The SightX Research Technology Platform: Clustering methods

By
Gian Ruzzi

sightX

Contents

Chapter 1

Introduction

Clustering is a fundamental technique in exploratory data analysis, commonly employed to identify natural groupings within datasets. When analyzing survey data, which often contain a mixture of categorical and numerical variables, the selection of appropriate clustering algorithms becomes critical for uncovering meaningful respondent segments. This document provides a detailed discussion of various clustering methods that are well-suited to mixed-type survey data. We discuss the theoretical principles, preprocessing considerations, and implementation of each algorithm, with the aim of guiding researchers in selecting and applying the most effective techniques for segmenting diverse survey populations.

1.1 Definitions

Before discussing the clustering algorithms, it is necessary to define several terms and concepts that will be used throughout this document. These definitions will help ensure that the explanations and analyses in the following sections are clear and understandable.

1.1.1 Variable types

Numerical variable

A variable that represents quantitative values, such as counts or measurements, and can be either continuous (e.g., height, weight) or discrete (e.g., number of children).

Categorical variable

A variable that represents qualitative values or categories without any inherent order, such as colors, types of ice cream, or gender.

Ordinal variable

A variable that represents categories with a clear, ordered relationship among them, but where the intervals between categories are not necessarily equal, such as rating scales (e.g., agree, neutral, disagree) or educational levels.

1.1.2 Summary statistics

Mean

The mean, also called the average, is a measure of central tendency that is calculated with the formula:

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \tag{1.1}$$

For example, if we have the dataset:

$$\mathbf{x} = [4, 1, 2, 5, 2]$$

the mean is:

$$\mathbf{x} = \frac{4+1+2+5+2}{5} = \frac{14}{5} = 2.8$$

Median

The median is another measure of central tendency; it is the value that separates a dataset into two equal halves, such that half the data points are less than or equal to the median and half are greater than or equal to it. In other words, it is the middle value when the data are arranged in ascending order. If the dataset has an even number of values, the median is the average of the two central numbers. For example, if we have the dataset:

$$\mathbf{x} = [4, 1, 2, 5, 2]$$

ordering in ascending order gives us:

$$\mathbf{x} = [1, 2, 2, 4, 5]$$

as such, the median is 2. And if we have the data set

$$\mathbf{x} = [1, 2, 2, 4, 4, 5]$$

the median is $(2+4)/2 = 3$.

Standard deviation

The standard deviation is a measure of how spread out the values in a dataset are around the mean. It quantifies the typical distance of each data point from the mean value. A low standard deviation indicates that most values are close to the mean, while a high standard deviation suggests that the values are more widely dispersed. For a dataset with mean μ, the standard deviation is calculated as:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{1.2}$$

If we have the dataset:

$$\mathbf{x} = [4, 1, 2, 5, 2]$$

then

$$\sigma = \sqrt{\frac{(4-2.8)^2 + (1-2.8)^2 + (2-2.8)^2 + (5-2.8)^2 + (2-2.8)^2}{5}} = \sqrt{\frac{10.8}{5}} = 1.47$$

Interquartile range (IQR)

The interquartile range (IQR) is another measure of dispersion in a dataset. It represents the range within which the central 50% of the data lie. It is calculated as the difference between the third quartile (Q_3) and the first quartile (Q_1):

$$\text{IQR} = Q_3 - Q_1.$$

The first quartile (Q_1) is the value below which 25% of the data fall, representing the 25th percentile. The third quartile (Q_3) is the value below which 75% of the data fall, corresponding to the 75th percentile. Calculating quartiles can vary depending on the chosen method. Here, we define Q_1 as the median of the lower half of the data and Q_3 as the median of the upper half.

For example, consider the dataset:

$$\mathbf{x} = [1, 2, 2, 4, 5]$$

The lower half is $[1, 2]$, so the median of the lower half is 1.5, which is Q_1. The upper half is $[4, 5]$, so the median of the upper half is 4.5, which is Q_3. And IQR $= 4.5 - 1.5 = 3$.

If the dataset is:

$$\mathbf{x} = [1, 2, 2, 4, 4, 5]$$

then the lower half is $[1, 2, 2]$, with median 2 (Q_1), and the upper half is $[4, 4, 5]$, with median is 4 (Q_3). And IQR $= 4 - 2 = 2$

1.1.3 Normal distribution

A normal distribution is a continuous probability distribution commonly used to model real-valued random variables, such as the heights of individuals in a population. Its probability density function is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)} \tag{1.3}$$

Here, μ is the mean of the distribution, and σ is the standard deviation. The normal distribution is characterized by its symmetric, bell-shaped curve centered at the mean, with most values falling close to the mean and probabilities decreasing as values move further away. The following figure illustrates the typical shape of a normal distribution:

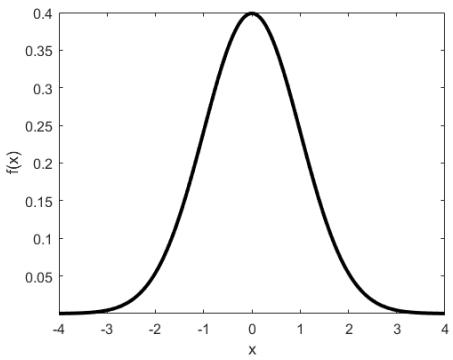

Figure 1.1: Normal probability distribution for $\mu = 0$ and $\sigma = 1$

1.1.4 Correlation

Correlation is a statistical measure that quantifies the strength and direction of a linear relationship between two variables. It reflects how changes in one variable are associated with changes in another. The most widely used measure is the Pearson correlation coefficient, which ranges from -1 to 1:

- A coefficient close to 1 indicates a strong positive relationship (as one variable increases, so does the other).
- A coefficient close to -1 indicates a strong negative relationship (as one variable increases, the other decreases).
- A coefficient near 0 indicates little or no linear relationship between the variables.

It is important to note that correlation measures association, not causation.

The Pearson correlation coefficient for n paired samples, x and y, is calculated as

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1.4}$$

where $\bar{x}$ and $\bar{y}$ are the mean values of x and y, respectively.

1.1.5 Euclidean distance

Euclidean distance measures the straight-line, or shortest possible, distance between two points in Euclidean space. For two points $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$, the distance between them is calculated as:

$$d_{xy} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \tag{1.5}$$

This is the most common method of measuring the distance between two numerical vectors.

1.1.6 Gower similarity coefficient

The Gower similarity coefficient quantifies the similarity between two data points that may include numerical, categorical, or ordinal variables. The coefficient ranges from 0, indicating complete dissimilarity, to 1, indicating identical data points. For two points i and j with v features, the similarity coefficient is defined as [1]:

$$S_{ij} = \frac{\sum_{k=1}^{v} w_k s_{ijk}}{\sum_{k=1}^{v} \delta_{ijk} w_k} \tag{1.6}$$

here, δ_{ijk} equals 1 if feature k can be compared between i and j, and 0 otherwise. The partial similarity s_{ijk} equals 1 if the two values fully agree, 0 if they are completely different, and may take an intermediate value for partial agreement. The weight w_k allows to adjust the contribution of each feature to the overall similarity. The partial similarity scores s_{ijk} are assigned as follows:

- **For categorical variables** we set $s_{ijk} = 1$ if the values of the kth feature are identical for data points i and j, and 0 otherwise.
- **For numerical and ordinal variables** let the values of feature k across all n data points be $x_1, x_2, \ldots, x_n$ (with ordinal categories assigned corresponding numerical values, e.g., disagree = 1, neutral = 2, agree = 3). The partial similarity is defined as

$$s_{ijk} = 1 - \frac{|x_i - x_j|}{R_k} \tag{1.7}$$

  where R_k is the range feature k can take.

To better illustrate the computation of the Gower similarity coefficient, consider the following example survey with three questions:

1. What is your favorite ice cream flavor?
   (Options: Chocolate, Vanilla, Strawberry, Peach)
2. How much do you agree with following statement?
   "It is important to me that the ice cream I consume is made from naturally sourced ingredients."
   (Response options: Completely disagree - Disagree - Neutral - Agree - Completely Agree)
3. How many days a week do you eat ice cream?

