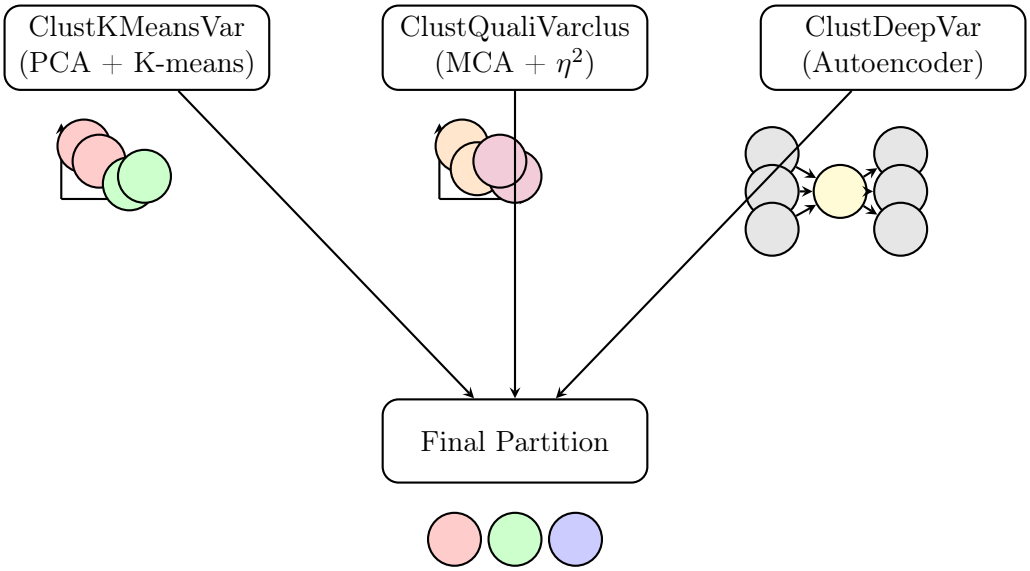
Université Lumière Lyon 2
Master in Statistics and Computer Science for Data Science (SISE)

ClustR

An R Package for Variable Clustering

Three Complementary Approaches: K-means, Qualitative, and Deep Learning



Authors:

Riad SAHRANE
Aya MECHERI
Thibaud LECOMTE

Supervisor: Ricco Rakotomalala

Academic Year 2024–2025
November 30, 2025

Abstract

This report presents **ClustR**, an innovative R package dedicated to **variable clustering**, whose purpose is to identify homogeneous groups of variables to improve interpretability, reduce dimensionality, and facilitate statistical and predictive modelling. Unlike classical clustering of observations, variable clustering focuses on relationships between *features*, requiring dedicated tools capable of capturing linear and nonlinear dependencies, as well as handling mixed data (numeric and categorical).

ClustR provides a **unified R6 interface**, inspired by the *scikit-learn* philosophy, enabling easy selection and comparison of complementary algorithms. The package also includes a **Shiny** application for interactive exploration, making the methodology accessible both to advanced users and non-programmers.

The three main algorithms implemented are:

- **ClustKMeansVar**: a reallocative method based on principal components, inspired by Vigneau & Qannari. It summarizes each cluster using a synthetic variable (PC1) and maximises a global separation criterion ($Q = B/T$).
- **ClustQualiVarclus**: a qualitative variable clustering method based on Multiple Correspondence Analysis (MCA) and the correlation ratio η^2.
- **ClustDeepVar**: a **deep clustering** algorithm using an autoencoder to project each variable into a nonlinear latent space, capturing complex relationships.

The resulting package offers a coherent toolkit capable of processing various real datasets, including correlated variables, categorical variables, and nonlinear structures. Experiments on both simulated and real datasets demonstrate the robustness and complementarity of the three methods.

Keywords: Variable clustering, R6, Autoencoder, MCA, PCA, Reallocative K-means, Factor analysis, Deep Learning, Shiny, R Package.

