Clustering K-prototypes

Équipe 19 – ANACLET Kylian, BIN DAIVIN Muhamad Zaiinizee, CHENNOUF Younes, KEBAIRI Ilyes

2025-12-10

Le jeu de données contient à la fois des variables numériques et catégorielles (région, condition d'éclairage, condition atmosphérique, etc.). L'algorithme K-means classique n'est donc pas adapté, car il repose sur la distance euclidienne, qui n'a pas de sens pour les variables catégorielles, même après encodage one-hot.

Il est préférable d'utiliser **K-prototypes**, qui combine la distance euclidienne pour les variables numériques et une mesure de dissimilarité spécifique pour les variables catégorielles. Cet algorithme permet ainsi d'obtenir des clusters plus fiables et cohérents.

K-prototypes

L'algorithme K-prototypes implémente la méthode de partitionnement par minimisation de la distance entre les observations et les centres de clusters. Plus précisément, il cherche à minimiser la fonction suivante :

$$\min_{c_1,\ldots,c_k} \sum_{k=1}^{K} \sum_{i,G(i)=k} d(x^{(i)}, c_k)$$

où $d(x^{(i)}, c_k)$ représente la distance entre l'observation $x^{(i)}$ et le centre du cluster c_k. Cette distance combine la distance euclidienne pour les variables numériques et une mesure de dissimilarité pour les variables catégorielles.

La minimisation est réalisée automatiquement par la fonction `kproto()` à travers un processus itératif. L'algorithme affecte chaque observation au cluster dont le centre est le plus proche, puis recalcule les centres, et répète ces étapes jusqu'à convergence. Le résultat de cette minimisation est observable via `tot.withinss`, qui représente la somme totale des distances intra-clusters.

A. Chargement et préparation des données

Le jeu de données final étant **très volumineux**, un échantillon aléatoire a été extrait pour faciliter le traitement et le clustering.

```
if(!require(dplyr)) install.packages("dplyr")
```

```
## Le chargement a nécessité le package : dplyr
```

```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag

## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(dplyr)

set.seed(321)
n_sample <- 5000

# Exécuter le script « nettoyage_donnees_clustering.Rmd », situé dans le dossier clustering, afin de gé
df <- read.csv("../data/donnees_clustering.csv")
df_complete <- df %>%
  mutate(across(where(is.character), as.factor)) %>%
  select(agg, lum, atm, col, vma)

# Convertir toutes les variables de type character en factor
# Puis sélectionner uniquement les 5 variables retenues pour le clustering
df <- df %>%
  mutate(across(where(is.character), as.factor)) %>%
  select(agg, lum, atm, col, vma)

# On tire par hasard 1000 accidents
df <- df[sample(nrow(df), n_sample), ]
```

On sélectionne ces 5 variables (agg, lum, atm, col, vma) car elles capturent les conditions principales de l'accident (localisation, luminosité, météo, type de collision et vitesse autorisée) et donnent les meilleurs résultats de clustering.

```r
str(df)
```

```
## 'data.frame':    5000 obs. of  5 variables:
##  $ agg: Factor w/ 2 levels "en agglo","hors agglo": 1 1 1 2 1 1 2 1 1 2 ...
##  $ lum: Factor w/ 5 levels "crepuscule ou aube",..: 2 5 5 5 2 5 1 5 5 1 ...
##  $ atm: Factor w/ 9 levels "autre","brouillard ou fumee",..: 8 3 6 6 6 6 6 6 3 6 ...
##  $ col: Factor w/ 7 levels "3+ veh - en chaine",..: 4 4 3 6 7 5 5 6 7 5 ...
##  $ vma: int  50 50 50 80 50 50 80 50 50 -1 ...
```

B. Verification de la légitimité de clustering

Avant de commencer on doit tout d'abord s'assurer que l'utilisation de la méthode de clustering est légitime ici (s'assurer que les données ne sont pas réparties de façon homogène). Pour ce faire on utilise la méthode statistique en calculant la statistique de Hopkins calculé avec la formule :

$$H = \frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

où :

- x_i la distance d'un i-éme point p_i choisi aléatoirement dans notre distribution à tster avec son plus proche voisin dans la même distribution
- y_i **:** la distance d'un i-éme point qi choisi aléatoirement dans une distribution aléatoire uniforme avec son plus proche voisin dans la distribution

Nos hypothèses étant :

- H_0 **:** Les données sont uniformément distribués
- H_1 **:** Les données ne sont pas uniformément distribués

```
if(!require(hopkins)) install.packages("hopkins")
```

```
## Le chargement a nécessité le package : hopkins
```

```
library(hopkins)

numeric_cols <- sapply(df, is.numeric)
n_numeric <- sum(numeric_cols)

if (n_numeric >= 2) {
  h_stat <- hopkins(df[, numeric_cols])
  cat("Statistique de Hopkins :", round(h_stat, 4), "\n")
} else {
  cat("Hopkins non calculable : seulement", n_numeric, "variable numérique (vma).\n")
  cat("   (La fonction nécessite au moins 2 variables numériques)\n")
}
```

```
## Hopkins non calculable : seulement 1 variable numérique (vma).
##    (La fonction nécessite au moins 2 variables numériques)
```

C. Normalisation des variables numériques

Les variables numériques sont standardisées pour garantir que toutes contribuent équitablement au calcul des distances, indépendamment de leur échelle de mesure. Cela évite qu'une variable avec de grandes valeurs domine artificiellement la formation des clusters.

```
numeric_cols <- sapply(df, is.numeric)
df_scaled <- df
df_scaled[numeric_cols] <- scale(df[numeric_cols])
```

Remarque : les variables catégorielles (facteurs) ne sont pas standardisées.

D. Choix du nombre de clusters (k)

On applique l'algorithme k-prototypes pour différentes valeurs de k.