We got three responses:

Respondent	Question 1	Question 2	Question 3
1	Chocolate	Agree	2
2	Chocolate	Disagree	1
3	Vanilla	Completely agree	4

For question 2, the response options are numerically coded as follows:
Completely disagree = 1, Disagree = 2, Neutral = 3, Agree = 4, Completely Agree = 5.
The range of values for this question is $R_2 = 5 - 1 = 4$, as the scale spans from 1 to 5.

For question 3, the range of values is $R_3 = 7 - 0 = 7$, with possible responses from 0 (never) to 7 (every day).

Between respondents 1 and 2 the partial similarity scores are

$$\begin{aligned}
s_{1,2,1} &= 1, \quad \text{because both respondents answered, Chocolate,} \\
s_{1,2,2} &= 1 - \frac{|4-2|}{4} = 1 - 0.5 = 0.5, \\
s_{1,2,3} &= 1 - \frac{|2-1|}{7} = 1 - 0.143 = 0.857,
\end{aligned}$$

We set $w_k = 1$ for all questions, as such the similarity coefficient between respondents 1 and 2 is

$$S_{12} = \frac{1 + 0.5 + 0.847}{1 + 1 + 1} = \frac{2.347}{3} = 0.782$$

And between respondents 1 and 3:

$$\begin{aligned}
s_{1,3,1} &= 0, \quad \text{because respondents chose different flavors,} \\
s_{1,3,2} &= 1 - \frac{|4-5|}{4} = 1 - 0.25 = 0.75, \\
s_{1,3,3} &= 1 - \frac{|2-4|}{7} = 1 - 0.286 = 0.714, \\
S_{13} &= \frac{0 + 0.75 + 0.714}{1 + 1 + 1} = \frac{1.464}{3} = 0.488
\end{aligned}$$

From these scores, we can say that respondent 1 si more similar to respondent 2 than to respondent 3, because $S_{12} > S_{13}$.

Chapter 2

Data preprocessing

Before applying clustering analysis to survey data, it is important to consider data preprocessing. This step ensures that all variables, regardless of their type or scale, are comparable and contribute appropriately to the analysis. In particular, normalizing or standardizing the data ensures that differences in scale or measurement units do not cause certain variables to dominate the clustering, leading to biased or artificial groupings. Additional preprocessing steps, such as encoding categorical variables, binning continuous variables into discrete categories, and detecting or handling outliers, further enhance the quality of the input data. By appropriately preparing the data, we improve the accuracy and interpretability of the clustering results, allowing for more meaningful segmentation of survey respondents.

2.1 Handling missing data

Missing data is a common challenge in survey datasets, as respondents may skip questions or questions may be conditionally omitted. Effectively addressing missing values is crucial, as they can impact the reliability of the analysis. Common strategies for handling missing data include imputation techniques that estimate and fill in missing values, as well as the removal of incomplete data points or variables. The most appropriate approach depends on the pattern and extent of the missing data, as well as the specific objectives of the analysis.

2.1.1 Remove incomplete records

The simplest and most common approach to handling incomplete records is to exclude them from the analysis. While this method is convenient, it is generally appropriate only when the proportion of missing data is small relative to the overall dataset. Otherwise, removing incomplete records can result in the loss of valuable information, a significant reduction in sample size, and a decrease in the accuracy or representativeness of the results [2].

2.1.2 Imputation

Imputation is the process of replacing missing values in a dataset with estimated or substituted values. While a common solution to incomplete data is simply to remove records with missing entries, this can lead to the loss of valuable information and a reduced sample size. Imputation addresses this issue by enabling analysts to retain incomplete records, thereby preserving more

data for analysis and enhancing the quality of insights that can be drawn. However, it is important to use imputation carefully, as excessive or inappropriate application can introduce bias and potentially distort clustering results.

Mean or Median imputation

Missing numerical values are replaced with the mean or median of the observed values in that variable.

Mode imputation

For categorical variables, missing values are replaced with the most frequent chosen option (mode).

Random Hot Deck Imputation

This imputation method involves identifying individuals who have similar responses on other variables to the respondent with the missing value, and then randomly selecting one of these individuals' observed values to impute the missing entry. The pool of potential candidates can be defined using demographic variables or other relevant characteristics, ensuring that the imputed value comes from a comparable context.

K-Nearest Neighbors Hot Deck Imputation

This approach is a variation of hot deck imputation that leverages similarity more systematically. Instead of randomly choosing from all similar individuals, we first identify the k nearest neighbors to the individual with missing data, based on a suitable distance metric such as the Gower distance (which accommodates both nominal and numeric variables). The imputed value is then chosen from these k closest entries: for nominal variables, the most frequent value among the neighbors is used, while for numerical variables, the average of the neighbors' values is taken [2].

Cold deck imputation

This method involves imputing missing values using data from an external, but comparable, dataset rather than from within the current dataset. The external source (the "cold deck") often originates from a previous study, another sample, or a relevant database that contains values for the variables with missing data. The effectiveness of cold deck imputation relies on the similarity between the external and current datasets; if the external data are not sufficiently comparable, this approach can introduce bias into the analysis.

2.2 Normalization and Standardization

Normalization and standardization are necessary preprocessing steps in preparing survey data for cluster analysis. These techniques change the values of numerical variables so that they are on a similar scale, even if the original questions used different ranges or units. For example, one survey question might have answers ranging from 1 to 5, while another might range from 0 to 100. By putting all variables on a comparable scale, normalization and standardization prevent

variables with larger ranges from having a bigger impact on the clustering results. This section explains the main concepts and practical methods for applying normalization and standardization to survey data. To illustrate these techniques, consider the following example: suppose we asked ten people, "On a scale from 1 to 10, how much do you like ice cream?" where 10 means "I like it very much." The responses are as follows:

$$\mathbf{x} = [5, 7, 6, 4, 1, 10, 1, 8, 6, 9]$$

2.2.1 Min-Max normalization

This technique rescales variables to fit within the [0,1] range. Each value is transformed according to the formula:

$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{2.1}$$

In our example, where $\max(x) = 10$ and $\min(x) = 1$, the normalized responses are:

$$\mathbf{x}' = [0.44, 0.67, 0.56, 0.33, 0, 1, 0, 0.78, 0.56, 0.89]$$

This technique maintains the proportional relationships between the original values, and it is useful when combining variables that have different ranges or units of measurement.

2.2.2 Z-score standardization

This technique centers the data by subtracting the mean μ and scales it by dividing by the standard deviation σ. As a result, each variable is transformed to have a mean of 0 and a standard deviation of 1. Each value is transformed according to the formula:

$$x'_i = \frac{x_i - \mu}{\sigma} \tag{2.2}$$

In our example,

$$\begin{aligned}
&\mu = 5.7 \\
&\sigma = 2.9 \\
&\mathbf{x}' = [-0.24, 0.45, 0.1, -0.59, -1.62, 1.48, -1.62, 0.79, 0.1, 1.14]
\end{aligned}$$

This technique is useful when your data follows a bell-shaped (normal) distribution and the variables have different ranges or units of measurement.

2.2.3 Max absolute scaling

This technique rescales variables to fit within the [-1,1] range by dividing by the maximum absolute value. Each value is transformed according to the formula:

$$x'_i = \frac{x_i}{\max(|x|)} \tag{2.3}$$

For example, if we have the following dataset:

$$\mathbf{x} = [-5, 7, 6, 4, -1, -10, 1, -8, 6, 9],$$

here, the maximum absolute value is $|-10| = 10$, as such the normalized data set is:

$$\mathbf{x} = [-0.5, 0.7, 0.6, 0.4, -0.1, -1, 0.1, -0.8, 0.6, 0.9]$$

This technique preserves sparsity, making it suitable for data that is already centered at zero and contains a large proportion of zero or null values.

2.2.4 Robust scaling

This technique centers the data by subtracting the median and scales it by dividing by the interquartile range (IQR), making it less sensitive to outliers. Each value is transformed using the formula:

$$x_i' = \frac{x_i - \text{median}(\mathbf{x})}{\text{IQR}(\mathbf{x})} \tag{2.4}$$

For our example, consider the sorted dataset:

$$\mathbf{x} = [1, 1, 4, 5, 6, 6, 7, 8, 9, 10]$$

as such,

$$\begin{aligned}
&\text{median} = 6 \\
&\text{IQR} = 8 - 4 = 4 \\
&\mathbf{x}_i' = [-1.25, -1.25, -0.5, -0.25, 0, 0, 0.25, 0.5, 0.75, 1]
\end{aligned}$$

This scaling method is especially useful for data containing outliers, which can disproportionately affect results when using Z-score or Min-Max scaling.