Contents

0.1 Introduction

0.1.1 Context and Motivation

Variable clustering is a multivariate analysis technique aiming to group **similar variables** into homogeneous clusters, in contrast to classical clustering which groups observations. This technique is particularly relevant in the following contexts:

- **Dimensionality reduction**: identifying groups of redundant variables to simplify analyses;
- **Variable selection**: choosing one representative variable per cluster to avoid multicollinearity;
- **Interpretation**: understanding the latent structure of data through coherent variable groups;
- **Feature engineering**: creating synthetic variables representing groups of correlated variables.

Despite its importance, existing variable clustering solutions in R suffer from several limitations:

- Lack of a unified interface (methods are dispersed across incompatible packages),
- Absence of deep learning–based methods,
- Difficulty handling mixed data (numeric + categorical),
- No automatic selection of the optimal number of clusters.

0.1.2 Objectives of the Project

This project aims to develop **ClustR**, a modern R package providing:

1. A **unified interface** using R6 classes (inspired by `scikit-learn`),
2. **Three complementary algorithms**:
   - Reallocative K-means (Vigneau & Qannari, 2003),
   - Qualitative VARCLUS (MCA + η^2),
   - Deep clustering using an autoencoder,
3. Robust **validation metrics** such as inertia, silhouette, and stability indices,
4. An automatic selection of the number of clusters based on a consensus of metrics,
5. A complete documentation with examples and a Shiny application.

0.1.3 Contributions

The main contributions of this work are:

- The **first R implementation** of an autoencoder dedicated to variable clustering;
- A **unified interface** enabling easy comparison of the three clustering methods;
- Advanced regularization techniques in ClustDeepVar (dropout, L2 penalty, early stopping);
- Projection of illustrative variables into the latent space for interpretation;
- Automatic selection of k based on a consensus of four internal clustering metrics.

0.2 State of the Art

0.2.1 General Principles of Variable Clustering

Variable clustering fundamentally differs from observation clustering. A comparison is shown below:

Aspect	Observations	Variables
Space	$\mathbb{R}^p$	$\mathbb{R}^n$
Typical distance	Euclidean	Correlation-based
Objective	Group individuals	Group features
Application	Segmentation	Dimensionality reduction

Table 1: Comparison of observation clustering vs. variable clustering

Distance Between Variables

For numerical variables, the most common distance is based on Pearson correlation:

$$d(X_j, X_k) = 1 - |\rho(X_j, X_k)|^2, \tag{1}$$

where ρ is the Pearson correlation coefficient.

For categorical variables, distances rely on measures such as:

- **Chi-square**: $\chi^2(V_1, V_2)$,
- **Cramér's V**:
$$V_C = \sqrt{\frac{\chi^2}{n \cdot \min(r-1, c-1)}},$$
- **Correlation ratio**:
$$\eta^2(V, Y) = \frac{\mathrm{Var}(\mathbb{E}[Y \mid V])}{\mathrm{Var}(Y)}.$$

0.2.2 Existing Methods

ClustOfVar (Chavent et al., 2012)

The main reference R package for variable clustering. It uses:

- Hierarchical clustering (CAH),
- Distance based on $1 - \rho^2$,
- Ward's criterion for aggregation,
- Mixed data support through FAMD.

Limitations:

- No soft clustering,
- Computational complexity $O(p^2 \log p)$,
- No projection of new variables into clusters.

VARCLUS (SAS)

A proprietary SAS algorithm based on:

- Divisive (top-down) clustering,
- Oblique PCA,
- Ratio $1 - R^2$ as split criterion.

Vigneau & Qannari (2003)

A K-means-like method adapted for variable clustering:

- Cluster centers represented by PC1 (synthetic variable),
- Distance: $1 - \text{cor}(X_j, y_g)^2$,
- Global criterion $Q = B/T$.

0.2.3 Deep Clustering

Autoencoder-based Clustering

Several deep clustering models exist:

- **DEC** — Deep Embedded Clustering (Xie et al., 2016),
- **DCEC** — Convolutional DEC (Guo et al., 2017),
- **VaDE** — Variational Deep Embedding (Jiang et al., 2017).

General workflow:

1. Pretrain autoencoder (reconstruction),
2. Extract latent embeddings,
3. Cluster embeddings (k-means / GMM),
4. Optionally fine-tune the model jointly.