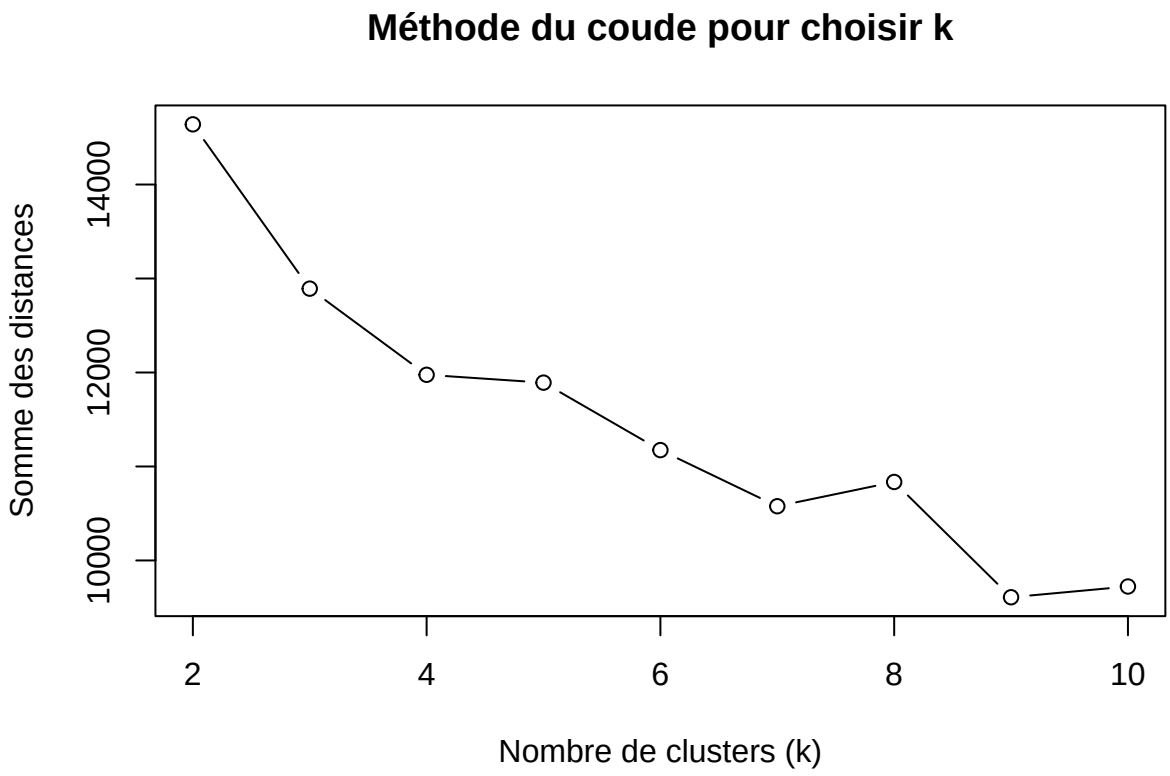
```
set.seed(321)
if(!require(clustMixType)) install.packages("clustMixType")
```

```
## Le chargement a nécessité le package : clustMixType
```

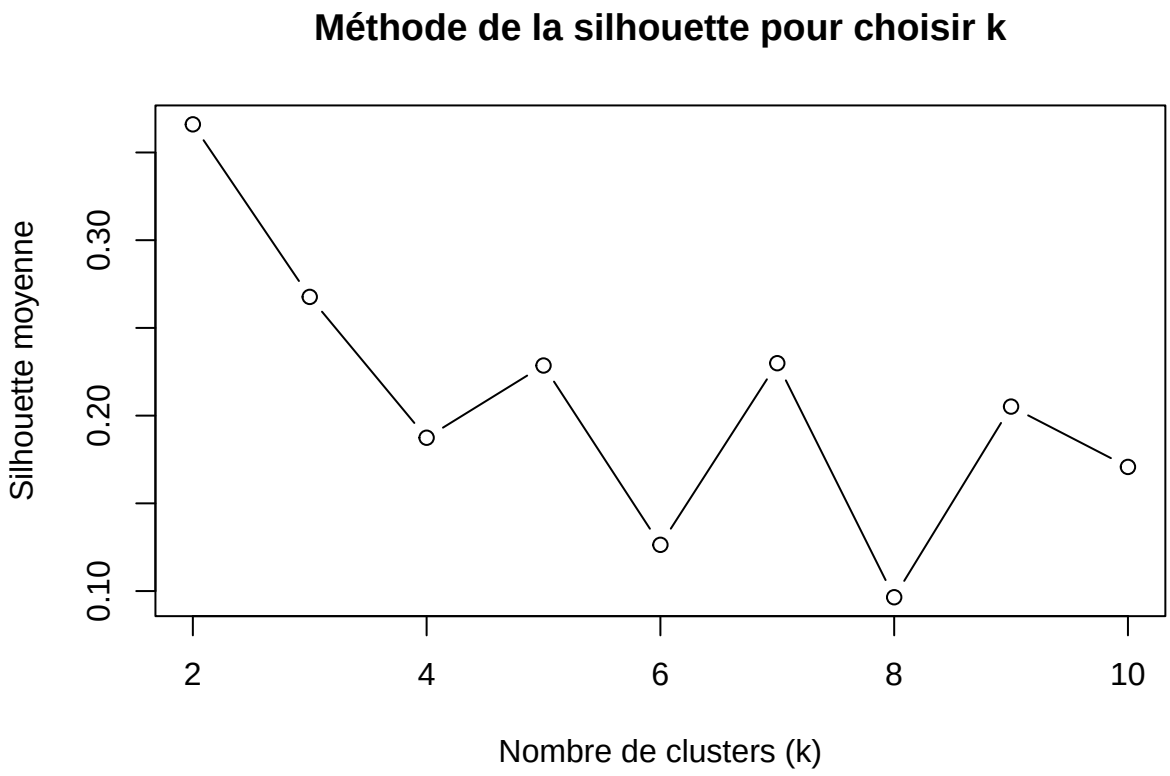
```r
if(!require(cluster)) install.packages("cluster")
```

```
## Le chargement a nécessité le package : cluster
```

```r
library(clustMixType)
library(cluster)

k_max <- 10
cost <- numeric(k_max)
sil_avg <- numeric(k_max)

# Calculer la distance de Gower sur les données non normalisées
d <- daisy(df, metric = "gower")

# Exécuter k-prototypes pour différentes valeurs de k
for (k in 2:k_max) {
  kp <- kproto(df_scaled, k, verbose = FALSE)
  cost[k] <- kp$tot.withinss

  sil <- silhouette(kp$cluster, d)
  sil_avg[k] <- mean(sil[, 3])
}

# Plot méthode du coude
plot(2:k_max, cost[2:k_max],
     main = "Méthode du coude pour choisir k",
     type = "b", xlab = "Nombre de clusters (k)",
     ylab = "Somme des distances")
```




```r
# Plot méthode de la silhouette
plot(2:k_max, sil_avg[2:k_max],
     main = "Méthode de la silhouette pour choisir k",
     type = "b", xlab = "Nombre de clusters (k)",
     ylab = "Silhouette moyenne")
```