2.3 Coding of ordinal variables

When working with ordinal variables, it is important to assign numerical values that reflect the inherent order of the categories. This can be achieved by mapping the categories to a uniformly spaced scale, or by applying a custom scale that captures the perceived distances between categories as appropriate for the analysis.

2.3.1 Uniform scale

The simplest and most common approach to coding an ordinal variable is to assign integer values in ascending order, ensuring equal spacing between categories. For example,

Completely disagree = 1, Disagree = 2, Neutral = 3, Agree = 4, Completely agree = 5

Fractional values may also be used, as long as the difference between adjacent categories remains consistent. The key requirement is that the intervals between codes accurately reflect the uniform progression of the ordinal scale.

2.3.2 Custom scale

A custom scale for ordinal variables is created by assigning numerical values to each category according to the actual or perceived distances between them, rather than using equally spaced values. This approach is useful when the difference in meaning or intensity between categories is not uniform, allowing the coding to more accurately represent the relationships among the categories. For example, if we have a survey question about their exercise frequency:

Option	Uniform coding	Custom coding
Never	1	0
Rarely	2	1
Sometimes	3	3
Often	4	7
Always	5	10

In the custom coding, the numerical values are assigned to reflect that the increase in frequency from “Sometimes” to “Often” (a jump from 3 to 7) is perceived as a bigger step than from “Never” to “Rarely” (from 0 to 1), or from “Rarely” to “Sometimes” (from 1 to 3). Similarly, “Always” is coded as 10 to emphasize its distinctiveness as the highest frequency. This non-uniform spacing may be based on expert judgment, empirical data, or how respondents perceive the differences between categories.

2.4 One-Hot encoding

One-hot encoding is a method used to convert categorical variables into a numerical format; it works by creating a new binary variable (column) for each possible category of the original variable. For each observation, the column corresponding to the observed category is set to 1, while all other columns are set to 0.
For instance, consider a dataset with two categorical variables:

- **Size**: Small, Medium, Large
- **Color**: Black, Red

Applying one-hot encoding produces the following representation:

Size	Color	Size_Small	Size_Medium	Size_Large	Color_Black	Color_Red
Small	Black	1	0	0	1	0
Medium	Black	0	1	0	1	0
Large	Black	0	0	1	1	0
Small	Red	1	0	0	0	1
Medium	Red	0	1	0	0	1
Large	Red	0	0	1	0	1

For example, if a respondent in our survey selects ”Medium” for size and ”Black” for color, their response would be encoded as shown in the second row above.

2.5 Variable selection and extraction

Variable selection and extraction are preprocessing steps that improve clustering by reducing dimensionality, removing noise and redundancy, and selecting the most important aspects of your survey data. Variable selection involves choosing the most relevant variables from the original dataset, while variable extraction creates new variables by transforming or combining existing ones.

2.5.1 Variability analysis

This technique consists on excluding variables whose variability is below a certain predefined threshold. Variables with very low variability (i.e., variables that do not change much across observations) are considered unlikely to be informative for clustering and should be excluded.

Numerical variables

For numerical variables we use the Quartile Coefficient of Dispersion, QCD, is defined as,

$$QCD = \frac{Q_3 - Q_1}{Q_3 + Q_1} \tag{2.5}$$

where Q_1 and Q_3 are the first and third quartiles. We will consider $QCD < 0.1$ to indicate low variability.

Categorical variables

For categorical variables we will use the proportions of the individual options, if any of the proportions is higher than 90% we consider the variable to have low variability.

Numerical variables

For numerical variables

2.5.2 Correlation analysis

This technique identifies variables that are highly correlated with each other, as such variables tend to provide redundant information. Retaining only one feature from each group of highly correlated variables improves computational efficiency and helps to ensure that clustering algorithms do not assign disproportionate weight to duplicated patterns, which could bias cluster formation. Furthermore, a less redundant dataset enhances the interpretability of the results by clarifying which features are driving the clustering.

Correlation between numerical variables

The correlation between numerical variables can be measured using Pearson's correlation coefficient squared. Between variables x and y it is defined as

$$r_{x,y} = \frac{\left(\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})\right)^2}{\left(\sum_{i=1}^{N}(x_i-\bar{x})^2\right)\left(\sum_{i=1}^{N}(y_i-\bar{y})^2\right)} \tag{2.6}$$

where $\bar{x}$ and $\bar{y}$ are the mean values of x and y, respectively. x_i and y_i are the ith observations of variable x and y, respectively.
If $r_{x,y} > 0.85$ then we consider that the variables are highly correlated.

Correlation between categorical variables

The correlation between categorical variables can be measured using Cramer's V.
Consider two categorical variables A and B, observed jointly in a sample of size N. Let A have n_A categories, indexed by j, and B have n_B categories, indexed by k. Define $n_{j,k}$ as the number of observations in which A takes its jth category (A_j) and B takes its kth category (B_k) simultaneously. Let n_j and n_k denote the total number of observations in which A_j and B_k occur, respectively.
The chi-squared statistic is given by

$$\chi^2 = \sum_{j,k} \frac{\left(n_{j,k} - \frac{n_j n_k}{N}\right)^2}{\frac{n_j n_k}{N}} \tag{2.7}$$

Cramer's V is then defined as

$$V_{x,y} = \sqrt{\frac{\chi^2/n}{\min(n_A-1, n_B-1)}} \tag{2.8}$$

If $V > 0.7$ then we consider that the variables are highly correlated.

Correlation between a numerical and categorical variable

The correlation between a categorical variable and a numerical variable can be measured using the effect size η^2. Given a categorical variable A with k categories and a numerical variable X jointly observed in a sample of size N, let A_i denote the ith category of A, and x_j be the jth observation of X. Then, η^2 is defined as:

$$\eta^2 = \frac{\sum_{i}^{k} n_i(\bar{x}_i-\bar{x})^2}{\sum_{j}^{N}(x_j-\bar{x})} \tag{2.9}$$

where:

- n_i is the number of observations in category A_i

- $\bar{x}_i$ is the mean of X for category A_i
- $\bar{x}$ is the overall mean of X
- x_j is the jth observation of X

If $\eta^2 > 0.85$ then we consider that the variables are highly correlated.

2.5.3 Principal Component Analysis (PCA)

While removing correlated variables is a common way to reduce redundancy, another effective approach is to apply Principal Component Analysis. PCA is a technique used to extract new features from numerical data. It achieves this by combining the original variables into a smaller set of uncorrelated components, called principal components, which represent most of the variability in the dataset. These components are ordered by the amount of variance they capture, with the first components retaining the most significant information. This process reduces the dimensionality of the data while preserving its essential structure [3].

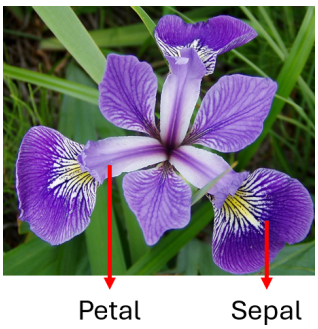

Figure 2.1: Iris versicolor flower morphology with labels for petal and sepal. Source: [4]

PCA is best understood through a practical example. Consider the well-known Iris flower dataset, introduced by the British statistician and biologist Ronald Fisher in his 1936 paper, *"The Use of Multiple Measurements in Taxonomic Problems"*. This dataset consists of 150 samples from three species of Iris flowers (*Iris setosa*, *Iris versicolor*, and *Iris virginica*); see Figure 2.1. Each sample is characterized by four numerical features: sepal length, sepal width, petal length, and petal width, all measured in centimeters. The Iris dataset is widely used in statistics and machine learning as a benchmark for classification, clustering, and visualization techniques. For visualization purposes, we will use only sepal length, petal width, and petal length. In Figure 2.2 the data points are plotted in three dimensions, with each point colored according to its species.

