SY19 - TP01
Positionnement multidimensionnel

Alice Ngwembou - Antoine Hars

October 18, 2013

Introduction

Au cours de ce TP, nous étudions les différentes types de techniques de positionnement multidimensionnel, dans un premier temps théoriquement, puis dans un second temps sur différents types de données (*mutations* et *airport*) au moyen de fonctions de R.

Exercice 1 : Exercice théorique

Première partie : ACP

Question 1 :
Afin de calculer les axes factoriels de l'ACP du nuage de points définis, nous centrons, dans un premier temps, la matrice X donnée, ce qui nous permet de prendre comme nouvelle origine le centre de gravité dans l'espace des individus.
Nous calculons ensuite la matrice de covariance S pour appliquer la fonction *eigen()* sur cette dernière, ce qui nous donne les résultats suivants :

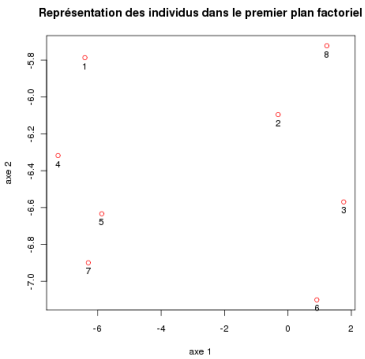
	$\lambda 1$	$\lambda 2$	$\lambda 3$	$\lambda 4$	$\lambda 5$	$\lambda 6$	$\lambda 7$	$\lambda 8$
Valeurs propres	24.26	0.006	3.85e-16	3.65e-17	1.52e-19	-1.84e-19	-7.17e-16	-1.50e-15
Inerties expliquées	98.45%	100%	100%	100%	100%	100%	100%	100%

Tableau des valeurs propres et inerties expliquées obtenues sur le nuage de points définis

D'après les valeurs obtenues, nous pouvons dire que seules les 2 premières valeurs propres sont intéressantes car les valeurs suivantes sont excessivement proches de zéro (positives ou négatives).
En calculant les inerties expliquées en nous basant sur la somme des valeurs absolues des valeurs propres, nous pouvons remarquer que les 2 premiers axes factoriels, donc le plan factoriel défini par ces 2 axes, cumulent environ 100% de l'information.

Question 2 :
La matrice de composantes principales nous permet de représenter les 8 individus dans le premier plan factoriel :

$$C = XMU = \begin{pmatrix} -3.62 & 0.60 \\ 2.47 & 0.29 \\ 4.54 & -0.18 \\ -4.47 & 0.07 \\ -3.09 & -0.24 \\ 3.69 & -0.71 \\ -3.52 & -0.51 \\ 4.01 & 0.67 \end{pmatrix}$$



L'ACP applique un changement de base au jeu de données car on a cherché à obtenir une représentation fidèle du nuage de points en le projetant sur un espace de plus faible dimension (dans notre cas, de dimension 2) sans perdre trop d'informations. Les variables obtenues par la projection sont donc des combinaisons linéaires des variables du nuage initial.

Question 3 :
$\Sigma_{\alpha=1}^{k} c_\alpha u'_\alpha$ pour $k = 1$ et $k = 2$ nous permet de retrouver notre matrice X de départ, à savoir la matrice contenant le jeu de données : $X = \Sigma_{\alpha=1}^{k} c_\alpha u'_\alpha$. En effet puisque $C = XMU$ alors $X = CU^t M^t$, or M est la matrice identité dans notre cas d'où $X = CU^t$.

Deuxième partie : MDS

Question 1 :
Le calcul de D^2 nous donne le tableau suivant (utilisation de la méthode R $dist(X, method = "euclidian")$) :

$$D^2 = \begin{vmatrix} 37.25 & & & & & & \\ 67.25 & 4.50 & & & & & \\ 1.00 & 48.25 & 81.25 & & & & \\ 1.00 & 31.25 & 58.25 & 2.00 & & & \\ 55.25 & 2.50 & 1.00 & 67.25 & 46.25 & & \\ 1.25 & 36.50 & 65.00 & 1.25 & 0.25 & 52.00 & \\ 58.25 & 2.50 & 1.00 & 72.25 & 51.25 & 2.00 & 58.00 \end{vmatrix}$$