Observations

La méthode du coude montre une stabilisation à partir de k = 4, tandis que l'indice de Silhouette est maximal à k = 2.

Pour choisir le meilleur k, nous testons k = 2, 3 et 4 en comparant leurs indices de qualité.

E. Application de k-prototypes et analyse des résultats

```
set.seed(321)
res_k2 <- kproto(df_scaled, k = 2, verbose = FALSE)
res_k3 <- kproto(df_scaled, k = 3, verbose = FALSE)
res_k4 <- kproto(df_scaled, k = 4, verbose = FALSE)
```

F. Test de la qualité du clustering

Afin d'évaluer la qualité du clustering on évalue les deux indices :

- l'indice de Silhouette pour chaque cluster qui pour chaque élément évalue à quel point un élément est "similaire" par rapport aux autres éléments du même cluster .
- l'indice de Dunn qui est défini comme suit :

$$D = \frac{min.separation}{max.diameter}$$

De plus on veille à adapter cette methode d'évaluation à la methode de clustering utilisée (k-prototype) et ce en utilisant justement la distance de gower qui prend en compte les variables catégorielles

```
if(!require(fpc)) install.packages("fpc")
```

```
## Le chargement a nécessité le package : fpc
```

```
library(cluster)
library(fpc)

dist_gower <- daisy(df, metric = "gower")

evaluer_k <- function(res, k_val) {
  cat("=== k =", k_val, "===\n")
  sil <- silhouette(res$cluster, dist_gower)
  cat("Silhouette moyenne :", round(mean(sil[,3]), 4), "\n")
  cat("Détail :", paste(round(summary(sil)$clus.avg.widths, 4), collapse = ", "), "\n")
  dunn <- cluster.stats(dist_gower, res$cluster)$dunn
  cat("Dunn :", round(dunn, 6), "\n\n")
}

evaluer_k(res_k2, 2)
```

```
## === k = 2 ===
## Silhouette moyenne : 0.366
## Détail : 0.3327, 0.3853
## Dunn : 0.001128
```

```
evaluer_k(res_k3, 3)
```

```
## === k = 3 ===
## Silhouette moyenne : 0.2677
## Détail : 0.2861, 0.3107, 0.2277
## Dunn : 0.001133
```

```
evaluer_k(res_k4, 4)
```

```
## === k = 4 ===
## Silhouette moyenne : 0.1874
## Détail : 0.3534, 0.1159, 0.1277, 0.0579
## Dunn : 0.00226
```

Bien que $k = 2$ offre la meilleure qualité statistique avec des clusters très cohérents, une partition en seulement deux groupes est trop simpliste pour refléter la diversité des accidents routiers. Nous retenons donc $k = 3$, qui propose trois profils plus nuancés et interprétables, malgré une qualité statistique légèrement moindre.

```r
table(res_k3$cluster)
```

Distribution des observations par cluster

```
##
##    1    2    3
## 1807 1135 2058
```

G. Comparaison échantillon vs données complètes

Moyennes des variables numériques par cluster

```r
library(clustMixType)

# Moyennes vma
cat("Moyenne vma sur l'échantillon :\n")
```

```
## Moyenne vma sur l'échantillon :
```

```r
print(round(tapply(df$vma, res_k3$cluster, mean), 1))
```

```
##    1    2    3
## 85.8 41.8 49.0
```

```r
res_full <- kproto(df_complete, k = 3, verbose = FALSE)
cat("\nMoyenne vma sur les données complètes :\n")
```

```
##
## Moyenne vma sur les données complètes :
```

```r
print(round(tapply(df_complete$vma, res_full$cluster, mean), 1))
```

```
##    1    2    3
## 46.2 88.8 76.8
```