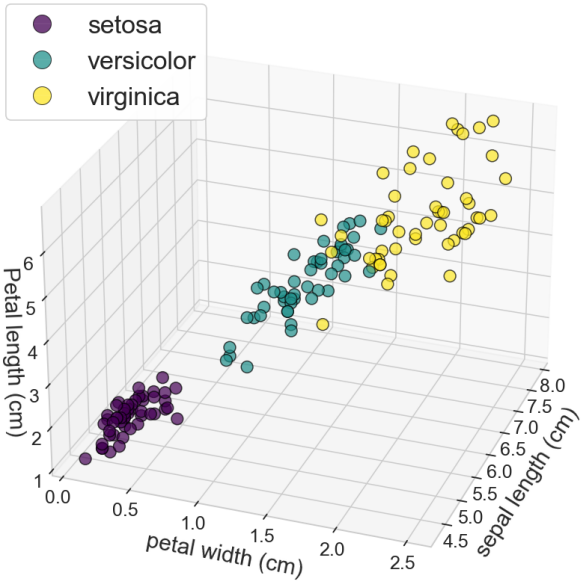

Figure 2.2: Plot of the iris dataset

In this plot, we observe that the data points are largely distributed along a plane that is diagonal to the coordinate axes. This suggests that we can fit a plane to the data and project the points onto it, effectively capturing the most significant structure while reducing the dimensionality from three to two. This dimensionality reduction can be achieved using PCA. We follow these steps

Standardize the data

The first step is to standardize the data using Z-score standardization (see Section 2.2.2). This is necessary because PCA is sensitive to the scale of the variables; standardization ensures that all variables contribute equally to the analysis.

Calculate the covariance matrix of the data

Next, we compute the covariance matrix of the standardized data using the formula:

$$C = \frac{1}{N-1} X^T X \tag{2.10}$$

where N is the number of observations (in this case, 150) and X is the data matrix, with each row representing a single observation and each column representing a variable. For our example, X will be a 150×3 matrix corresponding to sepal length, petal width, and petal length. The covariance matrix is

$$\mathbf{C} = \begin{bmatrix} 1.006711 & 0.823431 & 0.877604 \\ 0.823431 & 1.006711 & 0.969328 \\ 0.877604 & 0.969328 & 1.006711 \end{bmatrix}$$

Calculate the eigenvalues and eigenvector of C

The eigenvectors of **C** define the direction (principal components), and eigenvalues tell how much variance there is in each direction. We solve the characteristic equation

$$\det(\mathbf{C} - \lambda \mathbf{I}) = \mathbf{0} \tag{2.11}$$

The eigenvectors of our example are:

$$\mathbf{v}_1 = \begin{bmatrix} 0.559641 \\ 0.580468 \\ 0.591489 \end{bmatrix} \qquad \mathbf{v}_2 = \begin{bmatrix} 0.812704 \\ -0.524106 \\ -0.254606 \end{bmatrix} \qquad \mathbf{v}_3 = \begin{bmatrix} -0.162212 \\ -0.623194 \\ 0.765060 \end{bmatrix}$$

and the eigenvalues:

$$\lambda_1 = 2.788330 \qquad \lambda_2 = 0.200750 \qquad \lambda_3 = 0.031054$$

Select the Principal Components

To select the principal components, we first sort the eigenvectors in descending order based on their corresponding eigenvalues. The eigenvector associated with the largest eigenvalue becomes the first principal component, the second largest corresponds to the second component, and so on. We retain the components with the highest eigenvalues because these account for most of the variance in the data. In our example, the total variance is

$$\sigma^2 = \lambda_1 + \lambda_2 + \lambda_3 = 3.020134$$

The proportion of variance explained by each principal component is calculated as

$$\text{Component 1 :} \quad \frac{2.788330}{3.020134} = 0.92324 \equiv 92.32\%$$

$$\text{Component 2 :} \quad \frac{0.200750}{3.020134} = 0.06647 \equiv 6.65\%$$

$$\text{Component 3 :} \quad \frac{0.031054}{3.020134} = 0.01028 \equiv 1.03\%$$

As shown, the first component accounts for the vast majority of the variance in our sample. If we used only this component, the data would be projected onto a straight line. To achieve dimensionality reduction while retaining most of the information, we select the first and second components and represent the data in two dimensions.

Project Data onto Principal Components

To project the data onto the selected principal components, we multiply the original (standardized) data matrix X by the matrix whose columns are the chosen eigenvectors:

$$X' = X[v_1, v_2, \ldots, v_n]$$

In our example, we use the first two principal components, so X is multiplied by:

$$\begin{bmatrix} 0.559641 & 0.812704 \\ 0.580468 & -0.524106 \\ 0.591489 & -0.254606 \end{bmatrix}$$

This projection reduces the original data to a two-dimensional space, as illustrated in Figure 2.3. In this figure, it can be seen that the transformation preserves the most significant structure of the dataset while reducing its dimensionality by one. The transformed dataset will be used in subsequent clustering analysis.

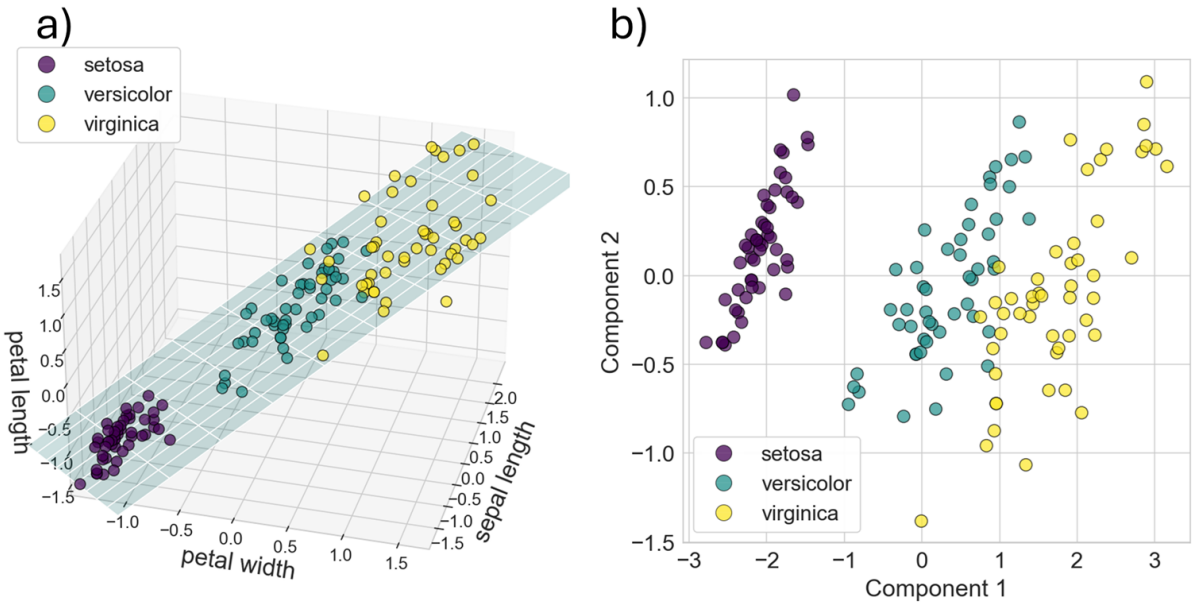

Figure 2.3: a) Standardized Iris dataset visualized in three dimensions (sepal length, petal width, and petal length), with the first two principal components represented as a fitted plane. b) Projection of the same data onto the two-dimensional space defined by these principal components, showing that the structure and separation among the three Iris species are preserved in the reduced space.

2.5.4 Multiple Correspondence Analysis (MCA)

MCA is a technique used to reduce the dimensionality of datasets that contain multiple categorical variables. It transforms categorical data into a smaller set of continuous variables, or components, that summarize the main ways the data varies. In this sense, MCA can be viewed as a generalization of principal component analysis (PCA) for categorical data rather than quantitative data. By preserving the most important relationships among variables, MCA makes it easier and more effective to apply clustering algorithms, helping to reveal more meaningful groupings and underlying structures within the dataset.

