

UNIVERSITÉ LUMIÈRE LYON 2

Master 2 SISE

Statistique et Informatique pour la Science des données

# Projet mmrClustVar

Clustering de variables et implémentation R6

**Auteurs :**

Marin NAGY

Mazilda ZEHRAOUI

Rina RAZAFIMAHEFA

**Encadrement :**

*Ricco RAKOTOMALALA*

Année universitaire : 2025–2026

# Table des matières

<b>1 Contexte et objectifs du projet</b>	<b>2</b>
<b>2 Cahier des charges et validation</b>	<b>3</b>
<b>3 Architecture du package</b>	<b>5</b>
<b>4 Méthodes de clustering implémentées</b>	<b>6</b>
4.1 k-means (numérique) . . . . .	6
4.2 k-modes (catégoriel) . . . . .	7
4.3 k-prototypes (mixte) . . . . .	8
4.4 k-medoids (dissimilarités mixtes) . . . . .	9
<b>5 Indicateurs et diagnostics</b>	<b>11</b>
<b>6 Visualisations</b>	<b>12</b>
<b>7 Application Shiny</b>	<b>13</b>
<b>8 Limites et perspectives</b>	<b>14</b>
<b>9 Conclusion</b>	<b>15</b>
<b>Bibliographie</b>	<b>15</b>

# Chapitre 1

## Contexte et objectifs du projet

Le projet `mmrClustVar` s'inscrit dans le cadre du module de programmation R avancée du Master 2 SISE. L'objectif est double :

- **pédagogique** : apprendre à concevoir un package structuré en R6, avec encapsulation, héritage, documentation et interface Shiny ;
- **méthodologique** : implémenter plusieurs algorithmes de *clustering de variables*, permettant de regrouper des variables homogènes entre elles.

La classification de variables est un enjeu central dans l'analyse multidimensionnelle : elle permet de résumer l'information, de construire des facteurs latents, de réduire la dimension ou d'identifier des familles de variables redondantes.

Le projet demandé implique :

- la gestion conjointe de variables numériques et catégorielles ;
- l'implémentation d'algorithmes incluant une phase de **réallocation** (en particulier pour k-medoids) ;
- la production de diagnostics, visualisations et outils d'interprétation ;
- la mise à disposition d'une application Shiny complète.

Notre package vise à fournir un environnement unique où l'utilisateur peut choisir une méthode, exécuter l'analyse, interpréter les groupes de variables et exporter les résultats.

## Chapitre 2

# Cahier des charges et validation

Le sujet fournit une liste précise d'attentes auxquelles notre package répond point par point.

### Méthodes de clustering exigées

- au moins deux méthodes différentes ;
- dont une méthode avec **réallocation explicite**.

**Validation.** Nous proposons quatre méthodes :

- **k-means** (numérique) ;
- **k-modes** (catégoriel) ;
- **k-prototypes** (mixte) ;
- **k-medoids** (méthode PAM, avec réallocation explicite).

La contrainte de réallocation est satisfaite via k-medoids.

### Représentation des clusters

- composantes latentes pour les numériques ;
- profils modaux pour les catégorielles ;
- combinaison pondérée pour les mixtes.

**Validation.** Chaque classe fille fournit :

- une composante latente issue d'une ACP locale (k-means, partie numérique de k-prototypes) ;
- un profil modal pour les catégorielles (k-modes, partie catégorielle de k-prototypes) ;
- un médoïde représentant le cluster (k-medoids).

### Interface utilisateur

Le sujet demande une application Shiny complète.

**Validation.** L'application fournie :

- charge jeux internes et fichiers CSV/XLSX ;
- permet la sélection d'actives et de supplémentaires ;
- offre la visualisation, les diagnostics, l'export, les dendrogrammes et les factor maps.

## Production de sorties

- résumés numériques ;
- représentations graphiques ;
- indicateurs d'adhésion ;
- export des clusters et d'un rapport synthétique.

**Validation.** Tous ces éléments sont implémentés dans les classes R6 et l'application Shiny.

## Chapitre 3

# Architecture du package

Le package suit une architecture orientée objet basée sur R6 avec :

- une **classe mère** `mmrClustVarBase` gérant :
  - la préparation des données ;
  - le pipeline `fit()` → `summary()` → `plot()` → `predict()` ;
  - les méthodes d'accès (centres, clusters, inertie, convergence) ;
  - les distances numériques et catégorielles ;
  - l'affichage générique et les tracés standards.
- quatre **classes filles** spécialisées : `KMeans`, `KModes`, `KPrototypes`, `KMedoids`.
- une **façade Interface** qui choisit la bonne méthode en fonction des variables.

