UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

Faculty of Science

Master of Science in Mathematical Statistics Thesis

Optimising Visual Clarity using Clustering Techniques for Overcrowded Biplots

Yamkela Balisa (1918991)

Supervisor: Dr Raeesa Ganey

09 June 2025

Plagiarism Declaration

I hereby declare that:

1. Plagiarism is the use of ideas, material, and other intellectual property of another's work and to present it as my own.
2. I agree that plagiarism is a punishable offence because it constitutes theft.
3. Accordingly, all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.
4. I also understand that direct translations are plagiarism.
5. I declare that the work contained in this assignment, except otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this assignment or another assignment.

I have read and understood the declaration above.

Initials and surname: Y. Balisa Date: 28 February 2025

Acknowledgements

I am deeply grateful to God for the strength, health, and determination that allowed me to complete this dissertation. Balancing a full-time job and new team responsibilities while pursuing my masters degree was a daunting task. However, through diligent effort, careful time management, and sacrifices, including countless late nights and working on weekends, I was able to overcome the challenges and achieve my goal.

I also express my sincere gratitude to my supervisor, Dr Raeesa Ganey, whose guidance, expertise, and encouragement have been invaluable throughout the course of this dissertation. The constructive feedback and insightful suggestions provided at each stage of my work have helped shape this dissertation into its present form. I greatly appreciate the patience and mentorship provided, which extended beyond academic advice and has been a source of inspiration throughout my academic journey.

Furthermore, I would like to thank the University of Witwatersrand, specifically the School of Statistics and Actuarial Sciences, for providing me with the opportunity to pursue my masters degree and for providing the necessary resources to make this study possible.

Lastly, I would like to express my heartfelt gratitude to my closest supporters, Thulisa Bongo, Sandile Shongwe, and Sikelela Balisa. Your unwavering support throughout this journey meant everything to me. This achievement would not have been possible without you—thank you so much!

Abstract

The increasing use of data in various industries has driven the need for effective data analysis and visualisation. Data visualisation is a key methodology for extracting insights from the data. One powerful visualisation technique based on dimensionality reduction methods is the biplot. Biplots are multivariate scatterplots that facilitate the visualisation of high-dimensional data by projecting it onto lower dimensional spaces, usually two or three dimensions. This reduction in dimensionality is achieved using techniques such as Principal Component Analysis (PCA) for continuous data. A biplot simultaneously represents both samples and variables within the same visualisation.

However, biplots often face challenges when dealing with a very large number of variables in data. A key issue is the overcrowding of variables within the biplot, making it difficult to obtain meaningful insights. To address this issue, this study explores the integration of unsupervised learning techniques, specifically clustering into the biplot framework. Unsupervised learning refers to a type of machine learning approach in which the algorithm learns patterns and relationships in the data without prior knowledge of the expected output. Clustering, a fundamental unsupervised learning technique, involves grouping similar data points into clusters, enabling the identification of underlying structures and relationships. By applying clustering, specifically the k-means clustering algorithm, this study aims to cluster similar variables into distinct clusters within the biplot. Similar variables are determined by the proximity of their endpoints and the angles they form within the biplot. Ultimately, the refined biplot displays only a representative cluster of vectors, thus enhancing the clarity and interpretability.

Keywords: high-dimensional data, biplots, principal component analysis, clustering, k-means algorithm

Contents

List of Figures

List of Tables

1 Introduction

In the world of today, data hold significant importance and are utilised across various industries to generate valuable insights. These insights facilitate informed decision making, drive research and innovation, personalise experiences, and more. However, obtaining useful information from the data requires a comprehensive understanding of it. Data visualisation serving as a pivotal tool in this endeavor.

According to Tsagaroulis (2020), data visualisation refers to techniques for graphically presenting information, aiming for clarity and simplicity, with added aesthetic elements to enhance appeal and capture the interest of an audience. Graphical representations serve as the foundation for data visualisation, facilitating a comprehensive display and summary of the structure of the dataset for effective analysis and interpretation.

There are various graphical representations, with scatterplots being the most common. These plots (scatterplots) are constructed so that the axes represent the variables, while the points represent the observations of the dataset. Although scatterplots are effective in data representation, they may not be as efficient when dealing with large datasets. Consequently, biplots have become increasingly useful and important in today's world of big data analysis.

Biplots are considered to be multivariate versions of scatterplots. They were first introduced by Gabriel (1971) and are considered one of the most effective visualisation tools for dealing with high-dimensional data. These plots provide a unique graphical representation that simultaneously displays both observations and variables (more than two) of a data matrix. In biplots, observations are visualised as points, and variables as non-orthogonal axes in a low-dimensional space.

The process of creating a biplot entails employing dimension reduction methods to project the original data into a space with fewer dimensions. According to Van Dyk (2022) dimension reduction methods are used to transform a dataset from a space with high dimensions to one with lower dimensions, keeping all key information and qualities of the original dataset. Planar displays are the most convenient for presentations of biplots.

The simplest and most fundamental type of biplot is based on PCA (Gabriel, 1971). The PCA biplot has essential elements such as the principal scores, loadings, and principal components that facilitate a comprehensive and interpretable visual representation of multivariate data. In this visualisation, observations are represented by principal scores, which are depicted as points that define their positions within the lower-dimensional space. The variables are represented by their loadings, which quantify the contribution of each variable to the underlying principal components. These principal components are the new variables that capture the maximum variance in the data. Each variable in the dataset has an arrow representing its associated loadings, which reveals associations between the principal components and variables. The loadings originate from the origin at an angle and have a certain length, extending in the direction of increasing variable variability (La Grange et al., 2009).

The biplot axes are calibrated to enable for accurate prediction and interpolation. Different calibrations are applied to facilitate reading off values and adding points to the biplot, making these axes essential for both purposes. Predictive axes are designed to ensure accurate orthogonal projection of a point onto an axis, thus enhancing graphical predictive ability. The interpolative axes are calibrated to allow the addition of new samples to an existing point configuration at the most feasible graphical position.

Using biplots in data analysis offers numerous advantages, such as aiding to uncover various data patterns, including clustering, multicollinearity, and multivariate outliers. However, biplots can become overcrowded when visualising many variables due to limited space and complexity, making interpretation difficult. Blasius et al. (2009) proposed a solution by suggesting that the axes be translated parallel to their initial positions to encircle the data points instead of intersecting at a central point. Although this approach retains all the information from the original biplot, it does not automatically result in aesthetically pleasing plots. Shifting the axes away from the data points can still result in overcrowding along the edges of the plot when dealing with large datasets.

Sandilands (2020) expanded on this idea by developing an algorithm for the automatic adjustment of the axes. Their approach aimed to improve biplot visualisation by dynamically repositioning the variable axes to better accommodate them within the biplot.

However, despite these advances, the issue of overcrowding persists with a very large number of variables. The automatically adjusted variable axes can still become overcrowded due to space constraints as the dataset grows. This ongoing challenge underscores the need for further refinement and innovation in biplot visualisation techniques to improve clarity within the plots when dealing with datasets containing many variables.

This study presents an alternative solution that uses one of the clustering techniques to migrate the overcrowding of variables in biplots. Specifically, the k-means clustering algorithm, introduced by MacQueen et al. (1967), is used to cluster variables into K distinct groups based on their similarities. The ultimate goal is for the biplot to display clusters of variables, each represented by a single centroid vector. This approach effectively alleviates overcrowding, enhancing both clarity and interpretability.

This study explores the following key research questions:

1. How to reduce overcrowding in a biplot when dealing with a dataset that has a large number of variables?

2. What are the most effective methods for presenting variables to enhance the visualisation of a biplot?

The software used for data analysis in this study is the R programming language by (R Core Team, 2025), specifically the biplotEZ package recently released by (Lubbe et al., 2023).

The study is structured as follows: Sections 2 and 3 are comprehensive reviews of the literature on dimension reduction and clustering techniques, respectively, highlighting their importance. Specifically, Section 2 covers dimension reduction techniques with a focus on PCA, as a popular method for analysing continuous data, aiming to create new uncorrelated variables called principal components. This section also covers the literature on biplots and their applications in various fields, as well as how the PCA biplot is constructed. Section 3 covers various clustering techniques, including k-means, reduced k-means (RKM), and factorial k-means (FKM), with a focus on their application to continuous data. Section 4 reviews existing solutions for visualising high-dimensional data in biplots, focusing on overcrowding issues. It discusses automated axis translation

by Sandilands (2020) as a recent improvement, while also noting its limitations with large variable sets, highlighting the need for further refinement. Section 5 presents the methodology of this study, which involves clustering variables within the biplot based on their directions and the angles they form. This approach is illustrated with examples from well-known datasets, such as the *Iris* dataset by Anderson (1935) and Fisher (1936), as well as the *mtcars* dataset by Henderson and Velleman (1981). The effectiveness of this methodology is evaluated by comparing it with the traditional method of clustering raw data without the biplot framework in Section 6. To further explore the effectiveness of the proposed solution, Section 6 also clusters the variables of a real-world dataset, the *Colon* data by Alon et al. (1999), which features a very large number of variables. The study concludes with Section 7, which summarises the main findings, highlights key implications, and identifies avenues for further research.

2 Dimension reduction techniques

This section provides an overview of dimension reduction techniques, particularly through the lens of biplots. It begins with a discussion of PCA, initially introduced by Pearson (1901) as a method for approximating a data matrix with n observations measured across p variables, $\mathbf{X}_{n\times p}$. Later, Hotelling (1933) expanded on PCA by formulating it in terms of the covariance matrix, $\mathbf{S}_{p\times p}$. The section then offers an overview of biplots, culminating in a theoretical framework for constructing a PCA biplot.

2.1 Introduction

Dimension reduction techniques are used to transform a dataset from a space with high dimensions to one with lower dimensions while retaining all key information and qualities of the original dataset (Van Dyk, 2022). The dimensionality of the data indicates the number of variables measured for each observation (Fodor, 2002). Therefore, high-dimensional data refer to datasets with numerous variables, while low-dimensional data typically involve fewer variables.

Working with high-dimensional data poses significant challenges. A key issue, highlighted by Fodor (2002), is the presence of redundant variables that do not contribute to understanding the underlying phenomena. Compounding this problem is the "curse of dimensionality", which leads to data sparsity and impracticality of analysis in high-dimensional spaces (Van der Maaten et al., 2009). Thus, dimension reduction techniques are crucial to effectively analyse and visualise high-dimensional data.

The choice of dimension reduction technique depends on the characteristics of the data. For example, PCA is commonly applied when the dataset consists solely of continuous variables. In contrast, Multiple Correspondence Analysis (MCA), a technique designed for categorical data, is more appropriate for datasets that include categorical variables. Other alternatives, such as categorical PCA, can also be used (Van Dyk, 2022).

2.2 Principal Component Analysis

PCA is a prevalent dimension reduction method to reduce the dimensionality of large datasets into fewer dimensions, while maximising the retention of data variability. By transforming the original variables into a smaller number of uncorrelated variables, called principal components, PCA captures the directions in which the data vary the most. PCA can be achieved in two ways: (1) through the singular value decomposition of a centered data matrix $\mathbf{X}_{n\times p}$ or (2) through the eigendecomposition of the covariance matrix of $\mathbf{X}$, $\mathbf{S}_{p\times p}$.

Beginning with the definition of Hotelling (1933), the principal components are computed as linear combinations.

$$\mathbf{y}_j = \mathbf{X}\mathbf{a}_j \tag{2.1}$$

for $j = 1, 2, \ldots, p$. The vectors $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_p$ are those uncorrelated p principal components whose variances are as large as possible and $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p$ are defined as principal component loadings. Thus, the first principal component is the linear combination of the original variables that has the largest possible variance; that is, it maximises the expression $\mathbf{a}_1'\mathbf{S}\mathbf{a}_1$. Subsequently, the second principal component is the linear combination that maximises $\mathbf{a}_2'\mathbf{S}\mathbf{a}_2$, subject to being uncorrelated with the first principal component. The j-th principal component maximises $\mathbf{a}_j'\mathbf{S}\mathbf{a}_j$ and is uncorrelated to the k principal components, for $k < j$.

The aim is to find the vector $\mathbf{a}_j$ that maximises this variance, which leads to maximising the quadratic form $\mathbf{a}_j'\mathbf{S}\mathbf{a}_j$ (Jolliffe and Cadima, 2016). To ensure a well-defined solution, the vector $\mathbf{a}_j$ is constrained to have unit-norm, meaning $\mathbf{a}_j'\mathbf{a}_j = 1$. This problem becomes equivalent to maximising

$$\mathbf{a}_j'\mathbf{S}\mathbf{a}_j - \lambda(\mathbf{a}_j'\mathbf{a}_j - 1), \tag{2.2}$$

where λ is a Lagrange multiplier. By differentiating this expression with respect to $\mathbf{a}_j$ and equating it to zero yields the following equation:

$$\mathbf{S}\mathbf{a}_j = \lambda \mathbf{a}_j, \tag{2.3}$$

indicating that $\mathbf{a}_j$ must be an eigenvector of $\mathbf{S}$, and λ is the corresponding eigenvalue. The eigenvalue represents the variance of the linear combination defined by the eigenvector $\text{var}(\mathbf{X}\mathbf{a}_j) = \mathbf{a}_j'\mathbf{S}\mathbf{a}_j = \lambda \mathbf{a}_j'\mathbf{a}_j = \lambda$. The largest eigenvalue, λ_1, and its corresponding eigenvector, $\mathbf{a}_1$, are of particular interest, as they maximise the variance captured by the first principal component (Jolliffe and Cadima, 2016).

The principal loadings can therefore be found from the eigendecomposition of $\mathbf{S}$, given as

$$\mathbf{S} = \mathbf{A}\mathbf{\Lambda}\mathbf{A}', \tag{2.4}$$

where the columns of $\mathbf{A}$ are used in the linear combinations to find the principal components, $\mathbf{Y} = [\mathbf{y}_1\, \mathbf{y}_2\, \ldots\, \mathbf{y}_p]$.

PCA can also be understood by approximating the data matrix $\mathbf{X}$ using singular value decomposition (SVD) as Pearson (1901) has pointed out. Consider the matrix $\mathbf{X}_{n\times p}$ of rank r, where $r \leq \min(n, p)$ and $\mathbf{X}$ is centered. Centering the matrix does not affect the solution (other than adjusting for the mean), since the covariance matrix remains unchanged regardless of whether the variables are centered. However, centering provides a clearer connection to a geometric approach to PCA (Jolliffe and Cadima, 2016).

PCA uses a least-squares criterion as the basis of the approximation, where the sum of squares of the differences between the corresponding elements of $\mathbf{X}$ and its approximation $\hat{\mathbf{X}}$ is minimised. This can be written as follows.

$$\min \left\| \mathbf{X} - \widehat{\mathbf{X}} \right\|^2 = \text{tr}\left\{ (\mathbf{X} - \widehat{\mathbf{X}})(\mathbf{X} - \widehat{\mathbf{X}})' \right\}. \tag{2.5}$$

The SVD of the centred matrix $\mathbf{X}$ is given as

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}', \tag{2.6}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices of dimensions $n \times p$ and $p \times p$, respectively. Therefore, $\mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbf{I}$, with $\mathbf{I}$ being the identity matrix of $p \times p$ dimensions. The

diagonal matrix $\mathbf{D}$ contains the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, representing the non-negative square roots of the non-zero eigenvalues of both $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}\mathbf{X}'$. The columns of $\mathbf{V}$, known as the right singular vectors, are the eigenvectors of $\mathbf{X}'\mathbf{X}$, while the columns of $\mathbf{U}$, the left singular vectors, correspond to the eigenvectors of $\mathbf{X}\mathbf{X}'$.