MCA is best understood through a practical example. Consider the following data from a survey about leisure preferences

Respondent	Favorite Sport	Preferred Genre	Weekend Activity	Snack Choice	Streaming Service
1	Soccer	Action	Reading	Popcorn	Netflix
2	Tennis	Comedy	Hiking	Chips	Hulu
3	Basketball	Drama	Movies	Fruit	HBO
4	Soccer	Comedy	Hiking	Chips	Netflix
5	Basketball	Action	Reading	Fruit	Amazon Prime
6	Tennis	Drama	Hiking	Popcorn	Hulu
7	Soccer	Action	Movies	Chips	HBO
8	Tennis	Drama	Reading	Fruit	Amazon Prime
9	Basketball	Comedy	Reading	Popcorn	Netflix
10	Soccer	Drama	Movies	Chips	Hulu

Here, we refer to the column names as *categories* and to the possible values within each category as *levels*. For example, "Favorite Sport" is a category, and "Soccer," "Tennis," and "Basketball" are its levels. To apply MCA to our data we follow these steps [5]:

One-hot encode the data

First, we need to one-hot encode our data, see Section 2.4. For our example, we get:

Res.	Soccer	Tennis	Basketball	Action	Comedy	Drama	Reading	Hiking	Movies	Popcorn	Chips	Fruit	Netflix	Hulu	HBO	Amazon
1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
2	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
3	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	0	0	0	1	0	0	1	0	0	1	0	1	0	0	0
5	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	1
6	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0
7	1	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0
8	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1
9	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0	0
10	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0

This table is called the *Indicator matrix*.

Center and normalize our data

Calculate the probability matrix $\mathbf{Z}$

$$\mathbf{Z} = N^{-1}\mathbf{X} \tag{2.12}$$

where N is the sum of all the entries in the indicator matrix, and $\mathbf{X}$ is the inidicator matrix

$$\mathbf{Z} = \begin{bmatrix}
0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 \\
0.02 & 0.00 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.02 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.02 \\
0.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.02 & 0.00 & 0.02 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 \\
0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.02 & 0.00 \\
0.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.02 & 0.02 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.02 \\
0.00 & 0.00 & 0.02 & 0.00 & 0.02 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 \\
0.02 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.02 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00
\end{bmatrix}$$

Let $\mathbf{r}$ denote the column vector of row totals of $\mathbf{Z}$ and $\mathbf{c}$ denote the columns vector of column totals of $\mathbf{Z}$. Define $\mathbf{D}_r$ as the diagonal matrix whose diagonal entries are the elements of $\mathbf{r}$, and similarly,

define $\mathbf{D}_c$ as the diagonal matrix whose diagonal entries are the elements of $\mathbf{c}$. With these, we compute

$$\mathbf{M} = {\mathbf{D_r}}^{-\frac{1}{2}} \left(\mathbf{Z} - \mathbf{r}\mathbf{c}^T \right) {\mathbf{D_c}}^{-\frac{1}{2}} \tag{2.13}$$

For our example, we get

$$\mathbf{M} = \begin{bmatrix}
0.13 & -0.08 & -0.08 & 0.18 & -0.08 & -0.09 & 0.13 & -0.08 & -0.08 & 0.18 & -0.09 & -0.08 & 0.18 & -0.08 & -0.06 & -0.06 \\
-0.09 & 0.18 & -0.08 & -0.08 & 0.18 & -0.09 & -0.09 & 0.18 & -0.08 & -0.08 & 0.13 & -0.08 & -0.08 & 0.18 & -0.06 & -0.06 \\
-0.09 & -0.08 & 0.18 & -0.08 & -0.08 & 0.13 & -0.09 & -0.08 & 0.18 & -0.08 & -0.09 & 0.18 & -0.08 & -0.08 & 0.25 & -0.06 \\
0.13 & -0.08 & -0.08 & -0.08 & 0.18 & -0.09 & -0.09 & 0.18 & -0.08 & -0.08 & 0.13 & -0.08 & 0.18 & -0.08 & -0.06 & -0.06 \\
-0.09 & -0.08 & 0.18 & 0.18 & -0.08 & -0.09 & 0.13 & -0.08 & -0.08 & -0.08 & -0.09 & 0.18 & -0.08 & -0.08 & -0.06 & 0.25 \\
-0.09 & 0.18 & -0.08 & -0.08 & -0.08 & 0.13 & -0.09 & 0.18 & -0.08 & 0.18 & -0.09 & -0.08 & -0.08 & 0.18 & -0.06 & -0.06 \\
0.13 & -0.08 & -0.08 & 0.18 & -0.08 & -0.09 & -0.09 & -0.08 & 0.18 & -0.08 & 0.13 & -0.08 & -0.08 & -0.08 & 0.25 & -0.06 \\
-0.09 & 0.18 & -0.08 & -0.08 & -0.08 & 0.13 & 0.13 & -0.08 & -0.08 & -0.08 & -0.09 & 0.18 & -0.08 & -0.08 & -0.06 & 0.25 \\
-0.09 & -0.08 & 0.18 & -0.08 & 0.18 & -0.09 & 0.13 & -0.08 & -0.08 & 0.18 & -0.09 & -0.08 & 0.18 & -0.08 & -0.06 & -0.06 \\
0.13 & -0.08 & -0.08 & -0.08 & -0.08 & 0.13 & -0.09 & -0.08 & 0.18 & -0.08 & 0.13 & -0.08 & -0.08 & 0.18 & -0.06 & -0.06
\end{bmatrix}$$

Compute left eigenvalues and eigenvectors

Compute the eigenvalues and eigenvector of $\mathbf{M}\mathbf{M}^T$, for this we solve the characteristic equation

$$\det(\mathbf{M}\mathbf{M}^T - \lambda \mathbf{I}) = \mathbf{0} \tag{2.14}$$

The eigenvectors of our example are the columns of the following matrix

$$\mathbf{U} = \begin{bmatrix}
0.06 & -0.28 & -0.45 & 0.28 & 0.48 & -0.04 & 0.14 & 0.04 & 0.53 & -0.32 \\
-0.45 & -0.08 & 0.29 & 0.03 & -0.38 & -0.34 & -0.22 & -0.29 & 0.46 & -0.32 \\
0.33 & 0.44 & 0.08 & -0.63 & -0.03 & -0.02 & 0.22 & 0.18 & 0.35 & -0.32 \\
-0.38 & -0.12 & -0.27 & 0.03 & -0.46 & 0.28 & 0.39 & 0.44 & -0.16 & -0.32 \\
0.54 & -0.17 & 0.04 & 0.25 & -0.28 & -0.21 & -0.46 & 0.40 & -0.11 & -0.32 \\
-0.28 & -0.12 & 0.44 & -0.10 & 0.53 & -0.31 & 0.09 & 0.32 & -0.35 & -0.32 \\
0.01 & 0.52 & -0.37 & 0.24 & -0.03 & -0.41 & 0.14 & -0.33 & -0.37 & -0.32 \\
0.34 & -0.14 & 0.47 & 0.30 & -0.06 & 0.33 & 0.43 & -0.38 & -0.06 & -0.32 \\
0.05 & -0.46 & -0.28 & -0.54 & 0.02 & 0.11 & -0.22 & -0.41 & -0.30 & -0.32 \\
-0.21 & 0.40 & 0.04 & 0.13 & 0.22 & 0.61 & -0.50 & 0.01 & 0.00 & -0.32
\end{bmatrix}$$

and the eigenvalues are the diagonal elements of the following matrix:

$$\Lambda = \begin{bmatrix}
0.59 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.49 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.45 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.22 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.21 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.08 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.00
\end{bmatrix}$$

Select the eigenvectors to use

We first sort the eigenvectors in descending order based on their corresponding eigenvalues. We retain the components with the highest eigenvalues because these account for most of the variance in the data. In PCA, the variance is simply the sum of the eigenvalues, here we need to correct our eigenvalues because the way we encode our data creates artificial additional dimensions since one category (e.g., Favorite Sport) is coded with several columns (e.g., 3 for Favorite Sport). As such, the variance of the resultant space is inflated, and the percentage of variance of the first dimensions is greatly underestimated [5]. We correct our eigenvalues λ_i as follows

$$\lambda_i^c = \begin{cases} \left[\left(\frac{K}{K-1}\right)\left(\lambda_i - \frac{1}{K}\right)\right]^2 & \text{if } \lambda_i > \frac{1}{K} \\ 0 & \text{if } \lambda_i \leq \frac{1}{K} \end{cases} \tag{2.15}$$

where K is the total number of categories, in our example this will be 5. Now, the proportion of variance is calculated as follows:

$$\tau_i = \frac{\lambda_i^c}{\sum_i \lambda_i^c} \tag{2.16}$$

for our example we get:

$$\begin{aligned} \tau_1 &= 51.5\% \\ \tau_2 &= 27.8\% \\ \tau_3 &= 20.6\% \\ \tau_4 &= 0.001\% \\ \tau_5 &= 0.0002\% \\ \tau_{6\ldots10} &= 0\% \end{aligned}$$

Here the first two dimensions explain about 79% of the variace so we can keep only the first two.