Deep Learning Applied to Variable Clustering

Innovation of ClustR: Transpose the data matrix so that variables become "observations":

$$X \in \mathbb{R}^{n \times p} \quad \Longrightarrow \quad X^T \in \mathbb{R}^{p \times n}.$$

Each variable is then represented by a vector of its observed values across individuals.

0.2.4 Positioning of ClustR

Package	**Methods**	**Interface**	**Deep Learning**
ClustOfVar	Hierarchical, mixed	Classical R	No
VARCLUS (SAS)	Divisive, oblique PCA	Proprietary	No
ClustR	K-means, Qualitative, Deep	R6 + Auto-k	**Yes**

Table 2: Comparison of variable clustering packages

0.3 Architecture of the ClustR Package

0.3.1 Design Philosophy

ClustR follows an object-oriented architecture using the **R6** system (mutable classes), inspired by the `scikit-learn` design philosophy.

Key principles:

- **Unified interface**: all algorithms share identical methods (`initialize`, `fit`, `predict`, `get_clusters`);
- **Modularity**: the `VariableClustering` class encapsulates specific algorithms;
- **Extensibility**: new clustering algorithms can be added easily;
- **Reproducibility**: parameters and results are consistently stored within each model.

0.3.2 Class Diagram

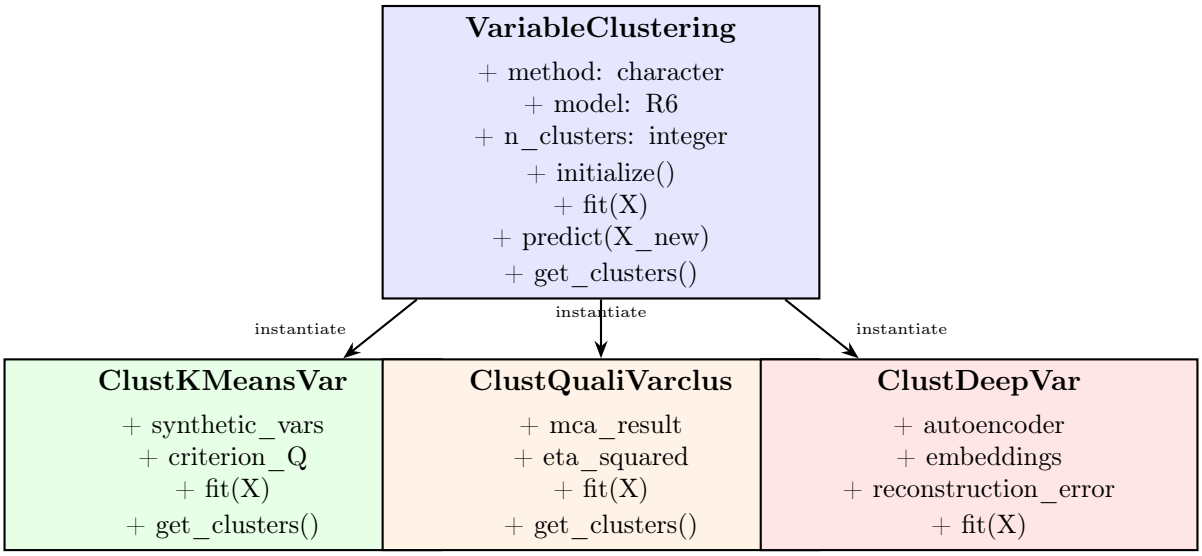


Figure 1: Simplified UML class diagram of ClustR

0.3.3 General Workflow

Algorithm 1 Typical Usage of ClustR

1: **Load** the dataset $X \in \mathbb{R}^{n \times p}$
2: **Select** the method: `"kmeans"`, `"quali_varclus"`, or `"deep"`
3: **Initialize** the model with its parameters
4: **Fit** the model on X
5: **Extract** clusters
6: **Validate** the clustering with internal metrics (inertia, silhouette, etc.)
7: **Visualize** results
8: **Export** cluster assignments and summaries

0.4 Reallocative K-means Algorithm

0.4.1 General Principle

The method of Vigneau & Qannari (2003) adapts classical K-means to variable clustering by introducing the concept of a **synthetic variable**, represented by the **first principal component (PC1)** of the variables in a cluster.