According to the Eckart and Young (1936) theorem, the optimal rank-r approximation $\widehat{\mathbf{X}}$ is obtained by keeping only the first r largest singular values and their corresponding singular vectors:

$$\widehat{\mathbf{X}} = \mathbf{U}_r\mathbf{D}_r\mathbf{V}_r', \tag{2.7}$$

where, $\mathbf{U}_r$ and $\mathbf{V}_r$ consist of the first r columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. $\mathbf{D}_r$ is the $r \times r$ diagonal matrix of the largest singular values.

The connection between the eigendecomposition of the covariance matrix $\mathbf{S}$ and the SVD of $\mathbf{X}$ is expressed by the following relationship:

$$\begin{aligned}(n-1)\mathbf{S} &= \mathbf{X}'\mathbf{X} && (2.8)\\ &= \mathbf{V}\mathbf{D}\mathbf{U}'\mathbf{U}\mathbf{D}\mathbf{V}' && (2.9)\\ &= \mathbf{V}\mathbf{\Lambda}\mathbf{V}', && (2.10)\end{aligned}$$

thus relating $\mathbf{V} = \mathbf{A}$. Consequently, the right singular vectors of the centered data matrix $\mathbf{X}$ serve as the principal loadings, and the principal components are obtained as $\mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{D}$.

Geometrically, the rank-r approximation projects the n points in p-dimensional space onto a r-dimensional subspace, minimising the sum of squared distances between the points and their projections. The variance explained by each principal component is proportional to the corresponding eigenvalue (or squared singular value). The proportion of total variance explained by the jth principal component is given by:

$$\pi_j = \frac{\lambda_j}{\sum_{j=1}^{p}\lambda_j} = \frac{\lambda_j}{\mathrm{tr}(\mathbf{S})} \quad \text{for } j = 1, 2, \ldots, p. \tag{2.11}$$

This criterion is commonly used to decide how many principal components should be retained for a lower dimensional representation of the data. In practice, the first two or three principal components are often used for visualisations. Figure 2.1 shows an example with samples plotted against two variables X_1 and X_2, and two vectors representing the first two principal components (PC1 and PC2).

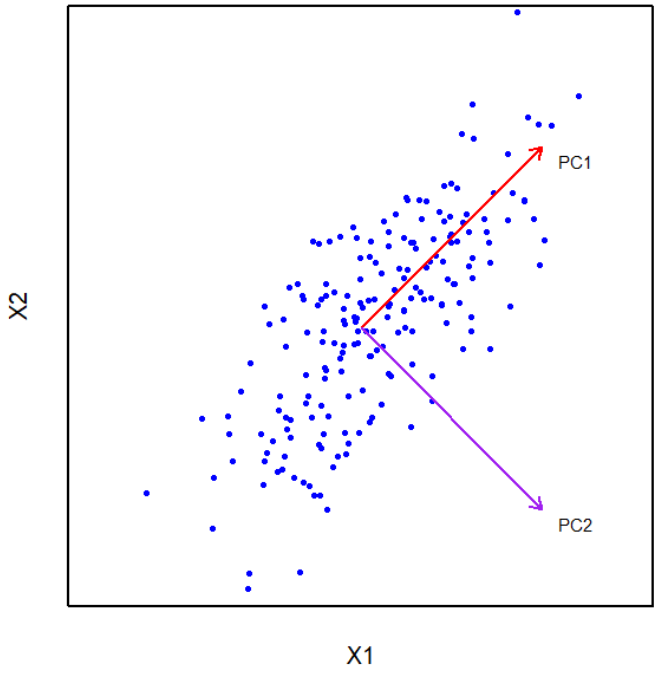

Figure 2.1: These principal components capture the maximum variation in the data, with PC1 denoting the 1st principal component and PC2 indicating the 2nd principal component.

2.3 Biplots

A biplot was first introduced by Gabriel (1971), as a tool used to visually assess the structure of a data matrix, where points represent observations and vectors represent variables. A subsequent refinement by Gower and Hand (1996) introduced calibrated axes to represent variables, while retaining points to represent observations. This allows for the projection of observations onto the axes, providing an approximation of the original variable values.

Biplots have two main types: asymmetric and symmetric. Asymmetric biplots visualise both rows (observations) and columns (variables) from a data matrix, whereas symmetric biplots represent observations and variables from a contingency table. The distinction between these types is rooted in the contribution of singular values: in asymmetric biplots, the contributions are not equal for variable and sample representation, whereas in symmetric biplots, the singular values contribute equally to both modes. This distinction aligns with the difference between principal and standard coordinates used in contingency tables. Additionally, while the interchange of rows and columns in symmetric biplots does not result in information loss, asymmetric biplots maintain distinct roles for variables and sample units that cannot be interchanged. Consequently, asymmetric displays are applicable in several multivariate techniques, including PCA, Correspondence Analysis (CA)—a technique used to explore relationships between categorical variables in a contingency table (Greenacre, 2017), and MCA.

This study focuses on asymmetrical biplots, with a specific focus on PCA as a fundamental example. The next subsection explains how to construct a biplot within the PCA framework.

2.4 Construction of a PCA biplot

The construction of a biplot involves representing both observations and variables. The positions of the observations are determined by their projections on the principal components, while the direction and length of the axis of each variable indicate its contribution to the principal components. The representation of observations and variables in a biplot is discussed in detail in the following sub-subsections.

2.4.1 Representation of samples

In a biplot, samples are projected orthogonally onto a reduced-dimensional plane, known as the biplot plane, to visualise high-dimensional data in two dimensions. The biplot plane, denoted by L is determined to minimise the sum of squared distances from the original data points to the plane (Gower et al., 2011). To project a sample point x onto

this plane, the formula used is

$$\mathbf{x}'_{proj} = \mathbf{x}'\mathbf{V}_2\mathbf{V}'_2, \tag{2.12}$$

where $\mathbf{V}_2$ represents the first two columns of the approximation of $\mathbf{X}$ in equation 2.7. Therefore, the coordinates of the projected points in the biplot plane are calculated using

$$\mathbf{z}' = \mathbf{x}'\mathbf{V}_2. \tag{2.13}$$

To illustrate this construction, a synthetic dataset of 100 observations across five variables, $Var1$, $Var2$, ..., $Var5$ was generated from a standard normal distribution using the `rnorm()` function. PCA was then applied, and the resulting scatter plot, shown in Figure 2.2, displays the projections of the observations onto the first two principal components.

The full R code used to generate this figure is provided in Appendix A.

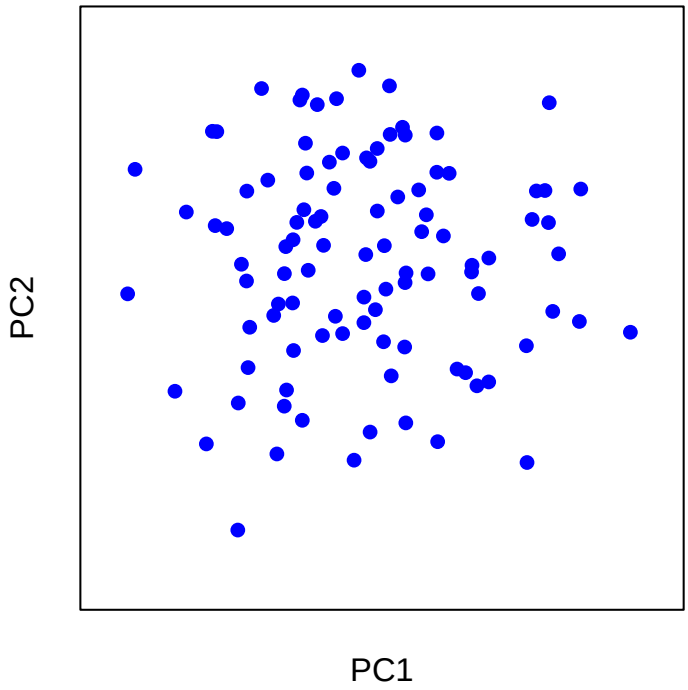

Figure 2.2: Observations representation in a biplot.

The next subsection extends this visualisation by incorporating the variable vectors, thereby providing a more comprehensive view of the relationships between observations and variables within the PCA framework.

2.4.2 Representation of variables

A biplot represents variables as vectors, as suggested by Gabriel (1971). Figure 2.3, which builds upon Figure 2.2, illustrates this using an inner product representation. In this representation, the biplot axes are shown as vectors v_k, with their endpoints $\mathbf{V}_k$ whose coordinates are extracted from the first two elements of the k-th row of the matrix

$\mathbf{V}$. The value $\hat{x}_{ik}$, associated with the point $\mathbf{P}_i$ and the vector v_k, is calculated as the product of the lengths $\mathbf{OP}_i$ and $\mathbf{OV}_k$, and the cosine of the angle θ formed at the origin. The matrix $\hat{\mathbf{X}}$ contains all np values of the inner product. However, visualising the inner product is complex but becomes clearer when comparing two points $\mathbf{P}_i$ and $\mathbf{P}_j$ with respect to the same variable v_k. All points $\mathbf{P}$ that project onto the same location on $\mathbf{OV}_k$ share the same inner product value.

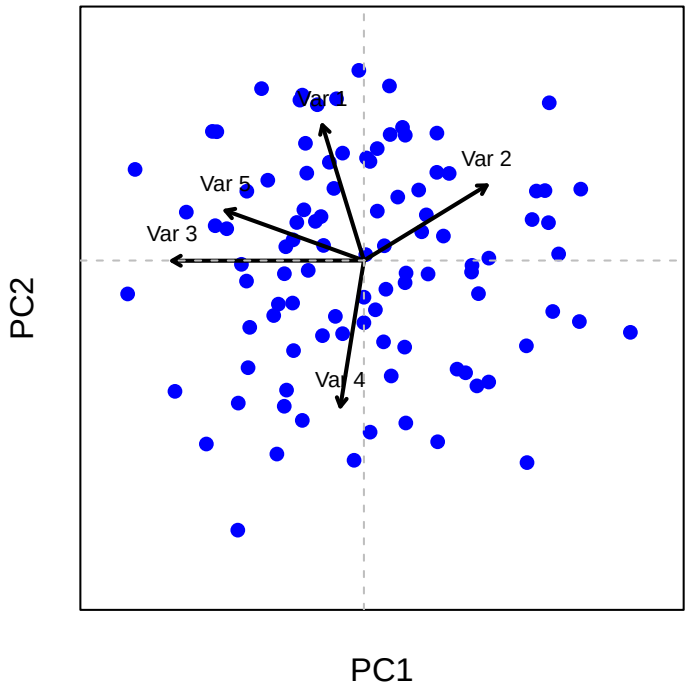

Figure 2.3: Variables and observations representation in a biplot.

This leads to the suggestion by Gower and Hand (1996) that biplot axes should be calibrated like standard coordinate axes. The calibration of axes in a biplot involves scaling them to represent the original variables, enabling the relationships between the

samples and variables to be interpreted accurately. A biplot with calibrated axes, as seen in Figure 2.4, provides a visual representation of the data, allowing the projection of samples onto the axes to determine their corresponding variable values. This approach retains the interpretability of scatterplots, while incorporating additional features such as multiple non-orthogonal axes.

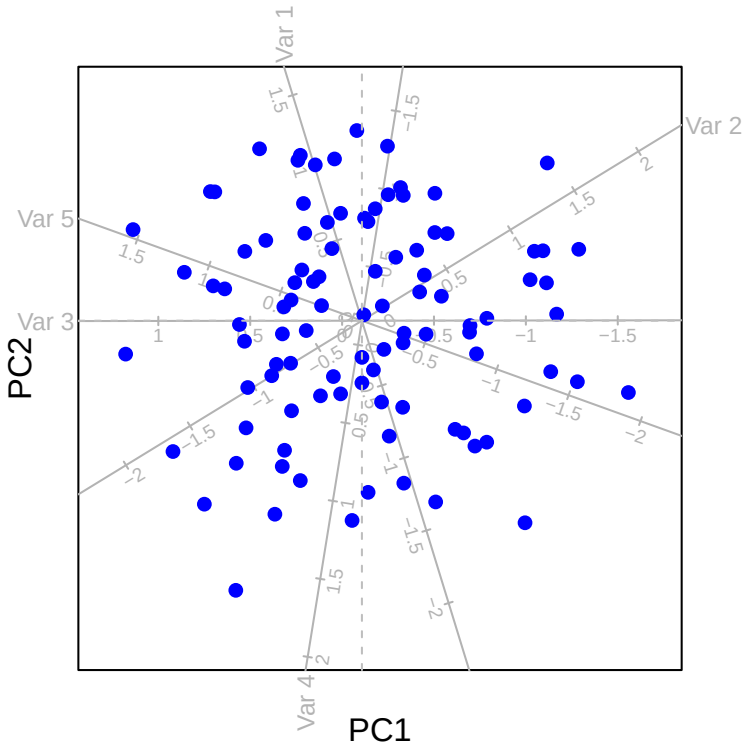

Figure 2.4: Calibrated biplot axes.

Mathematically, the calibration of axes on a biplot is based on the use of the inner product. For example, consider $\mathbf{A}$ as a $p \times 2$ matrix, where its rows represent the coordinates of a set of points. Let $\mathbf{B}$ be a $q \times 2$ matrix, where its rows represent the directions of the axes to be calibrated. Therefore, for a given sample a_i and axis b_k, the inner product $a_i' b_k$ is

constant (μ) for all points on the projection line of a_i onto b_k. This constant μ serves as the calibration marker for the projection point, λb_k.
The calibration process ensures that:

$$\lambda b_k' b_k = \mu, \tag{2.14}$$

solving for λ, we obtain:

$$\lambda = \frac{\mu}{b_k' b_k}. \tag{2.15}$$

Thus, the coordinates of the calibration point on the b_k-axis are:

$$\frac{\mu b_k}{b_k' b_k}. \tag{2.16}$$

Typically, μ is assigned values such as 1, 2, 3, ..., or other suitable increments for calibration, in order to obtain the necessary values for the inner products. Frequently, the inner product being approximated represents transformed values of some original variables, $a_i b_k = f(x_{ik})$, and the goal is to calibrate these values back to the original measurement units.