Selection of Eigenvectors

We begin by sorting the eigenvectors in descending order according to their associated eigenvalues. We retain the components with the largest eigenvalues, as these capture the greatest portion of variance in the data. In PCA, the total variance is simply the sum of the eigenvalues. However, in MCA, the encoding of categorical data, where each category is represented by multiple columns, creates artificial additional dimensions. For instance, a category such as *Favorite Sport* with three levels will be represented by three columns. This redundancy inflates the total variance, resulting in an underestimation of the proportion of variance explained by the first dimensions [5].
To address this, we correct the eigenvalues λ_i as follows:

$$\lambda_i^c = \begin{cases} \left[\left(\frac{K}{K-1}\right)\left(\lambda_i - \frac{1}{K}\right)\right]^2 & \text{if } \lambda_i > \frac{1}{K} \\ 0 & \text{if } \lambda_i \leq \frac{1}{K} \end{cases} \tag{2.17}$$

where K is the total number of categories; for our example, $K = 5$. The proportion of variance explained by each dimension is then calculated as:

$$\tau_i = \frac{\lambda_i^c}{\sum_i \lambda_i^c} \tag{2.18}$$

For our example, :

$$\begin{aligned}
\tau_1 &= 51.5\% \\
\tau_2 &= 27.8\% \\
\tau_3 &= 20.6\% \\
\tau_4 &= 0.001\% \\
\tau_5 &= 0.0002\% \\
\tau_{6\ldots10} &= 0\%
\end{aligned}$$

Thus, the first three dimensions capture approximately 99.9% of the total variance, and we retain only these three for further analysis.

Get the responses coordinates in the solution space

Using the selected eigenvectors, in our example the first 3, we obtain the responses coordinates following equation:

$$\mathbf{F} = \mathbf{D_r}^{-\frac{1}{2}} \mathbf{U}_s \Lambda_s^{\frac{1}{2}} \tag{2.19}$$

where $\mathbf{U}_s$ is a matrix whose columns are the selected eigenvectors, and $\Lambda_s^{\frac{1}{2}}$ is a diagonal matrix whose elements are the square root of the corresponding eigenvalues. For our example we get

$$\mathbf{F} = \begin{bmatrix}
0.14 & -0.61 & -0.95 \\
-1.10 & -0.17 & 0.62 \\
0.79 & 0.96 & 0.17 \\
-0.92 & -0.28 & -0.57 \\
1.32 & -0.37 & 0.08 \\
-0.69 & -0.26 & 0.92 \\
0.02 & 1.16 & -0.78 \\
0.83 & -0.30 & 1.00 \\
0.12 & -1.01 & -0.59 \\
-0.51 & 0.89 & 0.09
\end{bmatrix}$$

The transformed dataset, $\mathbf{F}$, will be used in subsequent clustering analysis. To illustrate the results, we display the data as a point cloud in Figure 2.4.

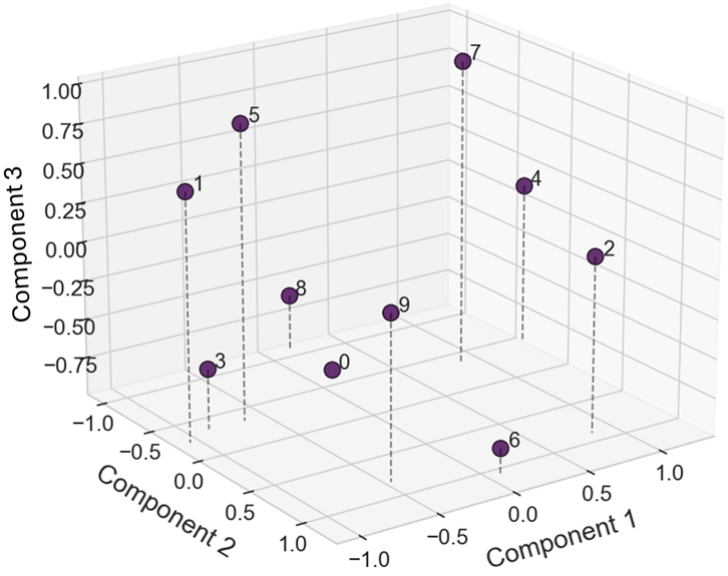

Figure 2.4: Point cloud of the leisure dataset obtained from MCA.

This figure demonstrates an additional benefit of MCA: it enables a geometric representation of categorical data, which in turn allows us to apply euclidean distances for clustering analysis.

2.6 Outlier detection and treatment

Outliers are data points that differ substantially from the majority of respondents and can result from data entry errors, misunderstandings of survey questions, or unusual but legitimate patterns in how people behave or respond. If not properly addressed, outliers can skew similarity measures, distort clustering results, and reduce the clarity and interpretability of identified segments. Careful detection and appropriate treatment of outliers help ensure that clustering algorithms yield more meaningful and reliable groupings, ultimately leading to more accurate insights from survey data.

Common methods for identifying and handling outliers include:

Z-score or Standard Deviation Threshold

Flag data points that lie beyond a specified number of standard deviations from the mean as outliers.

Interquartile Range (IQR) Threshold

Identify observations as outliers if they fall outside $1.5\times$ the IQR above the third quartile or below the first quartile.

Remove Outliers

Eliminate data points identified as outliers; this approach is often suitable when outliers are known errors or are irrelevant to the analysis. However, be cautious, as removing authentic data can

introduce bias or reduce the sample size.

Winsorization

Limit the influence of extreme values by capping them at a specified percentile (e.g., values above the 95th percentile are set equal to the 95th percentile). This retains all observations but reduces the impact of outliers.

Data Transformation

Apply transformations, such as logarithmic or square root functions, to reduce the skewness of the data and the influence of outliers, especially for highly skewed numerical variables.

Impute Outliers

Replace outlier values with a statistic such as the mean, median, or an estimate based on similar observations, thus preserving the dataset size, though this practice may introduce some bias.

2.7 Discretization or Binning

This method consists in converting continuous or ordinal numerical data into a finite number of categorical bins or intervals. For survey data, this technique is often used to simplify responses, group scores, or make the results easier to interpret. For example, age (a continuous variable) can be binned into categories such as *18-25*, *26-35*, and *36-45*, or satisfaction scores from 1 to 10 can be grouped into *Low*, *Medium*, and *High*. Binning can help reduce the effect of small variations or outliers, facilitate comparison across groups, and make clustering or segmentation analyses on survey data more robust and interpretable. However, the choice of bin edges should be made thoughtfully to preserve meaningful distinctions in the data.

Chapter 3

K-means clustering

K-means clustering is one of the most widely used algorithms for partitioning datasets into distinct groups based on similarity. The goal of clustering is to create groups that are internally cohesive and externally isolated, i.e., members of the same cluster should be as similar as possible, while members of different clusters should be as different as possible [6]. George Sebestyen (1962) and James MacQueen (1967) independently introduced the k-means method, which efficiently partitions data by seeking to minimize within-cluster variance. Since its development, k-means has become a standard approach in the literature on multivariate statistics, cluster analysis, statistical learning, and pattern recognition.

3.1 The method

The k-means clustering algorithm partitions a dataset of N samples, each described by P variables, into K clusters, where K is specified in advance. Methods for choosing an optimal K will be addressed in later chapters. The algorithm works by assigning each sample to the nearest cluster centroid, iteratively minimizing the within-cluster distances. Each centroid is the mean of its cluster's samples in P-dimensional space. I will first describe k-means for numerical variables using Euclidean distance, then discuss how it can be adapted to datasets with categorical variables.

To illustrate, consider the following example in two dimensions, where we record the weight and body length (from head to the start of the tail) of several mice:

Mouse	Length (cm)	Weight (g)
1	6.0	12.0
2	4.3	12.8
3	5.0	16.0
4	7.0	12.2
5	7.8	18.3
6	8.5	18.0
7	7.7	21.0
8	8.9	20.1
9	12.1	24.3
10	11.3	26.0
11	14.0	25.3

By plotting the data, Figure 3.1, we observe three distinct clusters. Let us now see how the k-means algorithm partitions the data when we specify $K = 3$ clusters.