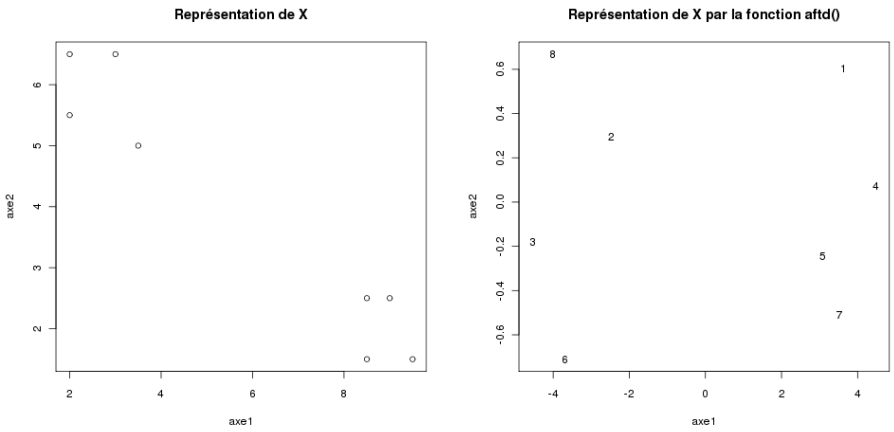
Question 2 :
Pour calculer la matrice des produits scalaires W, 2 méthodes sont possibles :
- En utilisant la matrice centrée des données de départ X : $W = X * X^t$
- En partant du tableau des distances euclidiennes D^2 : $W = -\frac{1}{2} * Q_n * D^2 * Q_n$ avec $n = 8$ individus

Question 3 :
Pour déterminer si W ou $\frac{1}{n}W$ est semi-définie positive, nous devons observer si les valeurs propres de la matrice $\frac{1}{n}W$ sont positives :
Le calcul des valeurs propres nous donne le même résultat que celui trouvé à partir de la matrice de covariance S dans la première partie. Ainsi nous pouvons observer que nous obtenons 5 valeurs propres positives (dont 3 valeurs comprises entre 10^{-16} et 10^{-19}) et 3 valeurs propres négatives (comprises entre 10^{-19} et 10^{-15}), nous les considérons comme nulles puisqu'elles qu'elles sont minimes (étant donné que les 2 premières valeurs propres regroupent environ 100% de l'information).
Les valeurs propres obtenues étant positives ou nulles, la matrice W (de même que $\frac{1}{n}W$) est semi-définie positive.

Question 4 :
La matrice diagonale des valeurs propres L est obtenue par la formule suivante : $L = \lambda * I_n$, avec λ correspondant au vecteur ligne des valeurs propres de $\frac{1}{n}W$.
La matrice des vecteurs propres normés V est obtenue via l'expression suivante : $V = \sqrt{n}\frac{1}{n}U$ avec U la matrice des vecteurs propres de W.

Question 5 :
Nous obtenons la représentation fournie par l'$AFTD$ avec l'expression des composantes principales suivantes : $C = V\sqrt{L}$.

Question 6 :
La représentation de X et celle fournit par l'$AFTD$ nous donne les graphiques suivants :