2.5 Measures of fit

Measures of fit in PCA biplots are used to assess how well the biplot represents the original dataset in a reduced-dimensional space. Consider the matrix $\mathbf{X}$, which can be decomposed into two parts: a fitted component

$$\hat{\mathbf{X}} = \mathbf{UJDJV}' = \mathbf{UDJ}_2(\mathbf{VJ}_2)' = \mathbf{UDV}'\mathbf{VJ}_2(\mathbf{VJ}_2)' = \mathbf{XVJV}', \tag{2.17}$$

where the matrix $\mathbf{J}$ of size $p \times p$ is defined as

$$\mathbf{J} = \begin{bmatrix} \mathbf{I_2} & 0 \\ 0 & 0 \end{bmatrix} \tag{2.18}$$

and a residual component given by $\mathbf{X} - \hat{\mathbf{X}}$.

According to Gabriel (1971), the extent of lack of fit is measured by minimising the

squared difference between $\hat{\mathbf{X}}$ and $\mathbf{X}$ because

$$\|\mathbf{X}\|^2 = \|\hat{\mathbf{X}}\|^2 + \|\mathbf{X} - \hat{\mathbf{X}}\|^2. \tag{2.19}$$

Therefore, the quality of fit is then quantified as

$$\text{quality} = \frac{\|\hat{\mathbf{X}}\|^2}{\|\mathbf{X}\|^2} = \frac{\text{tr}(\mathbf{X}\mathbf{X}')}{\text{tr}(\hat{\mathbf{X}}\hat{\mathbf{X}}')} = \frac{\text{tr}(\mathbf{X}'\mathbf{X})}{\text{tr}(\hat{\mathbf{X}}'\hat{\mathbf{X}})} = \frac{\text{tr}(\mathbf{V}\mathbf{D}^2\mathbf{V}')}{\text{tr}(\mathbf{V}\mathbf{D}^2\mathbf{J}\mathbf{V}')} \tag{2.20}$$

Alternatively, this quality measure can be expressed as a percentage.

$$\text{quality} = \left(\frac{d_1^2 + d_2^2}{d_1^2 + \cdots + d_p^2}\right) \times 100\%. \tag{2.21}$$

2.6 Adequacy of representation of the variables

In the context of PCA biplots, adequacy measures how well a variable is represented in a reduced-dimensional space (Gower et al., 2011). Adequacy values are calculated as

$$\frac{diag(\mathbf{V}_r\mathbf{V}_r')}{diag(\mathbf{V}\mathbf{V}')} = diag(\mathbf{V}_r\mathbf{V}_r'), \tag{2.22}$$

as described by Lubbe et al. (2023). These values range from 0 to 1, where 1 signifies perfect representation of the variable within the chosen dimensions, and values closer to 0 indicate a poorer representation (Gower et al., 2011).

A popular way to visualise the adequacy in biplots is to use a unit circle, as shown in Figure 2.5. The unit circle, centered at the origin of the biplot with a radius of one, represents the maximum possible adequacy for each variable. It is constructed such that variables with vectors extending to or near the boundary of this circle have high adequacy, indicating a strong representation in the reduced-dimensional space. Conversely, the further a variable's vector is from the unit circle boundary, the less adequately it is represented, suggesting that the variable is not well-captured in the chosen dimensions.

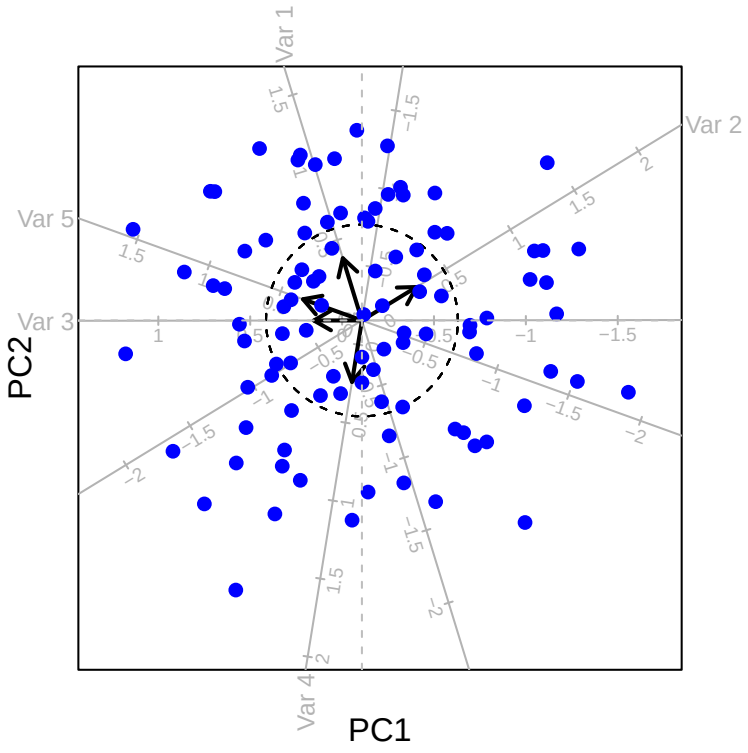

Figure 2.5: The PCA represents the variable vectors within the space defined by the unit circle.

2.7 Biplots applications

Biplots have proven to be a versatile tool in various fields since their introduction. For example, in the health sciences, Osmond (1985) applied biplots to visualise age and period-specific cancer mortality rates, highlighting changes in age-specific rates over time and identifying trends in age distribution through projections. In Economics, Barr et al. (1990) used covariance biplots to visualise multivariate time series data, particularly economic data such as the major currency exchange rates from 1976 to 1984. This technique effectively captures temporal changes in variables, their relationships, and how these re-

lationships evolve over time, serving as a valuable exploratory tool in economic analysis. Biplots have made significant contributions to finance, as demonstrated by Van den Honert and Barr (1988), who used them to study the impact of various explanatory factors on shareholder wealth in mergers and acquisitions. Using covariance biplots, the researchers visualised and compared the variability and dynamics of these characteristics, providing insights into their combined influence on shareholder outcomes. In ecological research, biplots have been instrumental in visualising species composition and diversity based on PCA biplots, as shown by ter Braak (1983).

Recently, Suryowati et al. (2021) applied biplots in the health field to map the spread of non-communicable diseases in 33 provinces of Indonesia. By visualising the relationships between diseases and provinces, they highlighted the proximity and correlation between different diseases and regions. This method helped to identify patterns and trends in the prevalence of twelve specific non-communicable diseases. In agriculture, Dang et al. (2024) used genotype $\times$ environment interaction (GGE) biplots to analyse crop yields (sugar beet varieties) under different environmental conditions. Guevara-Viejó et al. (2024) applied biplots in applied biosciences and bioengineering to evaluate the mycelial characteristics and antagonistic capacity of 50 Trichoderma spp. strains against Moniliophthora roreri using a PCA biplot, which identified five strains with notable antagonistic capacity in vitro. These applications underscore the versatility and effectiveness of biplots in facilitating insightful explorations and analyses across varied disciplines. There are many other studies that have also applied biplots, further demonstrating their wide-ranging utility.

2.8 Summary

This section introduced dimension reduction techniques in light of PCA. This technique, along with its biplot representation, is a widely used method for analysing continuous data. The section first looked at the fundamental goal of PCA, which is to generate a new set of uncorrelated variables known as principal components. Each principal component represents a linear combination of the original variables, with the principal components ordered according to the amount of variance they explain.

The section also explores biplots and the construction of PCA biplots, focusing on the representation of both samples and variables. This chapter additionally discusses measures of fit and the adequacy of variables within the plot, in order to assess how well the original dataset is represented in a biplot. Finally, the applications of biplots are discussed, highlighting their usefulness in various fields worldwide.

3 Clustering Techniques

Unsupervised learning focuses on analysing data without predefined labels or responses. Unlike supervised learning, where data have a dependent variable to model, unsupervised learning aims to uncover patterns and relationships among independent variables (Van Dyk, 2022). PCA, discussed in the previous section on dimension reduction techniques, and clustering, which is the focus of this section, are among the most popular methods for addressing unsupervised learning problems.

3.1 Introduction

The aim of cluster analysis is to classify the data into a certain number of sets where the elements of each set should be as similar as possible and dissimilar from those of other sets (Ruspini, 1969). Clustering plays an important role in today's data-driven era by uncovering hidden patterns and structures within the dataset. By partitioning datasets into coherent groups based on inherent similarities, clustering not only facilitates exploratory data analysis, but also supports decision making processes in different fields (James et al., 2023).

Clustering methods are often categorised into two distinct groups, similar to dimensional reduction techniques: continuous clustering for continuous data and categorical clustering for categorical data. For the purposes of this study, only the methods of clustering continuous data are considered.

3.2 Continuous clustering

Continuous clustering involves the grouping of numerical data points based on their similarities. Several unsupervised machine learning techniques for continuous clustering have been developed over time, each with its own strengths and suitable applications. Among the most widely recognised methods in the literature are k-means, reduced k-means (RKM), and factorial k-means (FKM) clustering. This study explores these methods in detail.

3.2.1 The *k*-means clustering algorithm

The primary goal of k-means clustering is to partition n observations into K distinct and non-overlapping clusters, where K is a predefined number (MacQueen et al., 1967). This approach divides the data set into (K) clusters, and in this study, (K) is specified by the user. Once the number of clusters is given, the k-means algorithm systematically assigns each data point to exactly one of the (K) clusters.

Consider a partition of the data into K distinct clusters, denoted $C_1, C_2, \ldots, C_K$. The objective is to group similar data points together while minimising the variation within each cluster. The within-cluster variation, $W(C_k)$, quantifies the variation and measures the degree of similarity between data points within a cluster. A lower within-cluster variation indicates higher similarity between the data points, whereas a higher variation suggests greater dissimilarity. The within-cluster variation for cluster k is mathematically represented as:

$$W(C_k) = \frac{1}{|C_k|} \left\{ \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}, \tag{3.1}$$

where $|C_k|$ is the number of observations in cluster k, and $(x_{ij} - x_{i'j})^2$ is the squared Euclidean distance between two observations in the cluster. Thus, in simpler terms, $W(C_k)$ measures the average of all pairwise squared Euclidean distances between its observations (James et al., 2023).

Therefore, the underlying objective of k-means clustering is to minimise the within-cluster variation. This is formulated as an optimisation problem:

$$\min_{C_1, C_2, \ldots, C_K} \left\{ \sum_{k=1}^{K} W(C_k) \right\}, \tag{3.2}$$

which combines equation 3.1 with equation 3.2, resulting in the following formulation:

$$\min_{C_1, C_2, \ldots, C_K} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2 \right\}. \tag{3.3}$$

The solution to the optimisation problem, or the objective function, in equation 3.3

involves developing an algorithm that divides the observations into K clusters in order to minimise its value. However, given that there are approximately K^n possible ways to partition n observations into K clusters, this task can be computationally challenging. To address this, James et al. (2023) proposed a straightforward algorithm designed to find a local minimum for the k-means optimisation problem. The algorithm proceeds as follows:

1. Randomly assign a number between 1 and K to each observation, which serves as its initial cluster designation.
2. Repeat the following steps until the cluster assignments stabilise:
   (a) For each of the K clusters, determine their centroid as the vector of mean values for the p variables of the observations within that cluster.
   (b) Reassign each observation to the cluster whose centroid is the nearest, based on the Euclidean distance.

This iterative process effectively minimises the objective function defined in equation 3.3 to its minimum value, which is shown by the following identity:

$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2, \tag{3.4}$$

where $\bar{x}_{kj}$ is the mean of variable j in cluster C_k, calculated as $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i\in C_k} x_{ij}$. Therefore, as the algorithm progresses, the clustering improves until it stabilises, meaning the objective function will not increase. Once no further changes occur, a local optimum has been attained. The k-means algorithm finds a local optimum rather than a global optimum, the ultimate outcome is influenced by the initial, randomly generated cluster assignments. Therefore, running the algorithm multiple times from different starting points is crucial, selecting the solution with the smallest value of the objective function.

3.2.2 Reduced *k*-means clustering

RKM and FKM clustering, introduced in this subsection and the next, respectively, are extensions of the k-means clustering method discussed in the previous section. These

techniques were developed to address the limitations of the tandem approach, which involves performing dimension reduction prior to clustering. This sequential process can lead to skewed results because dimension reduction and clustering aim to optimise different objectives. According to Yamamoto and Terada (2014), the first few principal components can be insufficient to reveal the cluster structure in the data, which can lead to suboptimal clustering results when using the tandem approach. Therefore, to address this limitation, RKM and FKM were developed, integrating dimension reduction and clustering into a single, simultaneous process that enhances data analysis.

To understand how RKM clustering works, consider a data matrix $\mathbf{X}$ comprising n observations across p variables. Define $\mathbf{Y}_K$ as a parameter matrix of dimensions $n \times k$, with a rank of k. Additionally, introduce a $p \times l$ matrix $\mathbf{B}$ that exhibits column-wise orthogonality, as expressed by the condition $\mathbf{B}'\mathbf{B} = \mathbf{I}_l$. Therefore, the objective function for RKM clustering is given by:

$$min_{\phi_{\text{RKM}}}(\mathbf{B}, \mathbf{Y}_K) = \|\mathbf{X} - \mathbf{PXBB}'\|^2, \tag{3.5}$$

where $\mathbf{P}$ is defined as $\mathbf{P} = \mathbf{Y}_K(\mathbf{Y}'_K\mathbf{Y}_K)^{-1}\mathbf{Y}'_K$ and $\|\mathbf{X} - \mathbf{PXBB}'\|^2$, representing the squared Frobenius norm of the matrix $\mathbf{X} - \mathbf{PXBB}'$.

Therefore, an alternative formulation of the objective function presented in equation 3.5 is:

$$\|\mathbf{X} - \mathbf{PXBB}'\|^2 = \text{tr}(\mathbf{X}'\mathbf{X}) - \text{tr}(\mathbf{B}'\mathbf{X}'\mathbf{PXB}). \tag{3.6}$$

Since minimising ϕ_{RKM} is equivalent to maximising $-\phi_{\text{RKM}}$, the within-cluster variance in RKM, $-\text{tr}(\mathbf{B}'\mathbf{X}'\mathbf{PXB})$, is maximised in the reduced space.

3.2.3 Factorial *k*-means clustering

The FKM method is designed to address specific limitations of the RKM. In particular, RKM may struggle when the variability of the data is substantial in directions orthogonal to the subspace containing the clusters. This can result in the misidentification of clusters or a failure to capture the underlying structure of the data effectively (Timmerman et al.,

2010).

The FKM and RKM algorithms have many similarities; however, their objective functions for optimisation differ. From the previous subsection on RKM, $\mathbf{X}$ represents the data matrix comprising n observations across p variables. Let $\mathbf{Y}_K$ be the parameter matrix of size $n \times k$, with a rank of k, and let $\mathbf{B}$ be a $p \times l$ matrix that is column-wise orthogonal such that $\mathbf{B}'\mathbf{B} = \mathbf{I}_l$. Additionally, the matrix $\mathbf{P}$ is defined as $\mathbf{Y}_K(\mathbf{Y}'_K\mathbf{Y}_K)^{-1}\mathbf{Y}'_K$. Therefore, the FKM loss function, which is to be minimised, is as follows:

$$min_{\phi_{\text{FKM}}}(\mathbf{B}, \mathbf{Y}_K) = \|\mathbf{XB} - \mathbf{PXB}\|^2, \tag{3.7}$$

with respect to $\mathbf{B}$ and $\mathbf{Y}_K$.