L'utilisateur ne manipule que la façade, ce qui garantit une interface homogène.

## Chapitre 4

# Méthodes de clustering implémentées

Dans cette section, on note :

- $n$  : nombre d'individus (lignes du tableau de données) ;
- $p$  : nombre de variables actives (colonnes) ;
- $x_j = (x_{1j}, \dots, x_{nj})^\top$  : la  $j$ -ème variable ;
- $K$  : nombre de groupes de variables recherchés ;
- $C_k$  : l'ensemble des indices de variables affectées au cluster  $k$ .

L'objectif général est de construire une partition  $(C_1, \dots, C_K)$  de  $\{1, \dots, p\}$  telle que les variables d'un même cluster soient « proches » d'un prototype de cluster, selon une distance adaptée au type de variable.

### 4.1 k-means (numérique)

Cette méthode s'applique lorsque toutes les variables actives sont numériques. Pour chaque cluster  $k$ , on construit une **composante latente**  $Z_k$  par ACP locale sur les variables du cluster :

$$Z_k = 1^{\text{ère}} \text{ composante principale de } \{x_j : j \in C_k\}.$$

Pour chaque variable  $x_j$  et chaque cluster  $k$ , on définit :

$$r(x_j, Z_k) = \text{cor}(x_j, Z_k), \quad a_{jk} = r(x_j, Z_k)^2, \quad d_{jk} = 1 - a_{jk}.$$

- $a_{jk}$  est l'**adhésion** de la variable  $j$  au cluster  $k$  (proportion de variance expliquée) ;
- $d_{jk}$  est la **distance intra-cluster** (plus elle est faible, plus  $x_j$  est bien représentée par  $Z_k$ ).

L'algorithme cherche à **maximiser** la somme des adhésions

$$\sum_{j=1}^p a_{j, \text{cluster}(j)}$$

ou, de manière équivalente, à **minimiser** la somme des distances

$$W = \sum_{j=1}^p d_{j, \text{cluster}(j)} = \sum_{j=1}^p (1 - r(x_j, Z_{\text{cluster}(j)})^2).$$

L'algorithme implémenté est un schéma de type k-means sur les variables :

```

Choisir K
D finir K variables comme noyau initial des groupes
Calculer la composante latente Z_k de chaque groupe (ACP locale)
TANT QUE non convergence
    POUR chaque variable x_j
        Pour chaque groupe k, calculer r^2(x_j, Z_k)
        Affecter x_j au groupe k* pour lequel r^2(x_j, Z_k) est
            maximal
    FIN POUR
    Recalculer chaque composante latente Z_k par ACP locale
FIN TANT QUE

```

## 4.2 k-modes (catégoriel)

Cette méthode s'applique lorsque toutes les variables actives sont catégorielles. Pour chaque cluster  $k$  et pour chaque individu  $i$ , on définit un **profil modal**  $m_k(i)$  comme la modalité la plus fréquente parmi les variables du cluster :

$$m_k(i) = \arg \max_c \#\{j \in C_k : x_{ij} = c\}.$$

Pour une variable catégorielle  $x_j$  et un profil modal  $m_k$ , on utilise la **dissimilarité simple matching** :

$$d_{jk} = d(x_j, m_k) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(x_{ij} \neq m_k(i)),$$

et l'**adhésion** est définie par :

$$a_{jk} = 1 - d_{jk} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(x_{ij} = m_k(i)).$$

L'objectif est de minimiser la somme des distances intra-cluster :

$$W = \sum_{j=1}^p d_{j, \text{cluster}(j)}.$$

```

Choisir K
D finir K variables comme noyau des groupes
Calculer le profil modal m_k(i) de chaque groupe

```

```

(modalit la plus fr quente parmi les variables du groupe pour
chaque individu i)

TANT QUE non convergence
POUR chaque variable x_j
    Pour chaque groupe k, calculer la dissimilarit d(x_j, m_k)
        (d = proportion de d saccords : simple matching)
    Affecter x_j au groupe k* pour lequel d(x_j, m_k) est minimale
FIN POUR

Recalculer le profil modal m_k(i) de chaque groupe
FIN TANT QUE

```

### 4.3 k-prototypes (mixte)

La méthode k-prototypes permet de traiter conjointement des variables numériques et catégorielles. Pour chaque cluster  $k$ , on combine :

- une composante latente numérique  $Z_k$  (ACP locale sur les variables numériques du cluster) ;
- un profil modal catégoriel  $m_k(i)$  (modalité la plus fréquente pour chaque individu  $i$ ).