Intuition

In standard K-means, a cluster center is the mean of the observations. But for variable clustering, the "observations" are variables, so this is not meaningful.

Instead:

- Each cluster is represented by its **PC1**, called the synthetic variable y_g.
- A variable is assigned to the cluster g for which it is most correlated with y_g.

This mimics K-means behaviour while keeping interpretability.

0.4.2 Mathematical Formalism

Notations

- $X = [X_1, \ldots, X_p] \in \mathbb{R}^{n \times p}$: matrix of p variables and n observations;
- k: number of clusters;
- C_g: set of variable indices in cluster g;
- $y_g \in \mathbb{R}^n$: synthetic variable of cluster g (its PC1).

Step 1: Standardization (optional)

Variables may be centered and scaled:

$$X_{\text{std}} = \frac{X - \mu}{\sigma}. \tag{2}$$

Step 2: Synthetic Variable

For cluster C_g, the synthetic variable is:

$$y_g = X_{C_g} w_g, \tag{3}$$

where w_g is the eigenvector maximizing:

$$w_g = \text{argmax}_{\|w\|=1} \, w^T S_g w, \quad S_g = X_{C_g}^T X_{C_g}. \tag{4}$$

Thus, y_g summarizes the variables inside cluster g.

Step 3: Distance Between a Variable and a Cluster

The distance between variable X_j and cluster g is:

$$d(j, g) = 1 - \rho(X_j, y_g)^2, \tag{5}$$

where

$$\rho(X_j, y_g) = \frac{\text{Cov}(X_j, y_g)}{\sigma_{X_j}\sigma_{y_g}}.$$

Assignment rule:

$$X_j \to \text{argmin}_g\, d(j, g).$$

Step 4: Global Criterion Q

The algorithm maximizes the ratio:

$$Q = \frac{B}{T}, \tag{6}$$

with:

- T: total inertia,
- W: within-cluster inertia,
- B: between-cluster inertia.

Since $T = W + B$, a good clustering has:

$$Q \approx 1.$$

0.4.3 Full Algorithm

Algorithm 2 Reallocative K-means (Vigneau & Qannari)

Require: Data matrix $X \in \mathbb{R}^{n \times p}$, number of clusters k
Ensure: Partition $\mathcal{C} = \{C_1, \ldots, C_k\}$

Step 1: Standardization (optional)
$X \leftarrow (X - \mu)/\sigma$
Step 2: Initialization
Create an initial random partition of the p variables.
repeat
    Step 3: Compute synthetic variables
    for $g = 1$ to k **do**
        $y_g \leftarrow \text{PC1}(X_{C_g})$
    end for
    Step 4: Reallocation
    for $j = 1$ to p **do**
        Compute $d(j, g)$ for all clusters g
        Assign X_j to the cluster with minimum distance
    end for
    Step 5: Update quality
    Compute $Q = B/T$
until convergence of Q or max iterations reached
return $\mathcal{C}$

0.4.4 Computational Complexity

- PCA per cluster: $O(n \cdot |C_g|^2)$,
- Distance computations: $O(p \cdot k \cdot n)$,
- Total per iteration: $O(\frac{np^2}{k})$,
- Typical iterations: < 20.

Total complexity:

$$O(T \cdot \frac{np^2}{k}),$$

where T is the number of iterations.

0.4.5 Advantages and Limitations

Advantages

- **Interpretable**: synthetic variables = PCA axes;
- **Fast**: converges in fewer than 20 iterations;
- **Robust**: works well with correlated variables;
- **Global criterion**: Q allows comparing clusterings.

Limitations

- Sensitive to initialization (multiple runs advised);
- Captures only linear relationships (PCA-based);
- Only works for numerical variables;
- Requires k to be set a priori.

0.5 Qualitative VARCLUS Algorithm

0.5.1 General Principle

Clustering categorical variables relies on:

- **Multiple Correspondence Analysis (MCA)**,
- The **correlation ratio** η^2 for association strength.

0.5.2 Mathematical Formalism

Step 1: Implicit One-Hot Encoding

Categorical variables are internally transformed into disjunctive (dummy) variables:

$$X \Rightarrow Z \in \{0,1\}^{n \times m}.$$

Note: Using `FactoMineR::MCA`, one-hot encoding is automatic.