3.2.4 Reduced *k*-means and factorial *k*-means

RKM and FKM are sensitive to irrelevant correlated variables that do not contribute to the cluster structure. As noted by Yamamoto and Terada (2014), the presence of such variables makes these methods more likely to fail in identifying an optimal subspace for the cluster structure. Therefore, to address the issue of irrelevant variables, Yamamoto and Terada (2014) proposed a method that combines subspace separation with clustering. This approach identifies and separates relevant and irrelevant variables into distinct subspaces.

The RKM objective function (equation 3.5) is decomposed to show its components:

$$\|\mathbf{X} - \mathbf{PXBB}'\|^2 = \|\mathbf{X} - \mathbf{XBB}'\|^2 + \|\mathbf{XB} - \mathbf{PXB}\|^2. \tag{3.8}$$

This decomposition reveals that RKM comprises two distinct components: the first part corresponds to PCA, while the second part relates to FKM.

However, Vichi et al. (2019) suggested an alternative approach to use a convex combination of these components to form a new objective function, instead of assigning equal weights to the two components. This method seeks to minimise a convex combination of the two components. Therefore, the new objective function is given by:

$$\min_{\phi_{\text{ClusPCA}}} (\mathbf{B}, \mathbf{Y}_K) = \alpha \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{B}'\|^2 + \beta \|\mathbf{X}\mathbf{B} - \mathbf{P}\mathbf{X}\mathbf{B}\|^2, \tag{3.9}$$

where $\alpha + \beta = 1$.

The objective function is simplified by expanding the squared Frobenius norms and applying the trace operator, a standard technique for such minimisation:

$$\|\mathbf{A}\|^2 = \operatorname{tr}(\mathbf{A}'\mathbf{A}). \tag{3.10}$$

Expanding the first term in equation 3.9, the result is:

$$\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{B}'\|^2 = \operatorname{tr}\left[(\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{B}')'(\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{B}')\right]. \tag{3.11}$$

Further expansion leads to:

$$\begin{aligned} \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{B}'\|^2 &= \operatorname{tr}(\mathbf{X}'\mathbf{X} - \mathbf{X}'\mathbf{X}\mathbf{B}\mathbf{B}' - \mathbf{B}\mathbf{B}'\mathbf{X}'\mathbf{X} + \mathbf{B}\mathbf{B}'\mathbf{X}'\mathbf{X}\mathbf{B}\mathbf{B}') \\ &= \operatorname{tr}(\mathbf{X}'\mathbf{X}) - 2\operatorname{tr}(\mathbf{X}'\mathbf{X}\mathbf{B}\mathbf{B}') + \operatorname{tr}(\mathbf{B}'\mathbf{X}'\mathbf{X}\mathbf{B}). \end{aligned} \tag{3.12}$$

Under the assumption that $\mathbf{B}$ has orthonormal columns (i.e., $\mathbf{B}'\mathbf{B} = \mathbf{I}$), the expression $\operatorname{tr}(\mathbf{B}\mathbf{B}'\mathbf{X}'\mathbf{X}\mathbf{B}\mathbf{B}')$ simplifies to $\operatorname{tr}(\mathbf{B}'\mathbf{X}'\mathbf{X}\mathbf{B})$, since the projection $\mathbf{B}\mathbf{B}'$ does not alter the subspace captured by $\mathbf{B}$.

Thus, the simplified expression is:

$$\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{B}'\|^2 = \operatorname{tr}(\mathbf{X}'\mathbf{X}) - \operatorname{tr}(\mathbf{B}'\mathbf{X}'\mathbf{X}\mathbf{B}). \tag{3.13}$$

Similarly, for the second term in equation 3.9:

$$\|\mathbf{X}\mathbf{B} - \mathbf{P}\mathbf{X}\mathbf{B}\|^2 = \operatorname{tr}\left[(\mathbf{X}\mathbf{B} - \mathbf{P}\mathbf{X}\mathbf{B})'(\mathbf{X}\mathbf{B} - \mathbf{P}\mathbf{X}\mathbf{B})\right]. \tag{3.14}$$

This simplifies to:

$$\|\mathbf{X}\mathbf{B} - \mathbf{P}\mathbf{X}\mathbf{B}\|^2 = \operatorname{tr}(\mathbf{B}'\mathbf{X}'(\mathbf{I} - \mathbf{P})\mathbf{X}\mathbf{B}). \tag{3.15}$$

By substituting these expressions back into equation 3.9, the objective function becomes:

$$\phi_{\text{ClusPCA}}(\mathbf{B}, \mathbf{Y}_K) = \alpha\,(\text{tr}(\mathbf{X}'\mathbf{X}) - \text{tr}(\mathbf{B}'\mathbf{X}'\mathbf{X}\mathbf{B})) + \beta\,\text{tr}(\mathbf{B}'\mathbf{X}'(\mathbf{I}-\mathbf{P})\mathbf{X}\mathbf{B}). \tag{3.16}$$

This can be rearranged as:

$$\phi_{\text{ClusPCA}}(\mathbf{B}, \mathbf{Y}_K) = \alpha\,\text{tr}(\mathbf{X}'\mathbf{X}) - \alpha\,\text{tr}(\mathbf{B}'\mathbf{X}'\mathbf{X}\mathbf{B}) + \beta\,\text{tr}(\mathbf{B}'\mathbf{X}'(\mathbf{I}-\mathbf{P})\mathbf{X}\mathbf{B}). \tag{3.17}$$

Since $\text{tr}(\mathbf{X}'\mathbf{X})$ is constant with respect to $\mathbf{B}$, minimising ϕ_{ClusPCA} reduces to maximising the terms involving $\mathbf{B}$:

$$\max_{\mathbf{B}} \text{tr}\left[\mathbf{B}'\mathbf{X}'\left(\beta(\mathbf{I}-\mathbf{P}) - \alpha\mathbf{I}\right)\mathbf{X}\mathbf{B}\right]. \tag{3.18}$$

Using the fact that $\alpha + \beta = 1$, this expression can be rewritten as:

$$\max_{\mathbf{B}} \text{tr}\left[\mathbf{B}'\mathbf{X}'\left((\alpha-\beta)\mathbf{P} - (1-\alpha)\mathbf{I}\right)\mathbf{X}\mathbf{B}\right]. \tag{3.19}$$

This leads to the following formulation:

$$\text{tr}\left(\mathbf{B}'\mathbf{X}'\left((\alpha-\beta)\mathbf{P} - (1-\alpha)\mathbf{I}\right)\mathbf{X}\mathbf{B}\right). \tag{3.20}$$

The optimisation problem is thus equivalent to the standard k-means objective function applied to $\mathbf{XB}$. This formulation can be integrated into an alternating least-squares algorithm, as detailed below.

1. Initialise the cluster assignment matrix $\mathbf{Y}_K$, for example, by randomly assigning objects to clusters.

2. Compute the loadings $\mathbf{B}$ by performing the eigendecomposition of

$$\mathbf{X}'\left((1-\alpha)\mathbf{P} - (1-2\alpha)\mathbf{I}\right)\mathbf{X}.$$

3. Update the cluster assignments $\mathbf{Y}_K$ by applying k-means on the reduced space defined by the coordinates $\mathbf{XB}$.

4. Repeat steps 2 and 3 iteratively, using the updated $\mathbf{Y}_K$, until convergence is achieved when $\mathbf{Y}_K$ no longer changes.

As a result, when $\alpha_1 = 0.5$, 0, or 1, the problem simplifies to RKM, FKM, or the tandem approach, respectively.

3.3 Summary

This chapter explored unsupervised learning, specifically clustering, as an essential tool for exploratory data analysis and decision making by uncovering hidden patterns and structures. Clustering algorithms group data based on similarities, improving biplot interpretability and identifying group-specific features. Techniques like k-means, RKM, and FKM are discussed, highlighting their use in continuous data clustering, with k-means focusing on minimising within-cluster variance and RKM and FKM incorporating PCA to enhance cluster structures.

4 Existing solutions

Biplots often encounter several issues when handling large datasets with many variables and observations. One major problem is overcrowding within the plot. As the number of variables and observations increases, the limited space in a biplot often results in overlapping variable axes and data points. This overcrowding can obscure relationships and make it difficult to interpret the plot effectively.

An existing approach proposed by Yan and Tinker (2006) involves presenting the variable axes across separate biplots. This strategy effectively mitigates overcrowding by accommodating fewer variables axes in each plot. However, a drawback of this approach is that it requires the analysis of multiple biplots for a single dataset, introducing complexities in both data analysis and interpretation.

Blasius et al. (2009) proposed a method to address this problem by translating the axes to encircle the data points rather than to intersect at the origin. However, this approach does not ensure aesthetically pleasing plots and can still result in overcrowded axes at the edges of the plot. Building on this, Sandilands (2020) developed an algorithmic solution for automatic axes translation to optimise space usage and improve visualisation clarity on biplots. Despite these advances, overcrowding remains a persistent issue, especially with larger and more complex datasets. Automatically translated axes still encounter spatial constraints, affecting readability and interpretability.

However, it is crucial to explore the solutions proposed by these authors, particularly those outlined in Sandilands (2020), as they represent the latest advances in the field. The subsequent chapter further investigates these approaches and their implications for improving biplot visualisations.

4.1 Automated axis translation

The algorithm for automatic translation of the axes proposed by Sandilands (2020) begins by determining the minimum distance (D_{base}) required to translate an axis parallel from the origin and sets a multiplication factor ($mult$) to indicate direction. It calculates the

slope (m_i) for each axis i using the starting point (a_i) and the end point (b_i) of the axis. A small incremental distance (δ_D) is initialised to fine-tune the axis shifting. The algorithm translates each axis i by computing the new positions (z_1, z_2) for the start and end points, adjusting the direction multiplier based on the axis index. For each axis i greater than 1, it is compared to all previously processed axes j to resolve intersections. If intersections are detected, a counter (t) is incremented, the direction multiplier is adjusted, and the shift distance (D_{shift}) is incremented by a factor of δ_D. The new positions (z_1, z_2) are recalculated using the updated D_{shift} and the direction multiplier. This process is repeated for each axis until all axes are translated and any intersections are resolved, ensuring the clarity and distinguishability of the axes and labels.

The results presented in Figure 4.1 were obtained directly from Sandilands (2020), using an algorithm introduced for the automatic translation of the axes. This figure offers a clear representation of the *mtcars* dataset (Henderson and Velleman, 1981) after the algorithm's application. By systematically translating the axes, the algorithm better aligns them with the data structure, enhancing intuitive and informative interpretation.

This algorithm yields biplots that are more discernible, interpretable, and aesthetically pleasing by systematically translating the axes away from the origin, thus preventing overlaps and enhancing clarity in visual representation. By ensuring that the axes do not intersect at the origin and are shifted parallel to their original positions, it facilitates a better understanding of the relationships between variables.

However, despite these improvements, this solution encounters limitations when faced with datasets containing a large number of variables. In such cases, the sheer volume of variables may still result in overcrowding of the translated axes, diminishing the efficacy of the biplot visualisation. This challenge underscores the need for further refinement or complementary strategies to effectively address overcrowding and maintain the utility of biplots in complex data analysis scenarios.

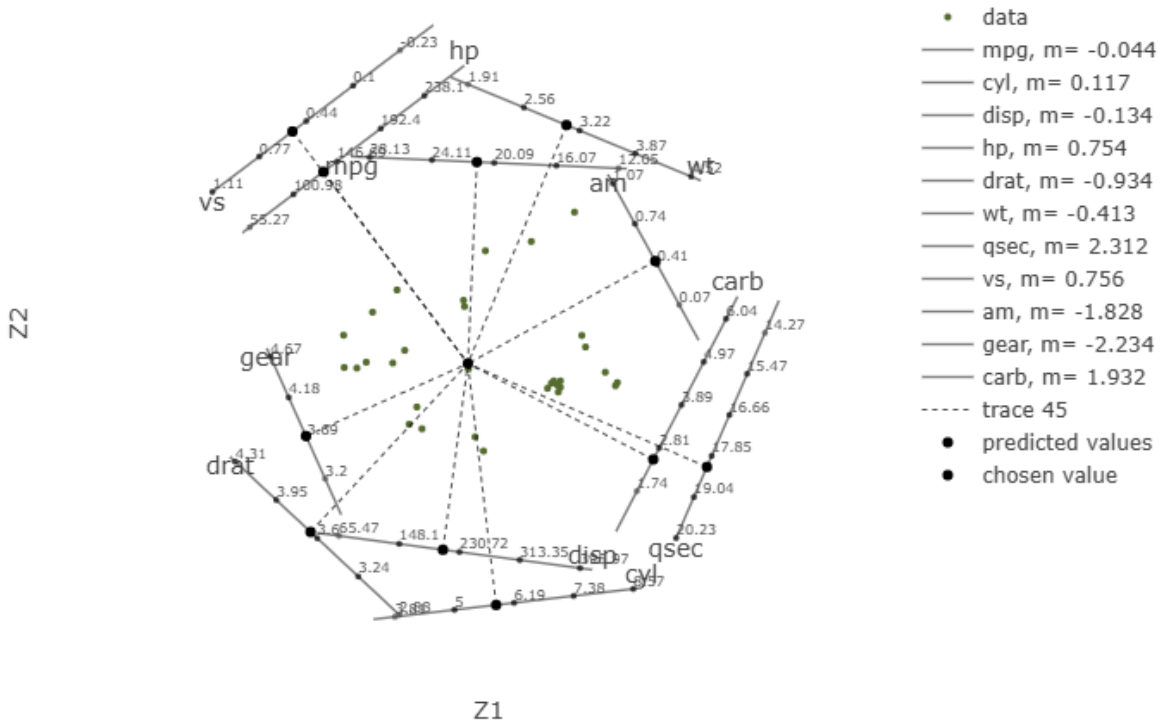

Figure 4.1: The PCA biplot of the *mtcars* dataset, produced by Sandilands (2020).

An HTML version of this plot can be accessed at https://doi.org/10.5281/zenodo.4066399. It offers various interactive functions, including the ability to view sample names when hovering over data points and the option to add or remove plot elements (Sandilands, 2020).

5 Methodology

In the previous chapter, existing methods for addressing overcrowding in a biplot were discussed. This method may still face challenges with very large datasets, as the automatically translated axes can become overcrowded due to space constraints. Therefore, this study proposes an alternative solution for visualising a large number of variables in a biplot by clustering variables with similar directions and positions using the k-means clustering algorithm. By displaying only the clusters of variable axes, this approach facilitates clearer interpretation and reduces visual clutter. This chapter provides a detailed exploration of the proposed methodology.

The proposed method commences with the extraction of the endpoints of each variable (vector) from the two-dimensional biplot. These endpoints are stored in the matrix $\mathbf{V}_2$, which consists of the first two columns of the matrix $\mathbf{V}$ in equation 2.6, and can be expressed as follows:

$$\mathbf{V}_2 = \begin{bmatrix} v_{1x} & v_{1y} \\ v_{2x} & v_{2y} \\ \vdots & \vdots \\ v_{px} & v_{py} \end{bmatrix} \tag{5.1}$$

where x represents the first column, and y represents the second column. The $\mathbf{V}_2$ matrix is then used to compute the angle of each vector as:

$$\theta_i = \tan^{-1}\left(\frac{v_{iy}}{v_{ix}}\right), \quad \text{for } i = 1, 2, 3, \ldots, p, \tag{5.2}$$

where θ_i represents the direction of the vectors in the biplot. The vector endpoints in equation 5.1 and the corresponding angles in equation 5.2 are illustrated in Figure 5.1.