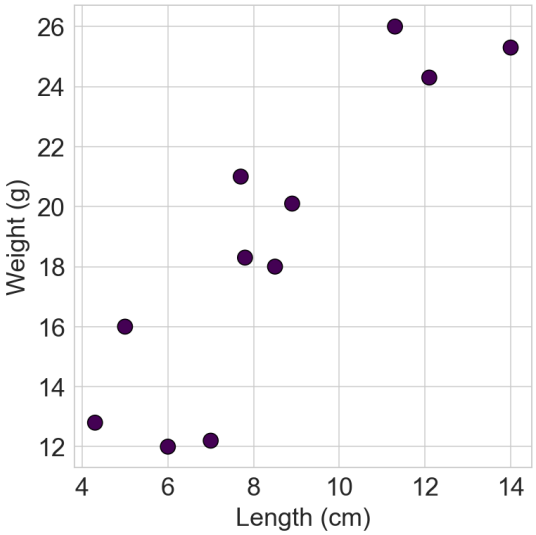

Figure 3.1: Scatter plot of mice showing the relationship between body weight and height. Each point represents an individual mouse.

Let's outline the steps involved in clustering our dataset:

3.1.1 Scale your variables

First, we need to ensure that our variables are on comparable scales. In this case, the maximum length is 14 cm while the maximum weight is 26 g, meaning weight has a greater range and may disproportionately influence the clustering results. To avoid this bias, we apply normalization or standardization as discussed in Chapter 2. Here, I will use Min-Max normalization, our normalized data looks as follows:

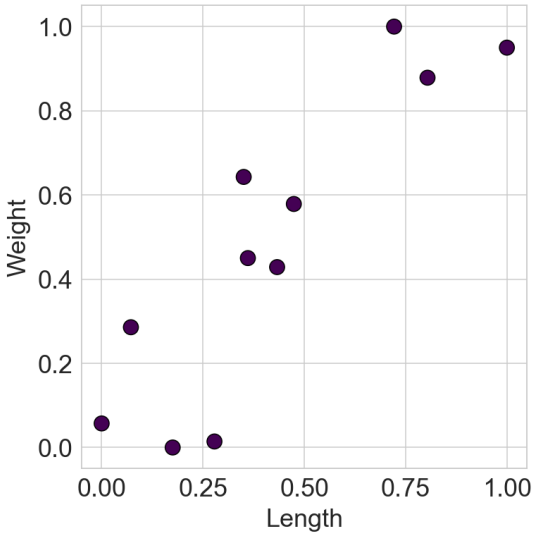

Figure 3.2: Scaled mice dataset.

3.1.2 Generate initial centroids

Given a desired number of clusters K, initial centroids can be generated in several ways:

- **Random selection:** Randomly choose K data points from the dataset to serve as the initial centroids.
- **Random partition:** Randomly assign each data point to one of the K clusters, then set each initial centroid as the mean of all points assigned to its cluster.
- **k-means++:** This widely-used method spreads initial centroids throughout the data to improve clustering results:
  1. Randomly select the first centroid from the dataset.
  2. For each data point x_i, compute $d(x_i)^2$, where $d(x_i)$ is the distance from x_i to its nearest chosen centroid.
  3. Let
  $$S = \sum_{i=1}^{N} d(x_i)^2$$
  4. Assign each data point x_i a probability
  $$P(x_i) = \frac{d(x_i)^2}{S}$$
  5. Select the next centroid at random, using this probability distribution (so that points farther from existing centroids are more likely to be chosen).
  6. Repeat steps 2–5 until K centroids have been chosen.

For our example, I use random selection for the initial centroids. The centroids are highlighted with red borders in the plot below:

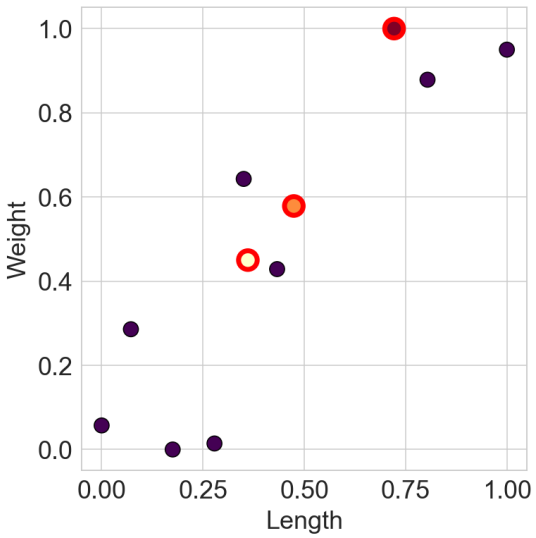

Figure 3.3: Random initial centroids for the mice dataset.

3.1.3 Assign points to their nearest centroids and update centroids

1. For each data point, we compute its Euclidean distance to each centroid and assign it to the closest one. The resulting initial cluster assignments are shown in the following plot:

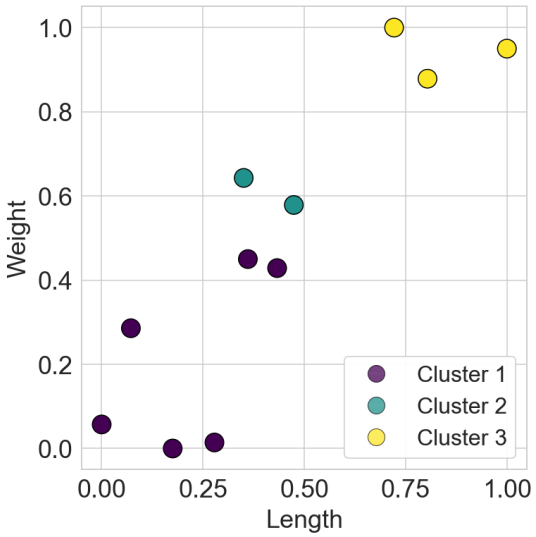

Figure 3.4: Initial clusters in the mice dataset

2. Next, the centroid of each cluster is recalculated as the mean of all data points assigned to that cluster. The updated centroids are marked by x markers in the following plot:

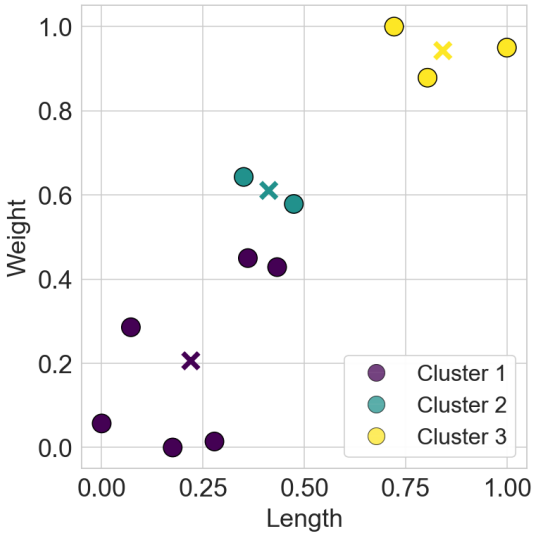

Figure 3.5: Initial clusters in the mice dataset

3. Repeat steps 1 and 2 until the centroids no longer change, or until a predefined number of iterations is reached.

In our example, we repeated steps 1 and 2 until the cluster assignments no longer changed. The final clusters are shown in the plot below:

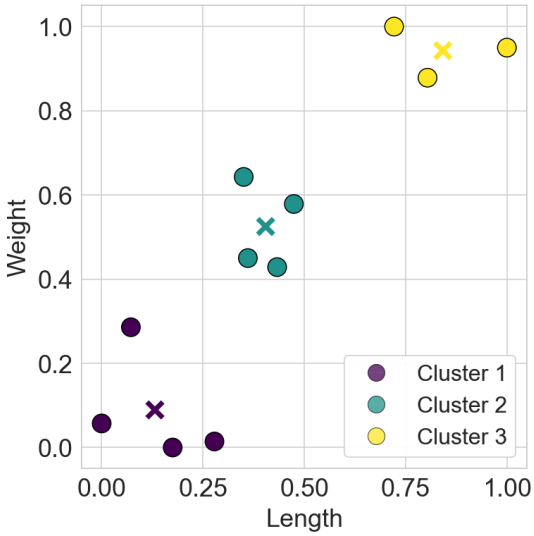

Figure 3.6: Final clusters in the mice dataset

3.2 Handling categorical data: k-prototypes clustering

Categorical data can be incorporated into clustering analyses in several ways. One approach is to use Multiple Correspondence Analysis (MCA), Section 2.5.4, to transform categorical variables into numerical features, enabling the use of Euclidean distances. Alternatively, the **k-prototypes** algorithm extends k-means to handle datasets with both numerical and categorical variables [7]. It achieves this by combining the Euclidean distance for numerical attributes and the simple matching dissimilarity (as used in k-modes [7]) for categorical attributes.