Répartition des variables catégorielles par cluster

```r
cat_cols <- sapply(df, is.factor)

for (col in names(df)[cat_cols]) {
  cat("\n=== Variable :", col, "===\n\n")

  cat("Échantillon :\n")
  print(table(df[[col]], res_k3$cluster))
```

```r
  cat("\nDonnées complètes :\n")
  print(table(df_complete[[col]], res_full$cluster))

  cat("\n")
}
```

```
##
## === Variable : agg ===
##
## Échantillon :
##
##                 1    2    3
##   en agglo     16 1090 1995
##   hors agglo 1791   45   63
##
## Données complètes :
##
##                  1     2     3
##   en agglo   19168   206   239
##   hors agglo   756  7493  3254
##
##
## === Variable : lum ===
##
## Échantillon :
##
##                                    1    2    3
##   crepuscule ou aube             220  113  209
##   nuit avec eclairage allume      69  350  533
##   nuit avec eclairage non allume  18   13   14
##   nuit sans eclairage            486   53   53
##   plein jour                    1014  606 1249
##
## Données complètes :
##
##                                     1     2     3
##   crepuscule ou aube             1805   114  1388
##   nuit avec eclairage allume     5422   312   252
##   nuit avec eclairage non allume  228    81    43
##   nuit sans eclairage             669  1989  1157
##   plein jour                    11800  5203   653
##
##
## === Variable : atm ===
##
## Échantillon :
##
##                          1    2    3
##   autre                 28    3   10
##   brouillard ou fumee   33    4    9
##   couvert               75   52   83
##   eblouissant           28   35   33
##   neige ou grele        16    3    6
```

```
##   normale              1375  864 1649
##   pluie forte            49   27   26
##   pluie legere          196  142  239
##   vent fort ou tempete    7    5    3
##
## Données complètes :
##
##                           1     2     3
##   autre                  92    68    61
##   brouillard ou fumee    98   128   102
##   couvert               804   296   166
##   eblouissant           374   139    42
##   neige ou grele         50    56    27
##   normale             15788  5966  2514
##   pluie forte           344   187    91
##   pluie legere         2345   835   479
##   vent fort ou tempete   29    24    11
##
##
## === Variable : col ===
##
## Échantillon :
##
##                         1    2    3
##   3+ veh - en chaine  108    5   60
##   3+ veh - multiple    97   11   50
##   autre               465  958    0
##   deux veh - arriere  323   59  295
##   deux veh - cote     324    0 1252
##   deux veh - frontal  218   48  216
##   sans                272   54  185
##
## Données complètes :
##
##                         1    2    3
##   3+ veh - en chaine  429  607  142
##   3+ veh - multiple   380  382  155
##   autre              5994 2715  250
##   deux veh - arriere 2282 1522  396
##   deux veh - cote    7576 1450  444
##   deux veh - frontal 1911  812  565
##   sans               1352  211 1541
```

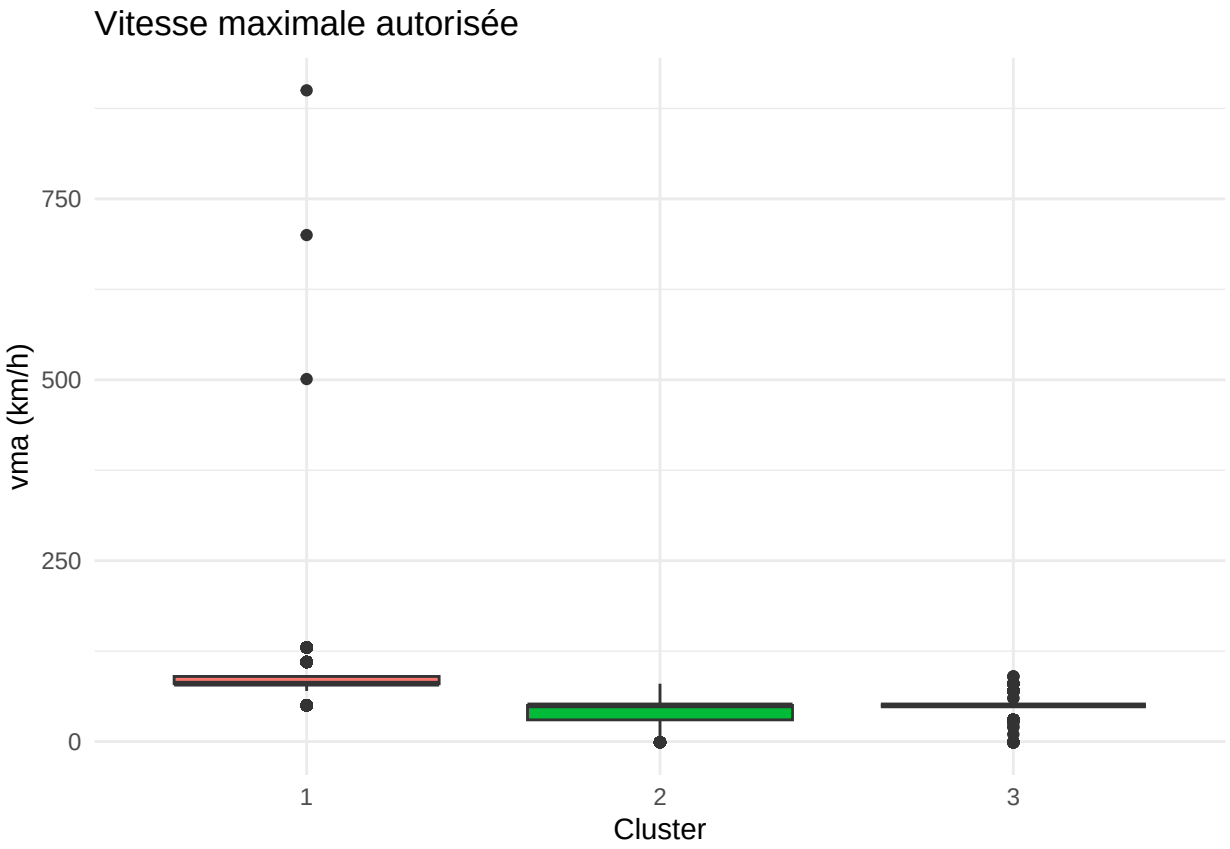
H. Visualisation des clusters k-prototypes

```
library(ggplot2)
library(dplyr)
library(gridExtra)
```