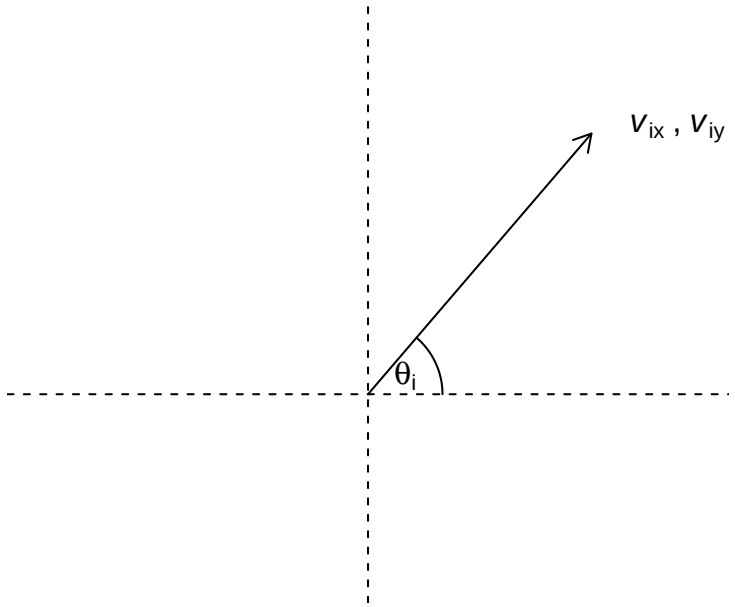

Figure 5.1: Vector endpoints and their corresponding angles.

$\mathbf{V}_2$, along with the angles θ_i that the vectors form, constitutes the first parameter of the k-means clustering algorithm. The second parameter is the user-specified number of clusters, denoted as K. Therefore, together, K, θ_i, and $\mathbf{V}_2$ form the foundation for the k-means clustering process, which is expressed as follows:

$$\mathbf{C} = k\text{-means}(\mathbf{F}, K). \tag{5.3}$$

where $\mathbf{F} = [\mathbf{V}_2 \ \theta_i]$, representing the combination of the vector endpoints of equation 5.1 and their corresponding angles of equation 5.2. This operation entails iterative refinement of cluster centroids to minimise the within-cluster sum of squares, which ultimately results in the partitioning of the data into K distinct clusters. During the clustering process,

each variable is assigned to the closest centroid of the cluster based on the matrix **F**. The centroids are then iteratively updated to reflect the mean position of the variables assigned to each cluster. This iterative process continues until convergence, where the cluster assignments and centroids stabilise.

The next subsection demonstrates the functionality of the proposed algorithm using two well-known real datasets: the *Iris* dataset and the *mtcars* dataset. These datasets are chosen for their distinct characteristics; the *Iris* dataset comprises a relatively small number of variables, while the *mtcars* dataset features a larger and more complex structure. This contrast allows for a comprehensive demonstration of the algorithm's capabilities across varying data complexities.

5.1 Illustrations

In this subsection, the proposed algorithm is applied to both datasets, demonstrating its effectiveness in handling data of varying sizes and complexities. The illustrations presented underscore the versatility and robustness of the algorithm in real-world scenarios.

5.1.1 The *Iris* data

Consider the *Iris* dataset, which includes four variables: *Sepal.Length*, *Sepal.Width*, *Petal.Length*, and *Petal.Width*. Figure 5.2 presents the PCA biplot of the *Iris* data, where an overlap is observed between the vectors representing the *Petal.Width* and *Petal.Length* variables.

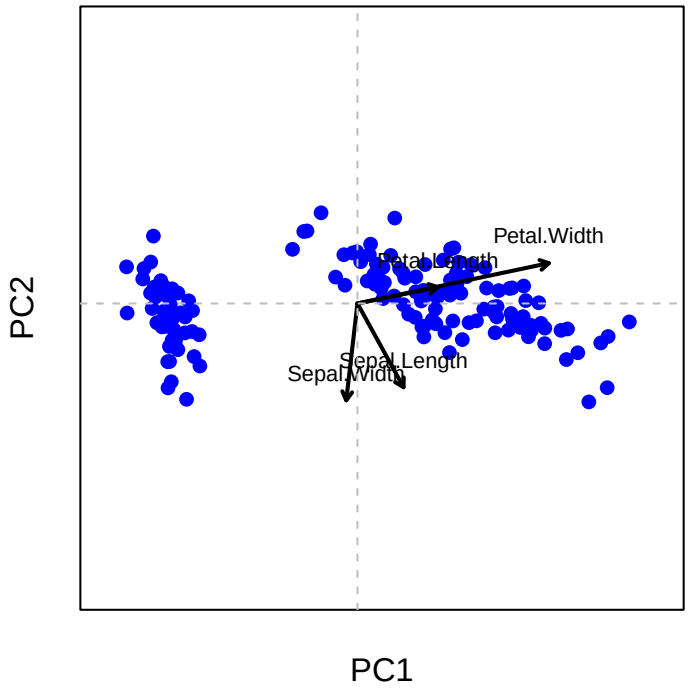

Figure 5.2: The PCA biplot of the *Iris* data displaying four variables of which two of them overlap.

To address this issue, it is essential to provide a clearer view of the vectors. For this purpose, Figure 5.3 illustrates the vectors within a unit circle, with the observations excluded from the plot for better clarity.

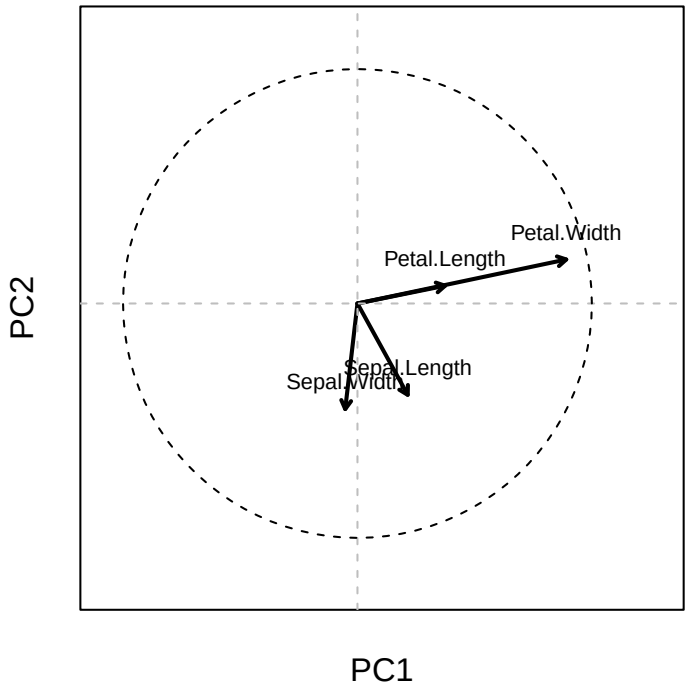

Figure 5.3: The PCA biplot of the *Iris* data displays original variables as vectors and includes a unit circle around the vectors.

Representing the variables as vectors within the unit circle offers valuable insight into the adequacy of each variable. This adequacy is the diagonal values of the product of the $\mathbf{V}_2$ matrix and its transpose. Thus, the adequacy of each vector shown in Figure 5.3 is summarised in Table 5.1:

Table 5.1: Variables adequacy

Variables	**Adequacy**
Sepal.Width	0.540
Sepal.Length	0.562
Petal.Length	0.764
Petal.Width	0.130

The adequacy values in Table 5.1 reveal varying levels of representation in the reduced-dimensional space. *Petal.Length* (0.764) is well represented and likely contributes valuable information, while *Sepal.Width* (0.540) and *Sepal.Length* (0.562) show moderate representation. In contrast, *Petal.Width* (0.130) has a low adequacy, suggesting that it may be less informative for analysis.

5.1.2 The application of *k*-means algorithm to the *Iris* dataset

In Figure 5.3, the overlap between the vectors representing *Petal.Length* and *Petal.Width* is particularly evident. The approach suggested by Sandilands (2020) is effective here, as it involves moving the vectors parallel to their original positions within the unit circle to enhance visibility. However, when dealing with a large number of variables in the dataset, this method can lead to an overcrowding of translated vectors within the unit circle.

In these instances, clustering of variables can be helpful. The first step involves inputting the endpoints of each vector from equation 5.1 into equation 5.2 to calculate the angles, summarised in Table 5.2.

Table 5.2: Coordinates and angles of variable vectors in the biplot of the *Iris* data

Variables	$v_{i,x}$	$v_{i,y}$	θ_i
Sepal.Length	0.361	-0.657	151.213°
Sepal.Width	-0.085	-0.730	−173.358°
Petal.Length	0.857	0.173	78.587°
Petal.Width	0.358	0.075	78.168°

Upon reviewing Table 5.2, it is observed that the angles formed by the vectors representing the *Petal.Length* and *Petal.Width* variables are similar in magnitude and direction. This similarity confirms the overlap observed in Figure 5.3 and suggests a common impact or shared influence between these variables. Consequently, clustering of these variables is warranted in this example.

Therefore, the k-means algorithm, as defined in equation 5.3, uses two key parameters: The first parameter consists of the vector endpoints and their corresponding angles, as shown in Table 5.2; the second parameter is K, the number of clusters specified by the user.In this case, the choice of K is 3, following the well-established convention of clustering the Iris dataset into three distinct groups. The results of this clustering process are presented in Table 5.3, along with the visualisation in Figure 5.4.

Table 5.3: Clusters within the *Iris* dataset

Cluster	**Variables**	$v_{i,x}$	$v_{i,y}$
1	Sepal.Width	-0.085	-0.730
2	Sepal.Length	0.361	-0.657
3	Petal.Length, and Petal.Width	0.607	0.124

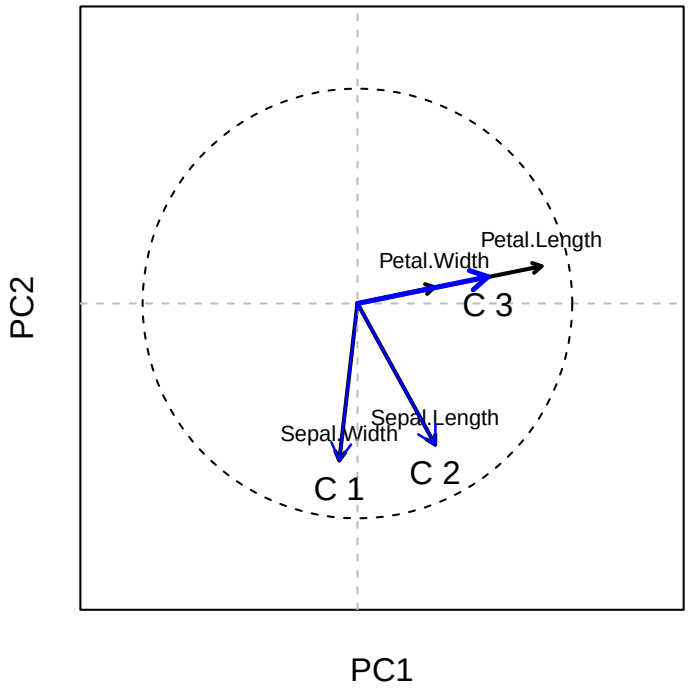

Figure 5.4: The PCA biplot illustrating clusters of vectors within a unit circle, where the vectors of *Petal.Length* and *Petal.Width* are grouped together in cluster C3.

The results in Table 5.3 are visually represented in Figure 5.4. This table provides the endpoints and corresponding angles for the vectors of clusters C1, C2, and C3. In particular, the information for the clusters C1 and C2 vectors remains the same as that of the *Sepal.Width* and *Sepal.Length* vectors, respectively, as presented in Table 5.2. This is also illustrated by the vectors (clusters) coloured in blue in Figure 5.4, which are identical in both size and direction to the *Sepal.Width* and *Sepal.Length* vectors in Figure 5.3. This consistency indicates that these variables were not grouped or clustered with other variables. In contrast, the endpoints and angles of *Petal.Length* and *Petal.Width* in Table 5.2 differ from those in Table 5.3. However, these two vectors now share the

same endpoints and angles, indicating that they have been grouped into a single cluster C3. The endpoints of C3 are determined by averaging the intercepts of *Petal.Length* and *Petal.Width* separately: the x-intercepts are averaged together, and the y-intercepts are averaged together.

To enhance clarity, the clusters in Figure 5.4 are visualised without the original vectors, as shown in Figure 5.5.

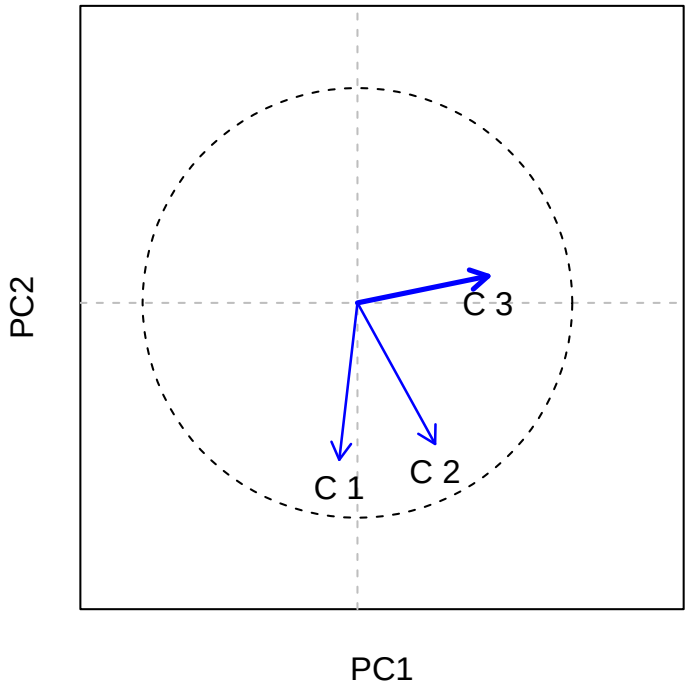

Figure 5.5: The PCA biplot illustrating clusters of vectors within a unit circle, excluding the original vectors.

In Figure 5.5, it is evident that cluster C3, representing the *Petal.Length* and *Petal.Width* variables, appears thicker than clusters C1 and C2, which represent *Sepal.Width* and

Sepal.Length, respectively. This observation confirms that cluster C3 consists of a denser grouping of variables compared to clusters C1 and C2. Subsection 5.2 explains the methodology for determining cluster vector thickness based on the number of variables within the cluster.

The adequacy values for each cluster vector, which indicate how well each cluster is represented in the biplot, are presented in Table 5.4.