On introduit un paramètre de pondération  $\lambda > 0$  pour ajuster l'importance du volet catégoriel.

**Distance pour une variable numérique  $x_j$ .**

$$d_{jk}^{(\text{num})} = 1 - r(x_j, Z_k)^2.$$

**Distance pour une variable catégorielle  $x_j$ .**

$$d_{jk}^{(\text{cat})} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(x_{ij} \neq m_k(i)).$$

Dans notre cas, une variable est soit numérique, soit catégorielle. Pour unifier l'écriture, on peut écrire :

$$d_{jk} = \begin{cases} d_{jk}^{(\text{num})}, & \text{si } x_j \text{ est numérique,} \\ \lambda d_{jk}^{(\text{cat})}, & \text{si } x_j \text{ est catégorielle.} \end{cases}$$

L'adhésion est définie par  $a_{jk} = 1 - d_{jk}^{(\text{num})}$  pour les numériques et  $a_{jk} = 1 - d_{jk}^{(\text{cat})}$  pour les catégorielles (avant pondération).

```

Choisir K et le param tre de pond ration
D finir K noyaux de groupes (variables initiales)
Calculer pour chaque groupe :
    - la composante latente num rique Z_k (ACP locale sur les

```

```

variables num riques)
- le profil modal cat goriel m_k(i) (modalité la plus fr quente
  par individu)
TANT QUE non convergence
  POUR chaque variable x_j
    Si x_j est num rique :
      Pour chaque groupe k, calculer d_num(x_j, Z_k) = 1 - r^2(
        x_j, Z_k)
    Si x_j est cat gorielle :
      Pour chaque groupe k, calculer d_cat(x_j, m_k)
        (d_cat = proportion de d saccards : simple matching)
      Pour chaque groupe k :
        Calculer la distance totale
        d_total(x_j, C_k) = d_num(x_j, Z_k) +      * d_cat(x_j,
          m_k)
        (en ignorant le terme non pertinent si x_j n'est que
          num rique
          ou uniquement cat gorielle)
      Affecter x_j au groupe k* pour lequel d_total(x_j, C_k) est
        minimale
    FIN POUR
    Recalculer Z_k (partie num rique) et m_k(i) (partie cat gorielle
      ) pour chaque groupe
  FIN TANT QUE

```

## 4.4 k-medoids (dissimilarités mixtes)

La méthode k-medoids se base sur une **matrice de dissimilarités** entre variables, ce qui permet de mélanger naturellement variables numériques et catégorielles.

On définit une dissimilarité  $D(j, \ell)$  entre les variables  $x_j$  et  $x_\ell$  :

$$D(j, \ell) = \begin{cases} 1 - r(x_j, x_\ell)^2, & \text{si } x_j \text{ et } x_\ell \text{ sont numériques,} \\ \frac{1}{n} \sum_{i=1}^n \mathbb{1}(x_{ij} \neq x_{i\ell}), & \text{si elles sont toutes deux catégorielles,} \\ 1, & \text{si les types sont différents (numérique vs catégoriel).} \end{cases}$$

Pour un cluster  $C_k$ , un **médoïde**  $m_k$  est une variable du cluster minimisant le coût total :

$$m_k = \arg \min_{j \in C_k} \sum_{\ell \in C_k} D(j, \ell).$$

La fonction objectif globale est la somme des distances de chaque variable à son médoïde :

$$W = \sum_{k=1}^K \sum_{j \in C_k} D(j, m_k).$$

```

Choisir K
Construire la matrice de dissimilarites D entre toutes les p
variables
(par exemple : 1 - r^2 pour les numeriques, simple matching pour
les qualitatives)
Choisir K variables comme m do des initiaux
TANT QUE non convergence
    POUR chaque variable x_j
        Pour chaque m do de m_k, calculer d(x_j, m_k)      partir de D
        Affecter x_j au groupe k* du m do de le plus proche
    FIN POUR
    POUR chaque groupe C_k
        Pour chaque variable candidate x_j dans C_k
            Calculer le co t total :
            co t(x_j) = somme des distances d(x_j, x_l) pour
                toutes les x_l dans C_k
        Choisir comme nouveau m do de la variable x_j avec le co t
            minimal
    FIN POUR
FIN TANT QUE

```

## Chapitre 5

# Indicateurs et diagnostics

Nous produisons :

- inertie intra-classe (numérique, catégorielle, totale) ;
- adhésion :
  - $r^2$  pour les variables numériques ;
  - proportion de correspondance pour les catégorielles ;
- distances, profils, statistiques par groupes ;
- inertie expliquée : rapport interne / totale.

