370, 20690, 23340, 84650,
           58603, 48250, 45800, 67300),
13 IDH = c(0.901, 0.939, 0.926, 0.645, 0.761, 0.765, 0.915,
  0.922, 0.929, 0.944,
14          0.824, 0.779, 0.718, 0.539, 0.705, 0.880, 0.891,
           0.916, 0.825,
15          0.854, 0.820, 0.934, 0.955, 0.945, 0.872, 0.924,
           0.765, 0.783, 0.707,
16          0.557, 0.706, 0.704, 0.805, 0.767, 0.661, 0.779,
           0.851, 0.811, 0.579,
17          0.828, 0.845, 0.872, 0.850, 0.942, 0.938, 0.939,
           0.920, 0.890),
18 IPM = c(0.007, 0.005, 0.008, 0.123, 0.038, 0.049, 0.004,
  0.006, 0.007, 0.003,

```

19 0.031, 0.051, 0.093, 0.187, 0.106, 0.025, 0.028,
 0.004, 0.045,
 20 0.018, 0.070, 0.004, 0.003, 0.002, 0.046, 0.005,
 0.031, 0.035, 0.127,
 21 0.197, 0.085, 0.052, 0.017, 0.064, 0.140, 0.056,
 0.007, 0.020, 0.165,
 22 0.044, 0.035, 0.055, 0.025, 0.002, 0.002, 0.002,
 0.002, 0.003),
 23 IG = c(0.29, 0.31, 0.41, 0.35, 0.38, 0.53, 0.33, 0.34, 0.32,
 0.34,
 24 0.41, 0.45, 0.39, 0.43, 0.62, 0.34, 0.32,
 0.36, 0.42,
 25 0.45, 0.42, 0.29, 0.28, 0.29, 0.33, 0.33,
 0.37, 0.38, 0.31,
 26 0.31, 0.42, 0.40, 0.41, 0.50, 0.32, 0.26,
 0.47, 0.29, 0.43,
 27 0.31, 0.30, 0.34, 0.32, 0.34, 0.45, 0.39,
 0.34, 0.38),
 28 IPH = c(0.045, 0.037, 0.059, 0.197, 0.085, 0.093, 0.041,
 0.046, 0.044, 0.035,
 29 0.073, 0.082, 0.154, 0.276, 0.229, 0.048, 0.047,
 0.037, 0.071,
 30 0.065, 0.057, 0.036, 0.033, 0.034, 0.054, 0.036,
 0.095, 0.100, 0.145,
 31 0.198, 0.120, 0.078, 0.040, 0.109, 0.156, 0.086,
 0.048, 0.052, 0.174,
 32 0.054, 0.045, 0.065, 0.045, 0.030, 0.025, 0.028,
 0.034, 0.020),
 33 IDG = c(0.995, 0.964, 0.992, 0.827, 0.942, 0.999, 0.993,
 0.975, 0.988, 0.981,













































34 0.955, 0.913, 0.903, 0.838, 0.894, 0.971, 0.972,
 0.985, 0.964,
 35 0.951, 0.940, 0.978, 0.976, 0.979, 0.960, 0.974,
 0.926, 0.927, 0.884,
 36 0.802, 0.899, 0.911, 0.948, 0.923, 0.891, 0.905,
 0.936, 0.914, 0.822,
 37 0.942, 0.930, 0.940, 0.935, 0.978, 0.976, 0.980,
 0.977, 0.967),
 38 IQV = c(7.9, 8.1, 7.2, 5.4, 6.7, 6.2, 8.0, 7.8, 7.9, 8.5,
 39 6.1, 6.0, 5.7, 4.9, 5.1, 7.5,
 7.4, 8.2, 6.8,
 40 6.7, 6.5, 8.1, 8.4, 8.6, 7.3,
 7.6, 6.3, 6.0, 5.6,
 41 5.2, 5.8, 5.7, 7.0, 6.2, 5.1,
 5.9, 7.4, 6.7, 5.3,
 42 7.1, 6.9, 7.3, 7.2, 8.3, 8.9,
 8.7, 8.4, 8.5),
 43 IPC = c(69, 80, 67, 40, 41, 38, 72, 77, 81, 73,
 44 29, 31, 38, 25, 44, 52, 57, 56, 42,
 45 53, 39, 83, 85, 84, 55, 75, 41, 29, 28,
 46 25, 36, 37, 48, 40, 25, 37, 70, 52, 27,
 47 44, 48, 55, 60, 85, 86, 88, 80, 83),
 48 ICG = c(79.8, 81.2, 83.7, 61.4, 73.9, 62.5, 82.6, 81.4, 83.3,
 83.1,
 49 64.8, 64.0, 63.4, 53.9, 60.1, 68.7, 68.0, 82.2, 66.5,
 50 70.4, 62.0, 82.7, 84.0, 84.2, 70.6, 81.2, 60.5, 59.3,
 57.6,
 51 51.5, 61.0, 60.8, 72.2, 61.5, 55.4, 60.2, 70.0, 63.7,
 52.9,
 52 67.4, 65.0, 68.0, 66.5, 84.6, 85.1, 85.2, 83.9, 84.5)
 53)