Table 5.4: Adequacy of the clustered vectors in the *Iris* dataset

Cluster	**Variables**	**Adequacy**
1	Sepal.Width	0.540
2	Sepal.Length	0.562
3	Petal.Length, and Petal.Width	0.384

The adequacy values presented in Table 5.4 show distinct differences compared to the individual variables in Table 5.1. Cluster C1, with an adequacy of 0.540, corresponds to the same level of representation as *Sepal.Width*. Similarly, cluster C2, with an adequacy of 0.562, maintains the same level as *Sepal.Length*. In contrast, cluster C3, which includes *Petal.Length* (0.764) and *Petal.Width* (0.130), achieves a combined adequacy of 0.384. Although the overall adequacy is relatively lower compared to other clusters, the stronger influence of variables with higher adequacy, such as *Petal.Length*, compensates for the weaker contribution of *Petal.Width*. As a result, each variable within the clusters in the biplot is represented fairly well, maintaining a balanced representation comparable to that of the other variables.

5.1.3 The *mtcars* data

This subsection explores a dataset with a large number of variables: the *mtcars* dataset. This dataset, widely recognised in statistical and data analysis contexts, details various characteristics of cars from the 1973-1974 model years. The dataset encompasses a range of attributes, which are visualised in Figure 5.6.

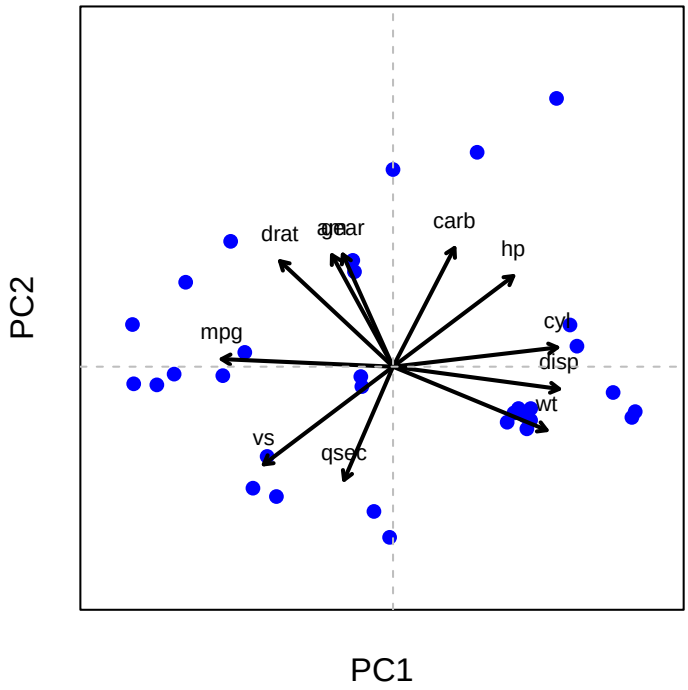

Figure 5.6: The PCA biplot of the *mtcars* data.

Figure 5.6 displays 11 variables as vectors along with observations within the biplot. The endpoints, angles, and adequacy of these variables are provided in Table 5.5.

Table 5.5: The endpoints, angles and the adequacy of the variables in the *mtcars* dataset.

Variable	$v_{i,x}$	$v_{i,y}$	θ_i	**Adequacy**
mpg	-0.363	0.016	$-87.476°$	0.132
cyl	0.374	0.044	$83.290°$	0.142
disp	0.368	-0.049	$97.584°$	0.138
hp	0.330	0.249	$52.964°$	0.171
drat	-0.294	0.275	$-46.913°$	0.162
wt	0.346	-0.143	$112.455°$	0.140
qsec	-0.200	-0.463	$-156.637°$	0.254
vs	-0.307	-0.232	$-127.078°$	0.148
am	-0.235	0.429	$-28.713°$	0.239
gear	-0.207	0.462	$-24.135°$	0.256
carb	0.214	0.414	$27.335°$	0.217

Figure 5.7, provides the same plot displayed in Figure 5.6 , but with the vectors positioned inside a unit circle and without observations for clarity.

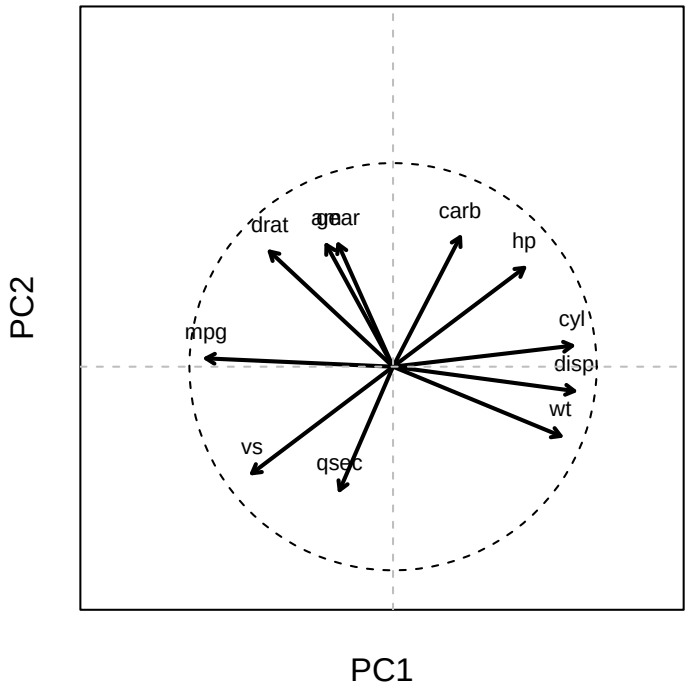

Figure 5.7: The PCA biplot of the *mtcars* data displaying the vectors of all 11 variables within a unit circle

The variable vectors displayed in Figure 5.7 are clustered into 4, 5, 6, and 7 groups, as shown in Figures 5.8, 5.9, 5.10, and 5.11, respectively. This was done to explore how varying the number of clusters (K) influences the grouping of variables in the *mtcars* dataset.

While multiple cluster configurations (e.g., 4, 5, 6, or 7) are possible, the 5-cluster solution is selected for illustrative purposes in this study. This choice allows for demonstrating the clustering process and its integration into the proposed methodology. No formal justification is provided for the selection of $K = 5$, and future work may consider more rigorous methods for determining the optimal number of clusters. Common approaches include

the Gap Statistic by Tibshirani et al. (2001), Scree Plots or the Elbow Method Ketchen and Shook (1996), Dunn's Index by Dunn (1974), and Silhouette Analysis by Rousseeuw (1987), each of which provides insights into the quality and stability of clustering results.

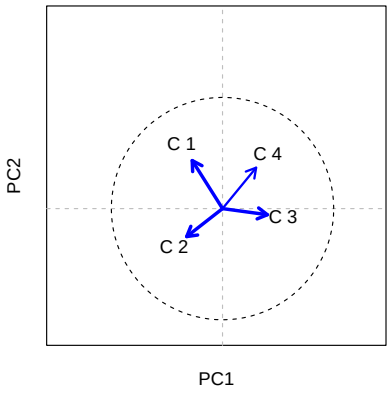

Figure 5.8: The PCA biplot displays 4 clusters from the *mtcars* dataset within the unit circle.

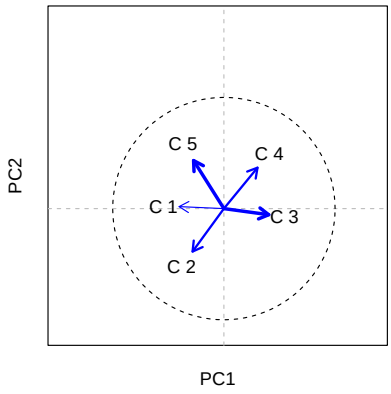

Figure 5.9: The PCA biplot displays 5 clusters from the *mtcars* dataset within the unit circle.

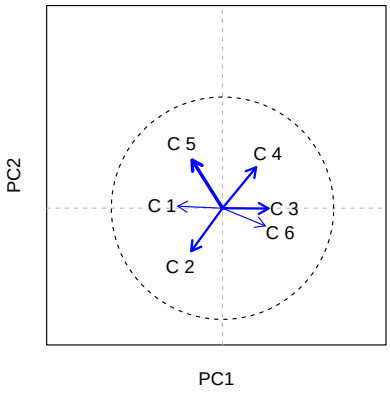

Figure 5.10: The PCA biplot displays 6 clusters from the *mtcars* dataset within the unit circle.

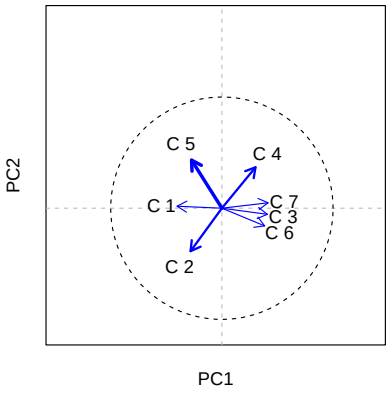

Figure 5.11: The PCA biplot displays 7 clusters from the *mtcars* dataset within the unit circle.

5.1.4 The application of *k*-means algorithm to the *mtcars* dataset

Using the endpoints along with their corresponding angles for each variable provided in Table 5.5, the vectors can be effectively clustered, as illustrated in Table 5.6 and Figure 5.12.

Table 5.6: Clusters identified within the *mtcars* dataset for $K = 5$

Cluster	**Variables**	$v_{i,x}$	$v_{i,y}$
1	mpg	-0.363	0.016
2	qsec, and vs	-0.254	-0.348
3	cyl, disp, and wt	0.363	-0.049
4	hp, and carb	0.272	0.332
5	drat, am, and gear	-0.245	0.389

Table 5.6 details the clusters identified with the chosen $K = 5$, and Figure 5.12 (equiva-

lently Figure 5.9) displays these clusters on the PCA biplot.

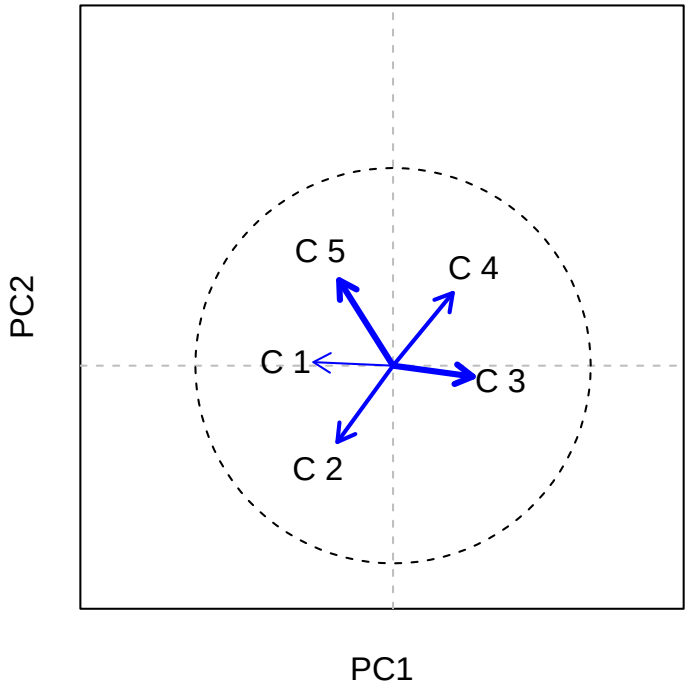

Figure 5.12: The PCA biplot representing the clusters within the *mtcars* dataset without the original vectors in the unit circle.

The solution presented in Figure 5.12 offers a clearer representation of the data, displaying only a few vectors within the unit circle, compared to the original biplot in Figure 5.7, which includes all 11 variables from the `mtcars` dataset. This approach also provides a clearer representation compared to the automatically translated axes shown in Figure 4.1 by Sandilands (2020), where all 11 variables are included as translated axes, potentially contributing to visual complexity. By reducing the number of displayed variables, this solution simplifies the visualisation and enhances the interpretability of complex data.

The adequacy values of the cluster vectors shown in Figure 5.12 are found in Table 5.7.

Table 5.7: Adequacy of the clustered vectors in the *mtcars* dataset

Cluster	**Variables**	**Adequacy**
1	mpg	0.132
2	qsec, and vs	0.186
3	cyl, disp, and wt	0.134
4	hp, and carb	0.184
5	drat, am, and gear	0.211

The variables demonstrating higher adequacy in Table 5.5 experience a slight decrease in adequacy within the clusters in Table 5.7. In contrast, the variables that initially showed lower adequacy in Table 5.5 show a marked improvement in their adequacy values once clustered. This phenomenon indicates that clustering can enhance the representation of certain variables by allowing them to benefit from the collective patterns observed within the clusters.

5.2 Cluster size adjustment based on variable count

In this study, visual clarity in the biplot is improved (see Figures 5.5 and 5.12). The proposed method not only clusters the vectors but also adjusts the relative sizes (thickness) of the clusters, reflecting the number of variables assigned to each cluster. For example, let n_j denote the number of variables in cluster j. The total number of variables across all clusters can be expressed as:

$$n_{\text{total}} = \sum_{j=1}^{k} n_j. \tag{5.4}$$

The size of each cluster is then adjusted proportionally to its share of the total count of variables. Specifically, the size adjustment for cluster j is given by:

$$S_j = \frac{n_j}{n_{\text{total}}} N, \tag{5.5}$$

where S_j is the adjusted size of the cluster j, and N is a positive scaling factor provided by the user to control the overall size of all the clusters. The higher the value of N, the thicker the clusters appear. S_j ensures that the size of the cluster j is proportional to its fraction of the total count of variables. By scaling each cluster in this way, clusters with more variables are represented with a larger size, while those with fewer variables are scaled down. This allows for a visual representation that accurately reflects the contribution of each cluster to the total number of variables.

5.3 Summary

In this chapter, the methodology using a novel method was introduced to address the challenge of overcrowding in biplots, particularly when visualising a large number of variables. The conventional methods, as discussed in the previous section, often fall short when applied to extensive datasets, as they can lead to cluttered visualisations that are difficult to interpret. To overcome this limitation, a solution is proposed that leverages clustering of k-means to group variables with similar directions and positions, thus reducing visual clutter and improving the interpretability of biplots.

Through practical illustrations using the *Iris* and *mtcars* datasets, the effectiveness of the proposed approach is demonstrated. The *Iris* and *mtcars* dataset shows how similar variables based on their configuration within the biplot, can be clustered together for a clearer view. The next section uses a larger dataset showcasing the applicability of the methodology.

6 Data analysis and Results

6.1 Introduction

This section is divided into two parts. The first part compares the proposed methodology to an alternative approach, that is, clustering variables in their raw form independent of how they are displayed in the biplot. To facilitate this comparison, the correlation matrix of the raw dataset is calculated to identify variables that are highly positively correlated, suggesting that they should be clustered together, and those that are less correlated, indicating that they should not be clustered.

The second part applies the proposed methodology to cluster variables in the biplot of a large dataset containing a very large number of variables. The complexity of this dataset poses significant challenges for traditional biplot techniques, as the high dimensionality can lead to overcrowding and overlapping of variables, ultimately hindering the effective interpretation of relationships.