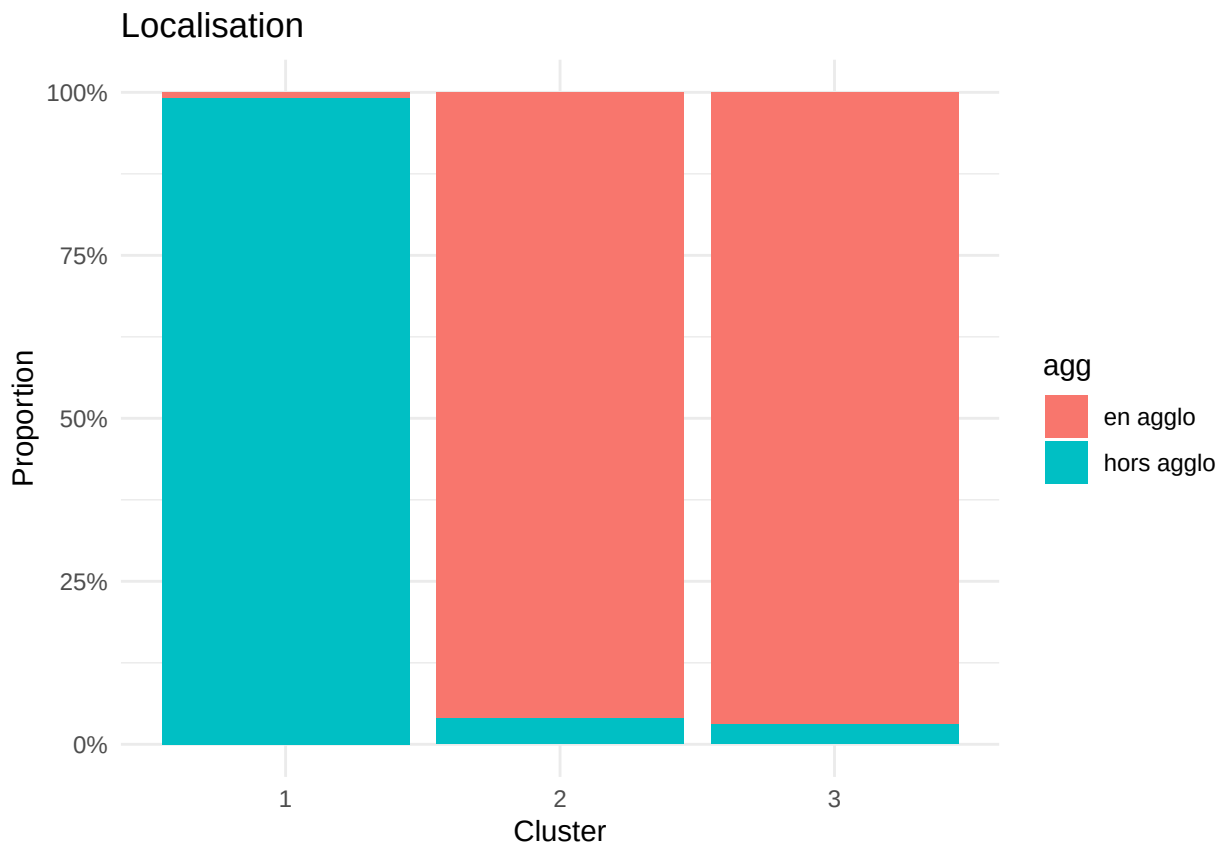
```
##
## Attachement du package : 'gridExtra'
```

```
## L'objet suivant est masqué depuis 'package:dplyr':
##
##     combine
```