```

54 data1
55 (classement <- data1)
56 head(classement)
57 colnames(classement)
58 #Number of Columns.
59 ncol(classement)
60 #Number of Rows.
61 nrow(classement)
62 #Check if there are missing values.
63 sum(is.na(classement))
64 duplicates <- classement%>%duplicated()
65 #Displays how many duplicates are present in a table. If a
    value is not a duplicate, it is placed in 'FALSE'. If the
    value is a duplicate, it is placed in 'TRUE'.
66 duplicates_count <- duplicates%>%table()
67 duplicates_count
68 #Scales/standardizes the data.
69 USdata <- scale(classement)
70 #Views the scaled data.
71 head(USdata)
72
73 #Complete Linkage Clustering Method
74 hcluster_com <- hclust(dist(USdata), method = "complete")
75 plot(hcluster_com, main = "Complete Linkage Dendrogram")
76
77 #Average Linkage Clustering Method
78 hcluster_ave <- hclust(dist(USdata), method = "average")
79 plot(hcluster_ave, main = "Average Linkage Dendrogram")
80
81 km.res <- kmeans(classement, 3, nstart = 20)
82 km.res

```

```
83 df_member <- cbind(classement, cluster = km.res$cluster)
84 head(df_member,48)
85
86 library(factoextra)
87 library(cluster)
88 fviz_cluster(km.res, data = classement,
89               palette=c("red", "blue", "black", "darkgreen"),
90               ellipse.type = "euclid",
91               star.plot = T,
92               repel = T,
93               ggtheme = theme())
```

Rank	Name	Symbol	Market Cap	Price	Volume(24h)
1	 Bitcoin	BTC	1,364E+12	69199.21	24992292082
2	 Ethereum	ETH	4,582E+11	3814.12	11906845041
3	 Tether USDt	USDT	1,122E+11	0.9998	59516840776
4	 BNB	BNB	9,296E+10	629.87	2564716156
5	 Solana	SOL	7,593E+10	165.14	1832007554
6	 USD Coin	USDC	3,238E+10	1.00	4534472131
7	 XRP	XRP	2,883E+10	0.5199	1013278416
8	 Dogecoin	DOGE	2,343E+10	0.1621	765904252
9	 Cardano	ADA	1,621E+10	0.4541	291327875
10	 Toncoin	TON	1,594E+10	6.61	494816288
11	 Shiba Inu	SHIB	1,475E+10	0.00002502	478836295
12	 Avalanche	AVAX	1,414E+10	35.97	309135586
13	 Chainlink	LINK	1,053E+10	17.94	316311584
14	 Polkadot	DOT	1,024E+10	7.12	162197910
15	 TRON	TRX	9,94E+09	0.1138	289874030
16	 Bitcoin Cash	BCH	9,186E+09	465.96	235652155
17	 NEAR Protocol	NEAR	7,872E+09	7.28	258444082
18	 Polygon	MATIC	7,011E+09	0.7076	273103291
19	 Pepe	PEPE	6,378E+09	0.00001516	1632195150
20	 Litecoin	LTC	6,242E+09	83.67	293622274
21	 Uniswap	UNI	5,797E+09	9.66	168860161
22	 Internet Computer	ICP	5,564E+09	11.98	59608138
23	 LUNA 2	LEO	5,519E+09	5.96	3024417
24	 Dai	DAI	5,349E+09	1.00	356695284
25	 Ethereum Classic	ETC	4,298E+09	29.17	176069660
26	 Aptos	APT	3,957E+09	9.05	103314330
27	 Render	RNDR	3,913E+09	10.07	158003491
28	 Hedera	HBAR	3,666E+09	0.1025	62069911
29	 Kaspa	KAS	3,563E+09	0.1496	75612129
30	 Dogwifhat	WIF	3,469E+09	3.47	702632894
31	 Filecoin	FIL	3,3E+09	5.90	149264473
32	 Cosmos	ATOM	3,299E+09	8.44	100628391
33	 Immutable X	IMX	3,285E+09	2.22	50689240
34	 Arbitrum	ARB	3,272E+09	1.13	255031614
35	 Mantle	MNT	3,181E+09	0.9746	47738064
36	 Stellar	XLM	3,096E+09	0.1067	51539887
37	 Cronos	CRO	2,975E+09	0.112	11201083
38	 Monero	XMR	2,9E+09	157.24	58058920
39	 The Graph	GRT	2,855E+09	0.3002	72022224
40	 First Digital USD	FDUSD	2,844E+09	1.00	5123516884
41	 Arweave	AR	2,842E+09	43.29	64688272
42	 OKB	OKB	2,814E+09	46.90	3935716
43	 Stacks	STX	2,781E+09	1.90	52605413
44	 Floki	FLOKI	2,672E+09	0.0002795	633292948