6.2 Alternative approach: Pre-clustering

Unlike the proposed methodology outlined in Chapter 5, which clusters variables based on their biplot representation by considering the endpoint positions and angular relationships of the variables, the pre-clustering approach first groups the variables in the original data space prior to constructing the biplot.

Consider the data matrix $\mathbf{X}$, with n observations and p variables. To cluster the n observations, the k-means algorithm is applied to the data matrix $\mathbf{X}$, along with the number of clusters chosen by the user. To cluster the p variables, the transpose of the data matrix, denoted $\mathbf{X}'$, is used, and the k-means algorithm is applied to this transposed matrix.

The alternative approach is demonstrated using the *Iris* and *mtcars* datasets. The results presented in Tables 6.1 and 6.2 were obtained using a random seed set to 123 in R.

Table 6.1: Clusters of the raw *Iris* data

Cluster	**Variables**
1	Sepal.Width, and Petal.Length
2	Petal.Width
3	Sepal.Length

Table 6.2: Clusters of the raw *mtcars* data

Cluster	**variables**
1	disp, and hp
2	vs, and am
3	mpg, and qsec
4	cyl
5	gear, carb, drat, and wt

The following subsection provides a comparative assessment of the proposed methodology and the alternative (pre-clustering) approach. Using the correlation matrix, the performance of both methods is evaluated to determine which one is more effective in clustering variables based on their underlying characteristics.

6.2.1 Correlation matrix analysis of *Iris* dataset

Consider the correlation matrix of the *Iris* dataset presented in Table 6.3. This matrix provides valuable information on the relationships between the variables, indicating how strongly they correlate with each other.

Based on the correlation matrix for the *Iris* dataset, the two variables that exhibit the highest correlation are *Petal.Length* and *Petal.Width*, with a correlation coefficient of $r = 0.963$. This strong positive correlation indicates that as the length of the petals increases, the width of the petals also tends to increase. Given this significant correlation,

Table 6.3: Correlation matrix for *Iris* dataset

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1.000	-0.118	0.872	0.818
Sepal.Width	-0.118	1.000	-0.428	-0.366
Petal.Length	0.872	-0.428	1.000	0.963
Petal.Width	0.818	-0.366	0.963	1.000

these two variables should be clustered together, as this will enhance the interpretability of the dataset.

The results of the proposed methodology, shown in Table 5.3, confirm that these variables are appropriately clustered. In contrast, the results of the pre-clustering method, presented in Table 6.1, indicate that *Sepal.Width* and *Petal.Length*, which exhibit a negative correlation coefficient of $r = -0.428$, are clustered together. This clustering choice is not optimal, as it does not reflect the underlying relationships indicated by the correlation matrix. Thus, the proposed methodology demonstrates its effectiveness in clustering the *Iris* dataset by identifying meaningful variable relationships and grouping them accordingly.

6.2.2 Correlation matrix analysis of the *mtcars* dataset

A correlation matrix analysis of the *mtcars* dataset presented in Table 6.4 is used for a further comparative evaluation of two cluster assignment approaches. This evaluation is based on the clustering results of both approaches, which are presented in Tables 5.6 and 6.2 for the proposed methodology and pre-clustering method, respectively. The comparison focuses on assessing the correlations between variables within each cluster for both approaches.

Starting with the results of the pre-clustering method, weak correlations exist between some of the variables grouped using this approach. For example, in cluster 2, the variables *vs* and *am* exhibit a weak positive correlation of $r = 0.168$. Furthermore, within cluster 5, which comprises the variables *gear*, *carb*, *drat*, and *wt*, a weak positive correlation of

$r = 0.274$ is observed between *gear* and *carb*. Furthermore, *carb* and *drat* show a weak negative correlation of $r = -0.091$.

However, the proposed methodology yields clusters with strong positive correlations. For example, cluster 2, comprising *qsec* and *vs*, exhibits a strong positive correlation of $r = 0.745$ between these variables. Cluster 3, consisting of *cyl*, *disp* and *wt*, shows very strong positive correlations of $r = 0.902$ between *cyl* and *disp*, $r = 0.782$ between *cyl* and *wt*, and $r = 0.888$ between *disp* and *wt*. Cluster 4, consisting of *hp* and *carb*, also has a strong positive correlation of $r = 0.750$. Lastly, cluster 5, comprising *drat*, *am* and *gear*, reveals strong positive correlations, with $r = 0.713$ between *drat* and *am*, $r = 0.700$ between *drat* and *gear*, and $r = 0.794$ between *am* and *gear*.

Therefore, based on the analysis, it is evident that the proposed methodology outperforms the pre-clustering method in identifying meaningful variable relationships. The proposed methodology yields clusters with strong positive correlations, indicating a higher degree of association between the variables within each cluster. In contrast, the pre-clustering method results in clusters with weak correlations, suggesting a lack of strong relationships between the variables. Therefore, the proposed methodology demonstrates its effectiveness in clustering variables based on their underlying relationships, making it a more reliable approach for exploratory data analysis.

The following section showcases the effectiveness of the proposed methodology in clustering a large and complex dataset, demonstrating its capacity to efficiently handle big data. Using a more intricate dataset, this section illustrates the methodology's robustness, scalability, and potential for tackling datasets of increased size and complexity. This real-world application provides valuable insight into the methodology's performance in scenarios where data volume and complexity pose significant challenges.

Table 6.4: Correlation matrix for *mtcars* dataset

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
mpg	1.000	-0.852	-0.848	-0.776	0.681	-0.868	0.419	0.664	0.600	0.480	-0.551
cyl	-0.852	1.000	0.902	0.832	-0.700	0.782	-0.591	-0.811	-0.523	-0.493	0.527
disp	-0.848	0.902	1.000	0.791	-0.710	0.888	-0.434	-0.710	-0.591	-0.556	0.395
hp	-0.776	0.832	0.791	1.000	-0.449	0.659	-0.708	-0.723	-0.243	-0.126	0.750
drat	0.681	-0.700	-0.710	-0.449	1.000	-0.712	0.091	0.440	0.713	0.700	-0.091
wt	-0.868	0.782	0.888	0.659	-0.712	1.000	-0.175	-0.555	-0.692	-0.583	0.428
qsec	0.419	-0.591	-0.434	-0.708	0.091	-0.175	1.000	0.745	-0.230	-0.213	-0.656
vs	0.664	-0.811	-0.710	-0.723	0.440	-0.555	0.745	1.000	0.168	0.206	-0.570
am	0.600	-0.523	-0.591	-0.243	0.713	-0.692	-0.230	0.168	1.000	0.794	0.058
gear	0.480	-0.493	-0.556	-0.126	0.700	-0.583	-0.213	0.206	0.794	1.000	0.274
carb	-0.551	0.527	0.395	0.750	-0.091	0.428	-0.656	-0.570	0.058	0.274	1.000

6.3 Clustering large, complex datasets

6.3.1 The *Colon* data

The *Colon* dataset obtained from the plsgenomics R package (Boulesteix et al., 2024) contains gene expression data for 2000 genes in 62 *Colon* tissue samples. It includes 40 tumor samples (coded 2) and 22 normal samples (coded 1). The dataset is structured as follows:

1. **X**: A 62×2000 matrix where rows represent samples and columns represent genes.
2. **Y**: A numeric vector of length 62 indicating the tissue type (tumor or normal).
3. **gene.names**: A vector with the names of the 2000 genes.

The dataset sample is visually represented in the following series of biplots, each showing a subset of the 2000 genes (variables, displayed as calibrated axes with marker points) and 62 tissues (observations). The first biplot displays 20 variables, focusing on a smaller gene selection to provide a clearer view of the data structure. The second biplot extends this to 40 variables, while the third and fourth biplots expand to 60 and 80 variables, respectively, incorporating more gene expression data. These subsets are visualised to avoid overcrowding that occurs when plotting all 2000 variables at once. As shown in the four biplots, increasing the number of variables leads to a progressively more cluttered representation.

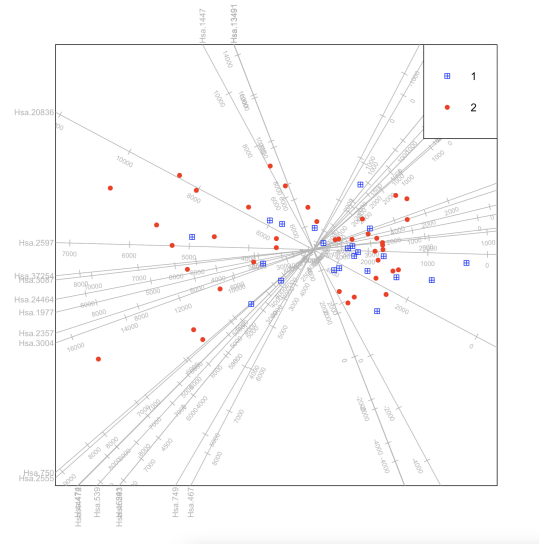

Figure 6.1: The PCA biplot presenting the first 20 variables of the *Colon* data.

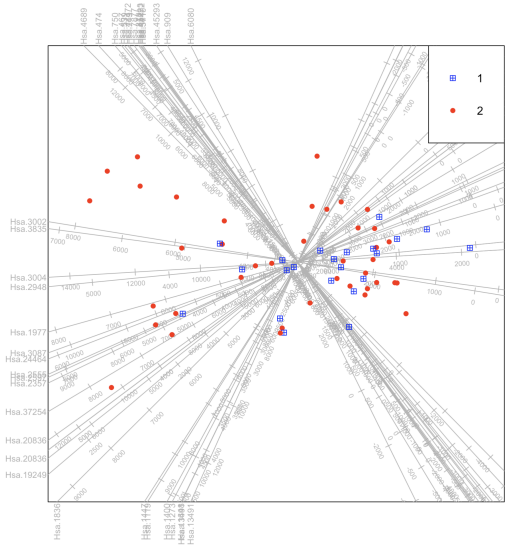

Figure 6.2: The PCA biplot presenting the first 40 variables of the *Colon* data.

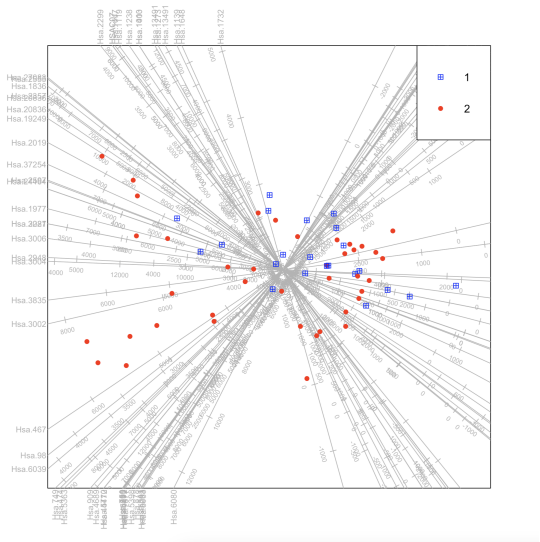

Figure 6.3: The PCA biplot presenting the first 60 variables of the *Colon* data.

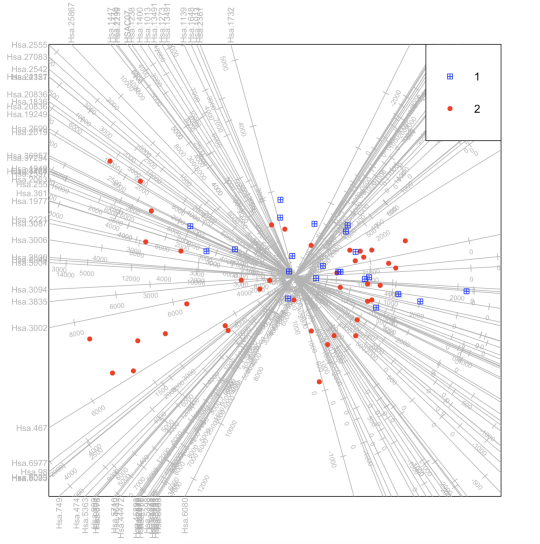

Figure 6.4: The PCA biplot presenting the first 80 variables of the *Colon* data.

6.3.2 Analysis and results of the data

The proposed methodology was applied to cluster all variables in the *Colon* dataset into $K = 7$ clusters. As with the examples in the proposed methodology, this choice is for illustration purposes, and the resulting clusters are shown in Figure 6.5.

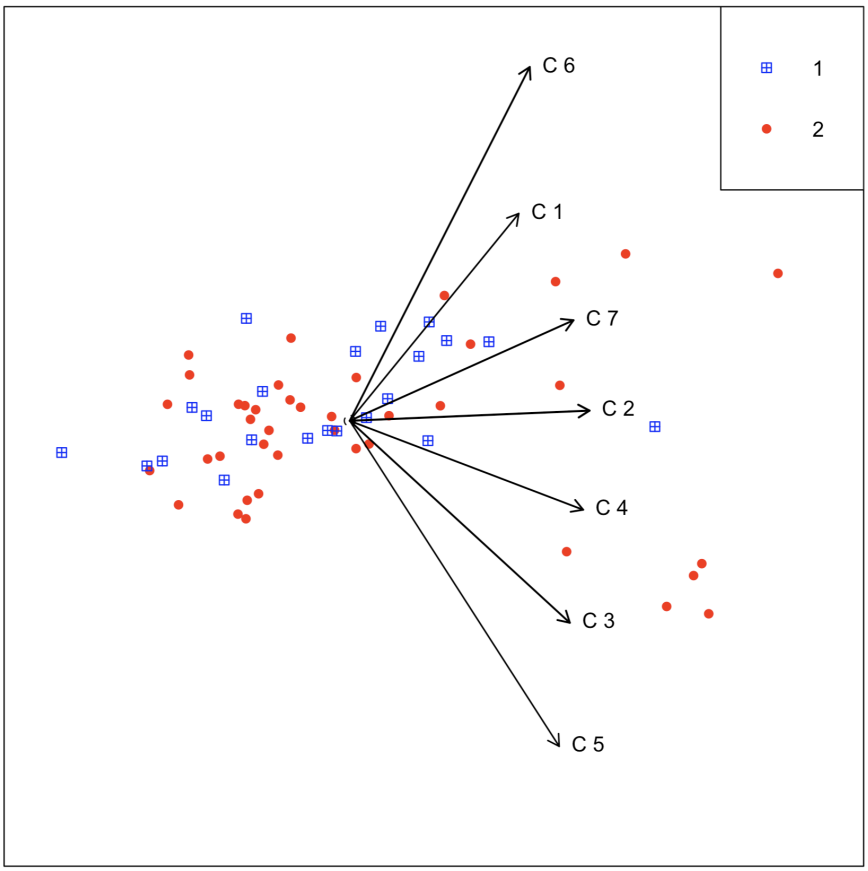

Figure 6.5: The PCA biplot displaying the clusters within the of the *Colon* data.

For this data, the clustered vectors each contain around 300 variables, and that is why they each have the same thickness. The direction of the vector indicates a positive direction, i.e., where the value of that variable is increasing.

The two groups, represented by red dots and blue squares, show distinct distributions in relation to these vectors. The normal group appears to spread further along certain

directions, while the tumor group remains more clustered around the origin. This suggests that the red group has more variation or higher values in the directions indicated by the vectors.