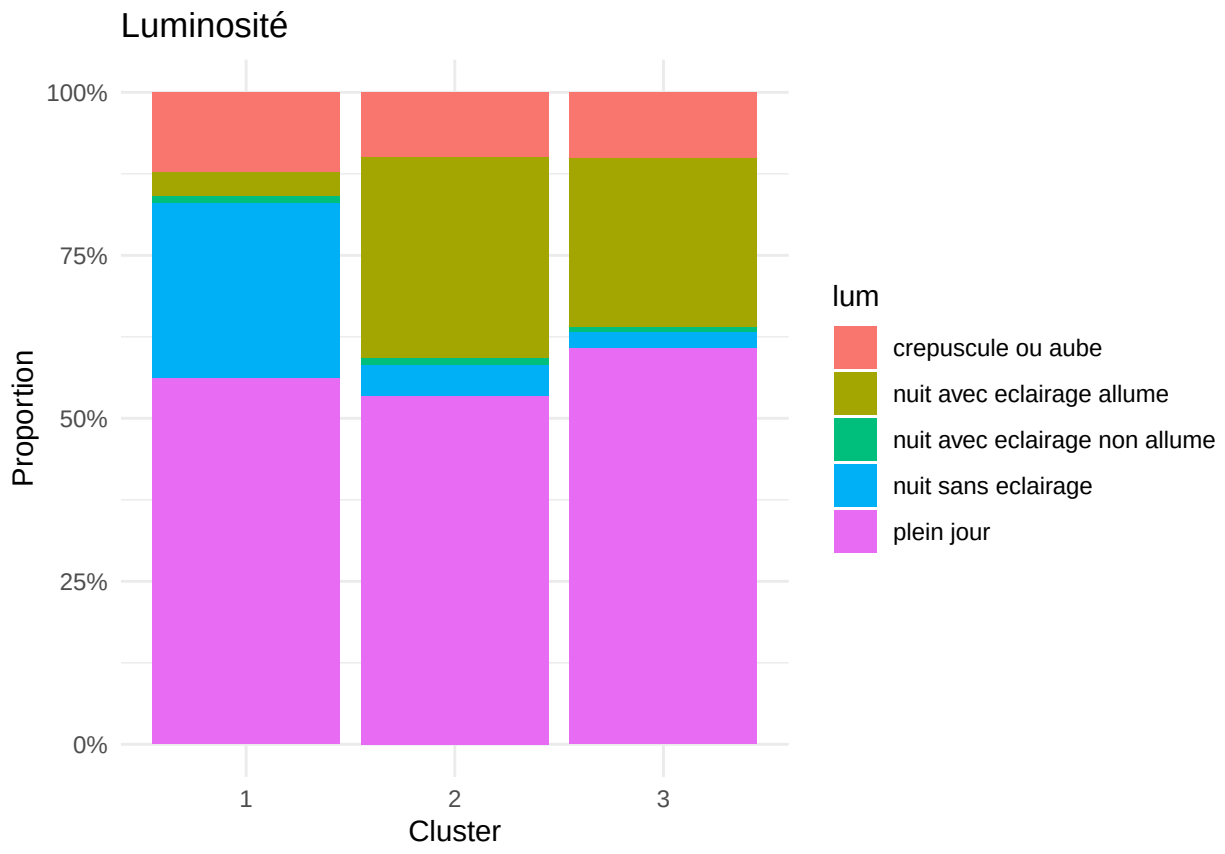
```r
library(scales)  # Pour percent()

df_vis <- df %>% mutate(cluster = as.factor(res_k3$cluster))

p1 <- ggplot(df_vis, aes(x = cluster, y = vma, fill = cluster)) +
  geom_boxplot() + theme_minimal() + theme(legend.position = "none") +
  labs(title = "Vitesse maximale autorisée", x = "Cluster", y = "vma (km/h)")

p2 <- ggplot(df_vis, aes(x = cluster, fill = agg)) + geom_bar(position = "fill") +
  scale_y_continuous(labels = percent) + theme_minimal() +
  labs(title = "Localisation", x = "Cluster", y = "Proportion")

p3 <- ggplot(df_vis, aes(x = cluster, fill = lum)) + geom_bar(position = "fill") +
  scale_y_continuous(labels = percent) + theme_minimal() +
  labs(title = "Luminosité", x = "Cluster", y = "Proportion")

p4 <- ggplot(df_vis, aes(x = cluster, fill = atm)) + geom_bar(position = "fill") +
  scale_y_continuous(labels = percent) + theme_minimal() +
  labs(title = "Conditions atmosphériques", x = "Cluster", y = "Proportion")

p5 <- ggplot(df_vis, aes(x = cluster, fill = col)) + geom_bar(position = "fill") +
  scale_y_continuous(labels = percent) + theme_minimal() +
  labs(title = "Type de collision", x = "Cluster", y = "Proportion")
p1
```


Vitesse maximale autorisée
vma (km/h)
750
500
250
0
1
2
3
Cluster

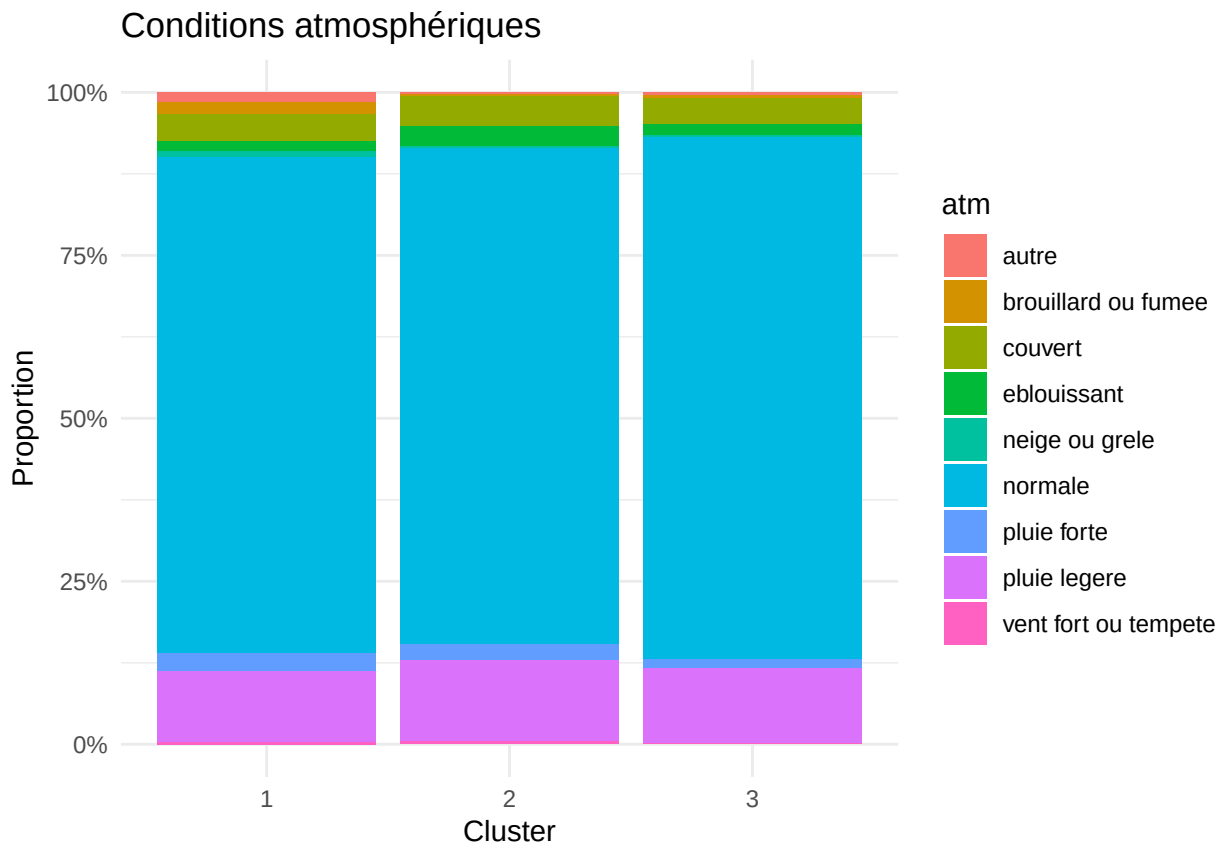
```
p2
```


Localisation
100%
75%
50%
25%
0%
Proportion
1
2
3
Cluster
agg
en agglo
hors agglo