Step 2: MCA per Cluster

For each cluster of categorical variables, MCA extracts:

- eigenvalues,
- principal coordinates of categories,
- the first MCA axis, which synthesizes the cluster.

Step 3: Correlation Ratio η^2

For a qualitative variable V and an MCA axis Y:

$$\eta^2(V,Y) = \frac{\mathrm{Var}(\mathbb{E}[Y \mid V])}{\mathrm{Var}(Y)} \in [0,1]. \tag{7}$$

Interpretation:

- $\eta^2 \approx 1$: V is well explained by the cluster axis,
- $\eta^2 \approx 0$: independent from the cluster.

Step 4: Distance Between Categorical Variables

Equivalent to:

$$d(V_1, V_2) = 1 - \eta^2(V_1, V_2),$$

or theoretically:

$$d = 1 - V_C, \quad V_C = \text{Cramér's V}.$$

However, **the algorithm uses η^2 derived from MCA axes directly.**

0.5.3 Complete Algorithm

Algorithm 3 Qualitative VARCLUS

Require: Categorical variables X, number of clusters k
Ensure: Partition $\mathcal{C}$

Step 1: Initialization
Partition variables into k groups (random or based on Cramér's V).
repeat
    Step 2: MCA on each cluster
    for $g = 1$ to k **do**
        Compute MCA on X_{C_g}
        Extract the first axis Y_g
    end for
    Step 3: Reallocation
    for each categorical variable V_j **do**
        Compute $\eta^2(V_j, Y_g)$ for all clusters
        Assign V_j to cluster with highest η^2
    end for
until partition is stable or max iterations reached
Return $\mathcal{C}$

0.5.4 Advantages

- Designed specifically for categorical data,
- Uses MCA structure (chi-square geometry),
- Produces interpretable clusters,
- Works with nominal and ordinal variables.

0.6 Deep Clustering Algorithm

0.6.1 General Principle

ClustDeepVar uses an autoencoder to learn **nonlinear latent representations** of variables, followed by K-means clustering in the latent space.

0.6.2 Autoencoder Architecture

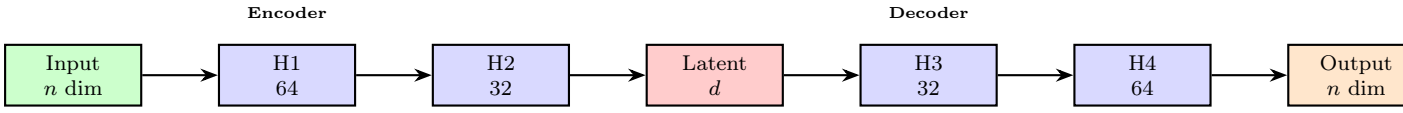


Figure 2: Example autoencoder architecture for ClustDeepVar

0.6.3 Mathematical Formalism

Step 1: Standardization

$$X_{\text{std}} = \frac{X - \mu}{\sigma}$$

Step 2: Transpose Matrix

Variables become "observations":

$$X^T \in \mathbb{R}^{p \times n}.$$

Step 3: Encoder — Latent Representation

$$Z = f_{\text{enc}}(X^T) \in \mathbb{R}^{p \times d}.$$

Step 4: Decoder — Reconstruction

$$\hat{X}^T = f_{\text{dec}}(Z).$$

Step 5: Loss Function

$$\text{Loss} = \text{MSE}(X^T, \hat{X}^T) + \lambda \|\theta\|_2^2.$$

Regularization:

- Dropout,
- L2 weight decay,
- Early stopping.

Step 6: Latent Clustering

$$\mathcal{C} = \text{k-means}(Z, k).$$

Step 7: Soft Clustering (optional)

$$P_{jk} = \frac{\exp(-d_{jk}/T)}{\sum_c \exp(-d_{jc}/T)}.$$

0.6.4 Complete Algorithm

Algorithm 4 ClustDeepVar — Deep Variable Clustering

Require: $X \in \mathbb{R}^{n \times p}$, k, latent dimension d
Ensure: Clusters, embeddings Z

1: Standardize X
2: Transpose X^T
3: Build encoder and decoder networks
4: Train autoencoder with MSE + L2 + dropout + early stopping
5: Extract embeddings Z
6: Cluster Z with k-means
7: Optionally compute soft clustering matrix P
8: **return** clusters, Z, P

0.6.5 Advantages

- Captures nonlinear relationships,
- Robust latent embeddings,
- Soft clustering available,
- Can project new variables,
- Scalable for large p.