There is an observation in the normal group located near the arrow for C2, indicating that this observation has a relatively high value for C2 compared to the other observations in the normal group. Since the vectors represent variable directions, this suggests that this specific observation aligns more strongly with the C2 cluster than the rest of its group.

This observation stands out because most of the blue squares are more clustered around the center, whereas this one is positioned further along the C2 axis. This could suggest that, while a normal sample generally does not score high on C2, this particular observation is an exception, possibly an outlier or a special case within the dataset.

Using a biplot with clustered vectors enables the analysis of relationships between the variables as well as the interactions between these variables and individual observations or groups of observations.

6.3.3 Comparison with the automatic axis translation approach

This subsection compares the proposed methodology—which clusters variables within the biplot with the automatic axis translation approach introduced by Sandilands (2020). Although Sandilands (2020)'s approach produces useful results when dealing with datasets having a relatively small number of variables, it is clear that their method would be ineffective when dealing with large datasets, such as the *Colon* data with 2000 variables. Therefore, in general, the comparison demonstrates that the proposed methodology is more effective in managing large datasets.

6.4 Summary

This chapter compares the alternative pre-clustering method of clustering variables based on their raw form, independent of their representation in the biplot, to the proposed method, which clusters variables within the biplot. The alternative approach applies the k-means algorithm to the transpose of the raw data. The *Iris* and *mtcars* datasets are

used to illustrate and showcase the functionality of this method. A correlation matrix analysis is then used to compare these methods. The proposed methodology outperforms pre-clustering by forming groups with stronger internal correlations, leading to more meaningful relationships. The final section applies the proposed methodology to a large dataset, showcasing its effectiveness in handling high-dimensional data while maintaining interpretability.

7 Conclusion

Biplots are a powerful tool for visualising multivariate data. However, when datasets have a very large number of variables, biplots can become overcrowded, making interpretation difficult. This study aimed to address this challenge by improving visual clarity within biplots, thus facilitating more effective data exploration and analysis.

This study focuses on continuous datasets. The initial step involved exploring dimension reduction techniques, specifically PCA, to reduce the data's dimensionality. By transforming high-dimensional data into a lower dimensional space, dimension reduction techniques simplify the data, making it easier to work with while preserving essential information.

To improve clarity within the biplot, this study considered unsupervised learning techniques, specifically clustering. Clustering is helpful in grouping similar data points together. By applying clustering techniques to the data, inference can be drawn about the underlying relationships between variables. Different types of clustering algorithms are discussed in Chapter 3, including k-means, reduced k-means, and factorial k-means.

Existing solutions were also explored in Chapter 4. In particular, the focus was on recent advances proposed by Sandilands (2020) which enable automatic translation of the axes within the biplot. However, this approach can also lead to overcrowding within the biplot, especially when dealing with large datasets, as the translated axes can also become overcrowded.

Therefore, the methodology proposed in Chapter 5 presents an alternative approach using the k-means clustering algorithm to address overcrowding in biplots. This method enables users to group similar variables by selecting the desired number of clusters. The resulting plot displays cluster vectors with varying thicknesses, where clusters comprising more variables appear thicker, and those with fewer variables appear thinner.

Chapter 6 presents a comparative analysis of the proposed methodology and the approach of clustering raw data independently of biplot visualisation. Additionally, this chapter showcases the effectiveness of the proposed methodology in clustering a large dataset with

2000 variables, generating impressive results.

Future studies can investigate methods for calibrating the axes representing the clusters, ensuring a more accurate representation and interpretation of the underlying data structure. Given that the current approach involves selecting the number of clusters K based on exploratory assessment for illustrative purposes, future work should consider objective methods such as average silhouette scores or the elbow method to support more robust and practical cluster selection. Other clustering algorithms may also be considered, and the application can be extended to other types of biplots, such as the Canonical Variate Analysis (CVA) biplot. CVA, introduced by Rao (1948), is a supervised dimension reduction technique that maximises the separation between predefined groups by projecting the data onto canonical variates—linear combinations of the original variables that best discriminate among the groups.

References

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750. https://doi.org/10.1073/pnas.96.12.6745.

Anderson, E. (1935). The irises of the gaspe peninsula. *Bulletin of American Iris Society*, 59:2–5. Available online: https://cir.nii.ac.jp/crid/1574231874659141760.

Barr, G. D., Underhill, L. G., and Kahn, S. B. (1990). The covariance biplot for graphical display of multivariate time series data. *American Journal of Mathematical and Management Sciences*, 10(1-2):1–15. https://doi.org/10.1080/01966324.1990.10737274.

Blasius, J., Eilers, P. H., and Gower, J. (2009). Better biplots. *Computational Statistics & Data Analysis*, 53(8):3145–3158. https://doi.org/10.1016/j.csda.2008.06.013.

Boulesteix, A.-L., Durif, G., Lambert-Lacroix, S., Peyre, J., and Strimmer., K. (2024). *plsgenomics: PLS Analyses for Genomics*. Available online: https://CRAN.R-project.org/package=plsgenomics.

Dang, X., Hu, X., Ma, Y., Li, Y., Kan, W., and Dong, X. (2024). Ammi and gge biplot analysis for genotype× environment interactions affecting the yield and quality characteristics of sugar beet. *PeerJ*, 12:e16882. Available online: https://peerj.com/articles/16882/.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104. Available online: https://doi.org/10.1080/01969727408546059.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218. https://doi.org/10.1007/BF02288367.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

Fodor, I. K. (2002). A survey of dimension reduction techniques. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States). Available online: https://cs.nju.edu.cn/_upload/tpl/01/0b/267/template267/zhouzh.files/course/dm/reading/reading03/fodor_techrep02.pdf.

Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, 58(3):453–467. https://doi.org/10.1093/biomet/58.3.453.

Gower, J. C. and Hand, D. J. (1996). *Biplots*. Chapman and Hall: London. ISBN 0-412-71630-5. Available online: https://ideas.repec.org/a/eee/csdana/v22y1996i6p655-651a.html.

Gower, J. C., Lubbe, S. G., and Le Roux, N. J. (2011). *Understanding Biplots*. John Wiley & Sons: Chichester. Available online: https://www.google.co.za/books/edition/Understanding_Biplots/66gQCi5JOKYC?hl=en&gbpv=0.

Greenacre, M. (2017). *Correspondence analysis in practice*. chapman and hall/crc. https://doi.org/10.1201/9781315369983.

Guevara-Viejó, F., Valenzuela-Cobos, J. D., Noriega-Verdugo, D., Garcés-Moncayo, M. F., and Basurto Quilligana, R. (2024). Application of biplot techniques to evaluate the potential of trichoderma spp. as a biological control of moniliasis in Ecuadorian cacao. *Applied Sciences*, 14(13):5481. https://doi.org/10.3390/app14135481.

Henderson, H. V. and Velleman, P. F. (1981). Building multiple regression models interactively. *Biometrics*, 37(2):391–411. https://doi.org/10.2307/2530428.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441. https://doi.org/10.1037/h0071325.

James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2023). *An introduction to statistical learning with application in R*. Springer: New York. Available online: https://www.statlearning.com/.

Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202. https://doi.org/10.1098/rsta.2015.0202.

Ketchen, D. J. and Shook, C. L. (1996). The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, 17(6):441–458. Available online: https://doi.org/10.1002/(SICI)1097-0266(199606)17:6⟨441::AID-SMJ819⟩3.0.CO;2-G.

La Grange, A., Le Roux, N., and Gardner-Lubbe, S. (2009). BiplotGUI: interactive biplots in R. *Journal of Statistical Software*, 30(12):1–37. https://doi.org/10.18637/jss.v030.i12.

Lubbe, S., Le Roux, N., Nienkemper-Swanepoel, J., Ganey, R., and Van der Merwe, C. (2023). biplotez: Ez-to-use biplots. *URL: https://CRAN. R-project. org/package= BiplotEZ. R Package Version*, 1. https://doi.org/10.32614/CRAN.package.biplotEZ.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA. Available online: http://projecteuclid.org/euclid.bsmsp/1200512992.

Osmond, C. (1985). Biplot models applied to cancer mortality rates. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 34(1):63–70. https://doi.org/10.2307/2347886.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572. https://doi.org/10.1080/14786440109462720.

R Core Team (2025). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. Available online: https://www.R-project.org/.

Rao, C. R. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203. Available online: https://www.jstor.org/stable/2983775.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65. Available online: https://doi.org/10.1016/0377-0427(87)90125-7.

Ruspini, E. H. (1969). A new approach to clustering. *Information and Control*, 15(1):22–32. https://doi.org/10.1016/S0019-9958(69)90591-9.

Sandilands, D. (2020). *Exploding biplots with density axes in Plotly*. PhD thesis, Stellenbosch: Stellenbosch University. Available online: https://scholar.sun.ac.za/items/5af86f31-0a2a-443c-83ef-0122f939db91.

Suryowati, K., JP, M. T., and Nasution, N. (2021). Application biplot analysis on mapping of non-convertive diseases in indonesia. *Parameter: Journal of Statistics*, 1(2):11–20. https://doi.org/10.22487/27765660.2021.v1.i2.15518.

ter Braak, C. J. (1983). Principal components biplots and alpha and beta diversity. *Ecology*, 64(3):454–462. Available online: http://pascal-francis.inist.fr/vibad/index.php?action=getRecordDetail&idt=PASCAL83X0482350.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the royal statistical society: series b (statistical methodology)*, 63(2):411–423. Available online: https://doi.org/10.1111/1467-9868.00293.

Timmerman, M. E., Ceulemans, E., Kiers, H. A., and Vichi, M. (2010). Factorial and reduced k-means reconsidered. *Computational Statistics & Data Analysis*, 54(7):1858–1871. https://doi.org/10.1016/j.csda.2010.02.009.

Tsagaroulis, P. (2020). *Data Visualization, Numeracy and Graph Literacy: Seeing and Thinking of Data Presented as Tables or Charts*. PhD thesis, The Chicago School of Professional Psychology. Available online: https://www.proquest.com/openview/0d121f049421059e712f7a7ef4bda3c3/1?pq-origsite=gscholar&cbl=44156.

Van den Honert, R. and Barr, G. (1988). Simultaneous representations of explanatory characteristics of mergers. *South African Journal of Business Management*, 19(4):161–169. https://doi.org/10.4102/sajbm.v19i4.987.

Van der Maaten, L., Postma, E., Van den Herik, J., et al. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71). Available online: https://members.loria.fr/moberger/Enseignement/AVR/Exposes/TR_Dimensiereductie.pdf.

Van Dyk, W. (2022). Visualising data through biplots using categorical pca and clustering. master in commerce dissertation, stellenbosch university. Available online: https://scholar.sun.ac.za/items/e880ef11-4296-4720-bbc0-7913a2c2cca8.

Vichi, M., Vicari, D., and Kiers, H. A. (2019). Clustering and dimension reduction for mixed variables. *Behaviormetrika*, 46(2):243–269. https://doi.org/10.1007/s41237-018-0068-6.

Yamamoto, M. and Terada, Y. (2014). Functional factorial k-means analysis. *Computational Statistics & Data Dnalysis*, 79:133–148. https://doi.org/10.1016/j.csda.2014.05.010.

Yan, W. and Tinker, N. (2006). Biplot analysis of multi-environment trial data: Principles and applications. *Can. J. Plant. Sci.*, 86:623–645. https://doi.org/10.4141/P05-169.

A R Simulated dataset

```r
# Set seed for reproducibility
set.seed(110)

# Simulate a dataset with 100 observations and 5 variables
data <- matrix(rnorm(100 * 5), ncol = 5)
colnames(data) <- paste0("Var ", 1:5)

# Generate a biplot object using the simulated data
obj1 <- biplot(data, scale = TRUE) |>
  PCA(group.aes = NULL) |>
  axes(which = NULL) |>
  plot()

# Add axis labels for the first two principal components
title(xlab = "PC1", ylab = "PC2", line = 1)
```

Simulated_dataset.R

B R Proposed methodology code

```r

# Load necessary libraries
library(biplotEZ)

# Function definition for clustering vectors close to each
# other in the biplot
cluster_close_vectors <- function(bp, k = K, centers = NULL){

  # Extract the vectors from the biplot object
  vectors <- bp$Vr

  # Calculate the angle of each vector in radians
  theta_radians <- atan2(vectors[,1], vectors[,2])

  # Convert the angle from radians to degrees
  theta_degrees <- theta_radians * (180 / pi)

  # Output the direction of each vector in degrees
  theta_degrees

  # Combine vectors and their angles into a feature matrix for
  # clustering
```

```
features <- cbind(vectors, theta_degrees)

# Perform K-means clustering
if (is.null(centers)) {
  cluster_result <- kmeans(features, centers = k)
} else {
  initial_centers <- features[centers, ]
  cluster_result <- kmeans(features, centers = initial_centers)
}

# Count the number of variables in each cluster
cluster_counts <- table(cluster_result$cluster)

# Variables to store the sum of centroids' coordinates
combined_centroid <- c(0, 0)

# Draw cluster centroids with line thickness based on the
# number of variables in each cluster
for (i in 1:k) {
  var_cluster_prop <- (cluster_counts / sum(cluster_counts)) * 50
  centroid <- cluster_result$centers[i, ]
  line_thickness <- var_cluster_prop[i]

  arrows(0, 0, centroid[1], centroid[2], col = "blue",
         lwd = line_thickness, length = 0.1)

  # Add labels to each cluster centroid
  text(centroid[1], centroid[2], labels = paste("C", i),
       pos = 1, col = "black")
}

# Draw a one-unit circle centered at the origin
draw.circle(0, 0, radius = 1, border = "black", lty = 2)

# Create a matrix of size p to store the new gradients
new_features <- cbind(cluster_result$centers
                      [cluster_result$cluster, ],cluster_result$cluster)
rownames(new_features) <- colnames(bp$X)

# Return the cluster assignments, and new gradients
bp$cluster <- list(cluster = cluster_result$cluster,
                   new_features = new_features,
                   centroids = cluster_result$centers)

# Return the updated biplot object
bp

# Create a list of variables assigned to each cluster
cluster_assignments <- split(rownames(new_features),
```

```
                                cluster_result$cluster)

  # Print the table-like output
  cat("Cluster assignments:\n")
  for (i in 1:k) {
    cat(paste("Cluster", i, ":", paste(cluster_assignments[[i]],
                                       collapse = ", "), "\n"))
  }
  return(cluster_assignments)
}

# Create the biplot using the dataset X and perform PCA on it
bp <- biplot(X, scale = TRUE) |> PCA()
|> axes(which = NULL) |> samples(which = NULL)
|> fit.measures() |> plot(0.0006)

# Add axis lines to the plot (vertical and horizontal at zero)
abline(v = 0, h = 0, lty = 2, col = "grey")

# Add axis labels for PC1 and PC2 to the biplot
title(xlab = "PC1", ylab = "PC2", line = 1)

# Display the quality of the biplot
bp$quality

# Call the clustering function and obtain the cluster
# assignments
cluster_assignments <- bp |> cluster_close_vectors()
```

Proposed_methodology.R