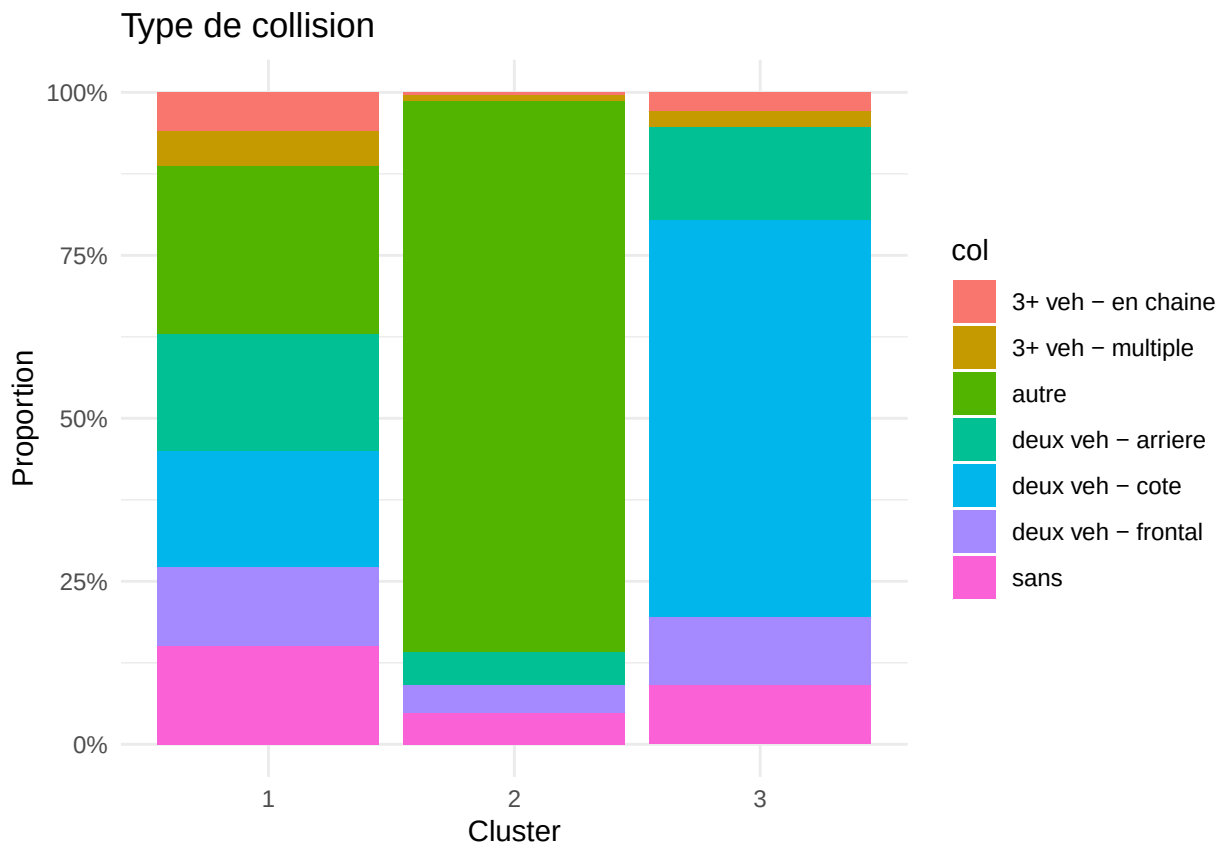
```
p3
```


Luminosité
100%
75%
50%
25%
0%
Proportion
1
2
3
Cluster
lum
crepuscule ou aube
nuit avec eclairage allume
nuit avec eclairage non allume
nuit sans eclairage
plein jour

```
p4
```


Conditions atmosphériques
100%
75%
50%
25%
0%
Proportion
1
2
3
Cluster
atm
autre
brouillard ou fumee
couvert
eblouissant
neige ou grele
normale
pluie forte
pluie legere
vent fort ou tempete

```
p5
```