0.7 Validation Metrics and Automatic Selection of k

0.7.1 Internal Metrics

Within-Cluster Inertia

$$W = \sum_{g=1}^{k} \sum_{j \in C_g} d(X_j, y_g).$$

Between-Cluster Inertia

$$B = \sum_{g=1}^{k} |C_g| \cdot d(\bar{y}_g, \bar{y}).$$

Criterion Q (K-means)

$$Q = \frac{B}{T}.$$

Silhouette

$$s_j = \frac{b_j - a_j}{\max(a_j, b_j)}.$$

0.7.2 Selecting the Number of Clusters

ClustR uses a **consensus of four metrics**:

1. Elbow,
2. Silhouette,
3. Calinski–Harabasz (CH),
4. Davies–Bouldin Index (DBI).

Consensus Algorithm

Algorithm 5 Consensus-based Selection of k

Compute k from elbow
Compute k maximizing silhouette
Compute k maximizing CH
Compute k minimizing DBI
$k^* = \text{mode}(\{k_1, k_2, k_3, k_4\})$
Confidence = $\#\text{votes}(k^*)/4$

0.8 Conclusion

0.8.1 Summary of Contributions

ClustR provides:

- a unified R6-based interface,
- a deep learning method for variable clustering,
- automatic selection of the optimal number of clusters,
- projection and visualization tools.

0.8.2 Comparison of the Three Algorithms

Criterion	**K-means**	**Qualitative**	**Deep**
Data type	Numerical	Categorical	Numerical
Relations	Linear	χ^2	Nonlinear
Interpretability	High (PC1)	Medium (MCA)	Low
Complexity	$O(np^2/k)$	$O(np^2)$	$O(\text{epochs} \cdot p \cdot d^2)$
Soft clustering	No	No	Yes
Projection	No	Yes	Yes

Table 3: Summary comparison of the three methods

0.8.3 Limitations

- K-means sensitive to initialization,
- Qualitative clustering only for categorical variables,
- Deep clustering requires tuning and is less interpretable.

Bibliography

[1] Vigneau, E., & Qannari, E. M. (2003). *Clustering of variables around latent components*. Communications in Statistics-Simulation and Computation, 32(4), 1131-1150.

[2] Chavent, M., Kuentz-Simonet, V., Labenne, A., & Saracco, J. (2012). *ClustOfVar: An R package for the clustering of variables*. Journal of Statistical Software, 50(13), 1-16.

[3] Xie, J., Girshick, R., & Farhadi, A. (2016). *Unsupervised deep embedding for clustering analysis*. In International conference on machine learning (pp. 478-487).

[4] Guo, X., Liu, X., Zhu, E., & Yin, J. (2017). *Deep clustering with convolutional autoencoders*. In International conference on neural information processing (pp. 373-382).

[5] Jiang, Z., Zheng, Y., Tan, H., Tang, B., & Zhou, H. (2017). *Variational deep embedding: An unsupervised and generative approach to clustering*. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (pp. 1965-1972).

[6] Zou, H., Hastie, T., & Tibshirani, R. (2006). *Sparse principal component analysis*. Journal of computational and graphical statistics, 15(2), 265-286.

[7] Husson, F., Lê, S., & Pagès, J. (2010). *Exploratory multivariate analysis by example using R*. CRC press.

[8] Chollet, F., et al. (2015). *Keras*. https://keras.io.

[9] 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.

[10] Caliński, T., & Harabasz, J. (1974). *A dendrite method for cluster analysis*. Communications in Statistics-theory and Methods, 3(1), 1-27.

[11] Davies, D. L., & Bouldin, D. W. (1979). *A cluster separation measure*. IEEE transactions on pattern analysis and machine intelligence, (2), 224-227.