For a data point x_i and a cluster prototype (centroid) c_j, the k-prototypes dissimilarity measure combines numerical and categorical variables as follows:

$$d(x_i, c_j) = \sum_{l=1}^{p} (x_{il} - c_{jl})^2 + \gamma \sum_{l=1}^{q} \delta(x_{il}, c_{jl}), \tag{3.1}$$

where:

- p is the number of numeric variables,
- q is the number of categorical variables,
- $p+q=N$, where N is the total number of variables,
- γ is a weighting parameter that balances the influence of categorical and numerical variables,
- $\delta(a,b)$ is an indicator function comparing categorical values a and b:

$$\delta(a,b) = \begin{cases} 0, & \text{if } a = b \\ 1, & \text{if } a \neq b \end{cases}$$

In this formulation, the centroid for each cluster consists of the mean for numeric variables and the mode (most frequent category) for categorical variables. For a given categorical attribute, if a data point matches the cluster mode, its contribution to the dissimilarity is 0; otherwise, it is 1.
The k-prototypes algorithm proceeds as follows:

- Initialize K prototypes, each consisting of means for numeric variables and modes for categorical variables. The initial centroids can be selected as described in Section 3.1.2, using Equation 3.1 as the distance measure. For categorical attributes, the initial mode for each prototype is simply the category present in the selected data point.
- Assign each data point to the cluster whose prototype minimizes the dissimilarity measure above.
- Update each prototype:
    1. For numeric attributes, set the prototype to the mean of the assigned points (as in k-means).
    2. For categorical attributes, set the prototype to the mode of the assigned points (as in k-modes).
- Repeat the assignment and update steps until the prototypes do not change or until a predefined number of iterations is reached.

3.2.1 Choosing γ

The choice of γ is important to ensure that both categorical and numerical variables contribute appropriately to the clustering process. Huang [7] suggests using the standard deviation of the numeric attributes as a guide to specify γ:

$$\gamma = \frac{1}{p}\sum_{i=1}^{p}\sigma_i \tag{3.2}$$

where σ_i is the standard deviation of the ith numeric variable, and p is the number of numeric variables. Most of the time, γ will depend on the project so that the contribution from numeric and categorical variables is balanced according to what makes sense in your application (sometimes categorical matches or mismatches are more or less meaningful).

Chapter 4

Choosing the optimal number of clusters

One of the key challenges in clustering analysis is selecting the appropriate number of clusters for a given dataset. If too few clusters are chosen, distinct groups in the data may be combined, obscuring important patterns. Conversely, too many clusters can result in meaningless divisions and make interpretation more difficult. To address this, several analytical methods and heuristics have been developed to help identify a suitable value for clusters. In this section, we review some of the most common strategies, including both visual and statistical approaches.

4.1 Elbow method

The Elbow Method is a popular heuristic for estimating the optimal number of clusters, in algorithms such as k-means or k-prototypes. This method examines how the clustering cost, typically quantified by the within-cluster sum of squares (WCSS) or "inertia", changes as the number of clusters increases:

- For each candidate value of K (e.g., from 1 to 10), run the clustering algorithm and compute the total inertia. Inertia is the sum, across all clusters, of the squared distances between each point and its assigned centroid. For data containing categorical variables, use the k-prototypes algorithm and its corresponding dissimilarity measure (see Equation 3.1) to calculate the clustering cost.
- As K increases, inertia will decrease, since points are assigned to centroids that are closer. However, after a certain point, the incremental decrease in inertia becomes minimal.
- The optimal K is identified at the "elbow" point in the inertia versus K plot, where the rate of decrease sharply diminishes, indicating that adding more clusters provides little further improvement.

Consider the mice example used in Section 3.1, we run k-means clustering for different values of K and plot the inertia as a function of K:

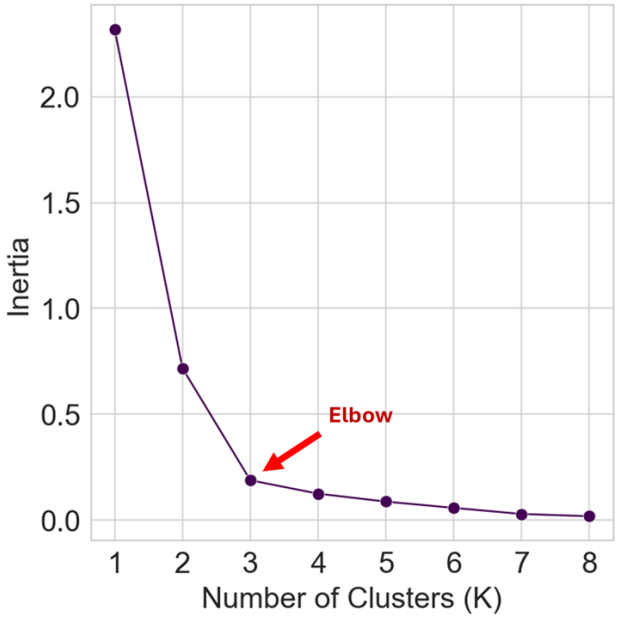

Figure 4.1: Elbow plot for the mice dataset.

As shown in the figure, the "elbow" occurs at $K = 3$, suggesting that three clusters is the optimal choice for segmenting this dataset.

4.2 Silhouette score

The silhouette score is a metric that helps evaluate how well your data points are grouped into clusters. It measures how similar each point is to other points in its own cluster compared to points in other clusters. The score ranges from -1 to 1, with higher values indicating that clusters are well separated and more cohesive, suggesting a better clustering result.

For a data set, S, with N observations, partitioned into K clusters, $C_1, C_2, \ldots C_k$. For each data point p we calculate the average distance between p and all other points that belong to the same cluster as p,

$$a(p) = \frac{\sum_{p' \in C_i, p \neq p'} d(p, p')}{|C_i| - 1} \tag{4.1}$$

where $d(p, p')$ is the distance between points p and p', and $|C_i|$ is the number of points in cluster C_i. Similarly, we calculate the minimum average distance from p to all other clusters that not contain p,

$$b(p) = \min_{C_j : 1 \leq j \leq k, j \neq i} \frac{\sum_{p' \in C_j} d(p, p')}{|C_j| - 1} \tag{4.2}$$

Thus, the silhouette coefficient of p is defined as

$$s(p) = \frac{b(p) - a(p)}{max\{a(p), b(p)\}} \tag{4.3}$$

And we define our silhouette score as the average of all the silhouette coefficients of our data set. We choose the K that gives us the largest silhouette score.

Bibliography

[1] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, vol. 27, no. 4, pp. 857–871, 1971.

[2] J. Luengo, S. García, and F. Herrera, "On the choice of the best imputation methods for missing values considering three groups of classification methods," *Knowledge and Information Systems*, vol. 32, no. 1, pp. 77–108, 2012.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, New York: Springer, 2 ed., 2009.

[4] Danielle Langlois, "Iris versicolor 3 (Wikimedia Commons)," 2005. Accessed: 2024-04-25. Licensed under CC BY-SA 3.0.

[5] H. Abdi and D. Valentin, "Multiple correspondence analysis," *Encyclopedia of Measurement and Statistics*, pp. 651–657, 2007.

[6] D. Steinley, "K-means clustering: A half-century synthesis," *British Journal of Mathematical and Statistical Psychology*, vol. 59, no. 1, pp. 1–34, 2006.

[7] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, 1998.