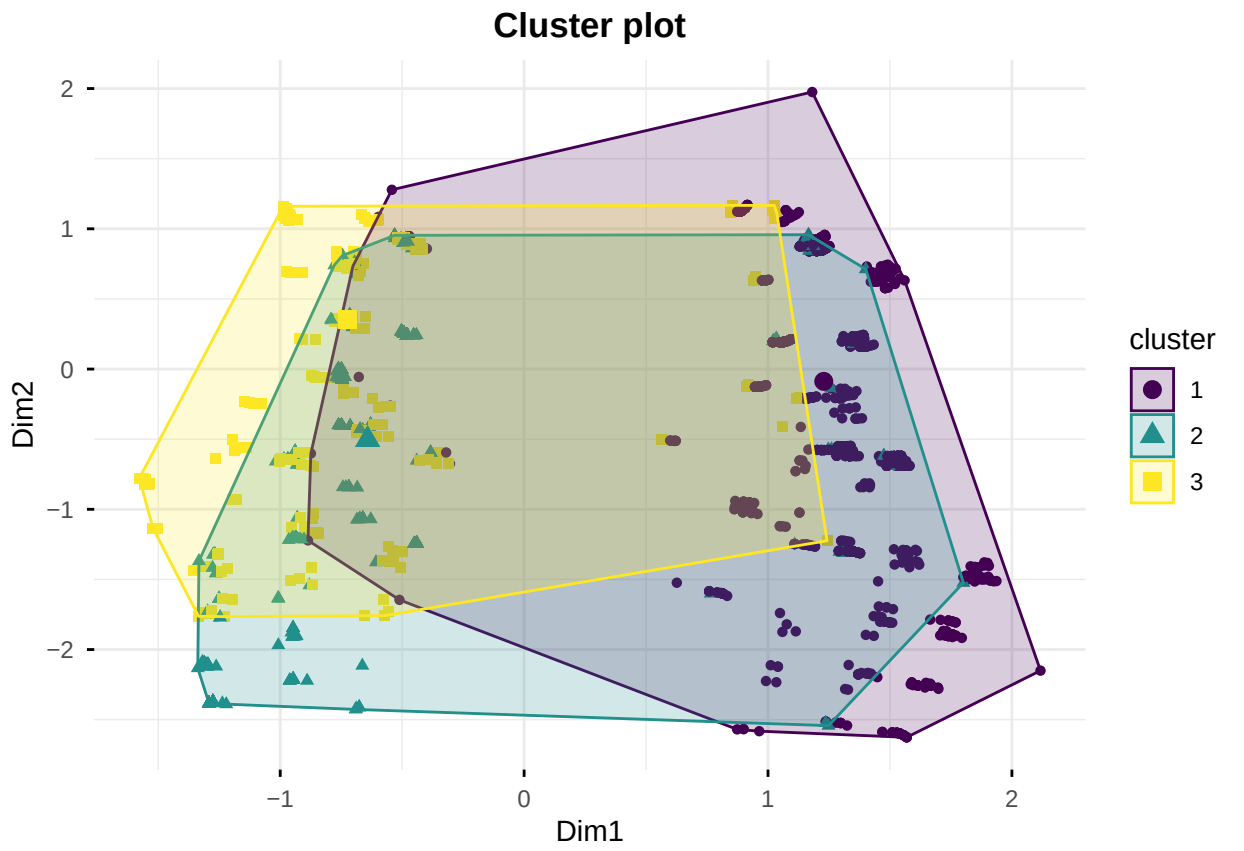

```
set.seed(321)
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(viridis)
```

```
## Le chargement a nécessité le package : viridisLite

##
## Attachement du package : 'viridis'

## L'objet suivant est masqué depuis 'package:scales':
##
##     viridis_pal
```

```
mds_df <- cmdscale(dist_gower, k = 2) %>% as.data.frame()
colnames(mds_df) <- c("Dim1", "Dim2")

fviz_cluster(list(data = mds_df, cluster = res_k3$cluster),
             geom = "point", palette = viridis(3), ellipse.type = "convex",
             show.clust.cent = TRUE, ggtheme = theme_minimal(),
             ) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```




```
ggsave("../Figures/clusters_fviz_final.png", width = 10, height = 8, dpi = 300)
```

Les groupes peuvent se chevaucher car la visualisation 2D (via PCA) projette un espace multidimensionnel sur un simple plan. Cette réduction dimensionnelle entraîne une perte d'information, et des clusters bien séparés dans l'espace original peuvent alors apparaître mélangés sur le graphique.

I. Inteprétation de chaque cluster

Cluster 1 : Accidents routiers hors agglomération

- **VMA moyenne** : 85.8 km/h (échantillon) / 48.3 km/h (données complètes)
- **Localisation** : Majoritairement hors agglomération (99% dans l'échantillon, 56% dans les données complètes)
- **Éclairage** : Principalement en plein jour (56%) et nuit sans éclairage (27%)
- **Conditions atmosphériques** : Normales (76%)
- **Type de collision** : Diversifié - collisions par le côté (18%), par l'arrière (18%), et accidents sans collision (15%)
- **Profil type** : Accidents sur routes hors agglomération, souvent à vitesse élevée, en conditions normales de jour ou la nuit sans éclairage public.

Cluster 2 : Accidents routiers à haute cinétique

- **VMA moyenne** : 41.8 km/h (échantillon) / 87.3 km/h (données complètes)
- **Localisation** : Très majoritairement hors agglomération (96% données complètes)
- **Éclairage** : Forte proportion de nuit sans éclairage (29% données complètes) et plein jour (56%)
- **Conditions atmosphériques** : Normales (76%)
- **Type de collision** : Surtout "autre" (26%), collisions en chaîne (7%) et par l'arrière (18%)
- **Profil type** : Accidents sur routes rapides hors agglomération (voies express, nationales), caractérisés par des vitesses élevées et souvent en conditions de faible visibilité nocturne.

Cluster 3 : Accidents urbains

- **VMA moyenne** : 49.0 km/h (échantillon) / 42.2 km/h (données complètes)
- **Localisation** : Très majoritairement en agglomération (97% échantillon, 93% données complètes)
- **Éclairage** : Principalement nuit avec éclairage allumé (66% données complètes) et plein jour (12%)
- **Conditions atmosphériques** : Normales (72%)
- **Type de collision** : Dominé par les collisions par le côté (61% échantillon) et "autre" (62% données complètes)
- **Profil type** : Accidents urbains à vitesse modérée, survenant souvent la nuit en zone éclairée, avec une prédominance de collisions latérales aux intersections.

J. Conclusion

L'algorithme K-prototypes a identifié trois groupes d'accidents distincts en se basant sur les variables contextuelles (localisation, conditions d'éclairage, conditions atmosphériques, type de collision, vitesse maximale autorisée). Cette sélection de variables a permis d'obtenir le meilleur indice de qualité de clustering, offrant ainsi une segmentation optimale pour analyser les différents profils d'accidents et identifier les facteurs de risque spécifiques à chaque groupe.