Ces indicateurs permettent :

- d'évaluer la compacité des clusters ;
- d'identifier les variables les plus représentatives ;
- de détecter des clusters hétérogènes ou peu stables.

# Chapitre 6

## Visualisations

L'application Shiny offre :

- courbe d'inertie (choix de  $K$ ) ;
- distribution des clusters ;
- heatmaps des profils de clusters ;
- factor maps (k-means) ;
- dendrogrammes Gower (méthodes mixtes) ;
- visualisation des adhésions.

Ces graphiques complètent les diagnostics numériques pour interpréter les groupes.

## Chapitre 7

# Application Shiny

L’interface Shiny permet :

- l’import de jeux intégrés ou de fichiers externes ;
- la sélection d’actives / supplémentaires ;
- l’exécution d’un clustering (avec choix de  $K$ ,  $\lambda$ , normalisation) ;
- la visualisation des résultats (résumés, plots, diagnostics) ;
- l’export d’un rapport texte et d’un bundle complet.

La structure en `ui.R` et `server.R` exploite la façade du package, garantissant une intégration propre et modulaire.

**Note sur le dataset `metal_universe`.** Nous incluons un jeu fictif permettant de tester l’application avec des données mixtes. Les quatre premières colonnes (groupe, sous-genre, pays, genre du chanteur) sont plausibles, le reste (scores numériques, dérivées, variables bruitées) a été généré de manière artificielle dans des gammes cohérentes.

## Chapitre 8

# Limites et perspectives

Les points suivants constituerait des améliorations pertinentes :

- optimisation automatique du paramètre  $\lambda$  dans k-prototypes ;
- ajout de métriques externes (silhouette gower, stability bootstrap) ;
- intégration d'une méthode hiérarchique complète (HCPC-variables) ;

## Chapitre 9

# Conclusion

Le package `mmrClustVar` propose une implémentation complète et modulaire du clustering de variables en R6. Il couvre l'ensemble du cahier des charges : quatre algorithmes complémentaires, visualisations, diagnostics, export, application Shiny.

Ce projet a permis d'aborder conjointement :

- la programmation orientée objet en R ;
- l'implémentation d'algorithmes de regroupement ;
- la construction d'outils d'analyse reproductibles ;
- la mise en place d'une interface interactive.

L'ensemble constitue une base solide pour des extensions futures et fournit un outil pédagogique complet pour l'analyse de variables.

# Bibliographie

- [1] 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.  
<https://arxiv.org/pdf/1112.0295>
- [2] Husson, F., Josse, J., & Pagès, J. (2017). *Exploratory Multivariate Analysis by Example using R* (2<sup>e</sup> éd.).  
Chapman and Hall/CRC.  
<http://staff.ustc.edu.cn/~ynyang/vector/books/Husson-Le-Pages.pdf>
- [3] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297).  
<https://www.cs.cmu.edu/~bhiksha/courses/mlsp.fall2010/class14/macqueen.pdf>
- [4] Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2, 283–304.  
<https://cse.hkust.edu.hk/~qyang/Teaching/537/Papers/huang98extensions.pdf>
- [5] Kaufman, L., & Rousseeuw, P. J. (2005). *Finding Groups in Data : An Introduction to Cluster Analysis*.  
Wiley.  
[https://www.researchgate.net/publication/220695963\\_Finding\\_Groups\\_in\\_Data](https://www.researchgate.net/publication/220695963_Finding_Groups_in_Data)
- [6] Rakotomalala, R. (2025). R programming & classification lectures. Université Lyon 2.  
<https://tutoriels-data-science.blogspot.com/>
- [7] R Core Team. (2025). *R : A Language and Environment for Statistical Computing*.  
R Foundation for Statistical Computing.  
<https://www.r-project.org/>
- [8] Chang, W. (2025). *R6 : Encapsulated Object-Oriented Programming for R*.  
R package documentation.  
<https://r6.r-lib.org>