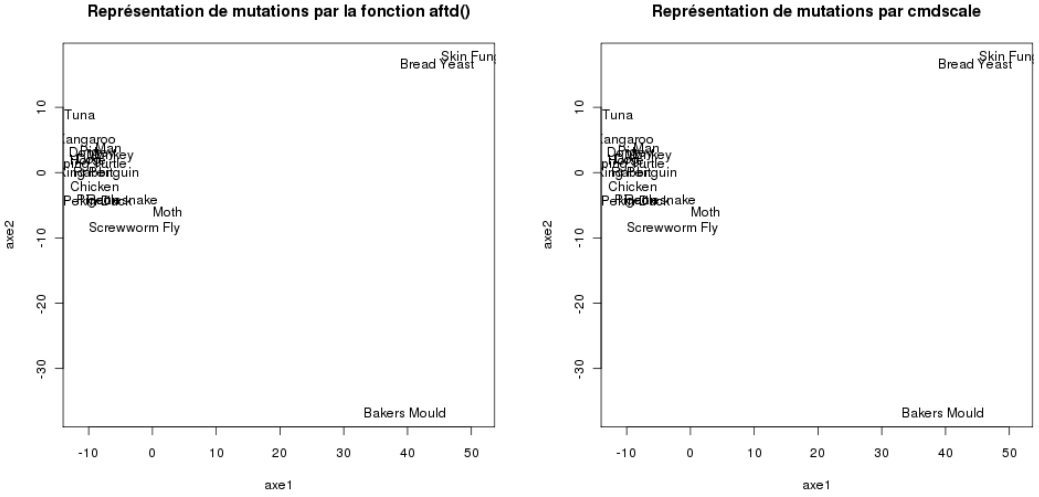
Nous remarquons sur les 2 graphiques que nous avons 2 classes d'individus bien identifiés. La représentation par l'$AFTD$ réduit les distances qu'il y a entre chaque point.

Question 7 :
La fonction *aftd()* créée est la suivante :

```
aftd <- function (d) {

  # Transformation de la matrice de distances en matrice
  dim_mat = as.matrix(d)
  d2 = dim_mat^2

  # Récupération de la première dimension de la matrice (nbre de lignes = nbre d'individus)
  dimension = diag(dim(dim_mat))[1]

  # Création de la matrice identité
  id = diag(dimension)

  # Création de la matrice unitaire
  un = matrix(rep(1, dimension^2), dimension)

  # Création de la matrice Q de centrage
  q = id - (1/dimension) * un

  # Calcul de la matrice w
  w = -(1/2) * q %*% d2 %*% q

  # Matrice associée aux vecteurs propres
  V = eigen(1 / dimension * w)$vectors[,1:dimension]
  V = sqrt(dimension) * V

  # Matrice associées aux valeurs propres
  L = diag(c(eigen(1 / dimension * w)$values[1:dimension]))

  # Calcul composante principale
  C = V %*% sqrt(L)

  # Calcul du pourcentage d'inertie pour les 7 premières valeurs propres
  quality = sum(diag(L)) / sum(eigen(1 / dimension * w) $values) * 100

  result <- new.env()
  result$quality <- quality
  result$C <- C
  return(result)
}
```

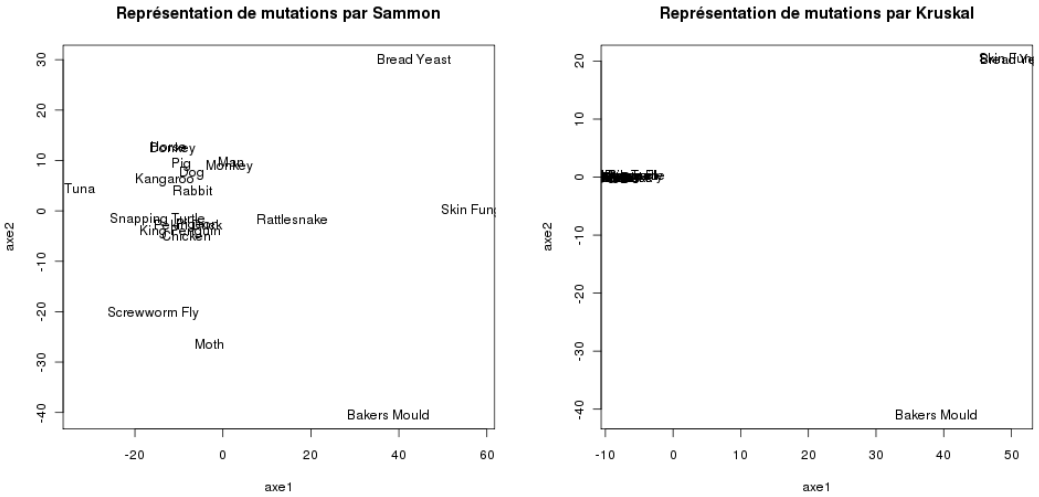
Exercice 2 : Les données de mutations

Question 1 :



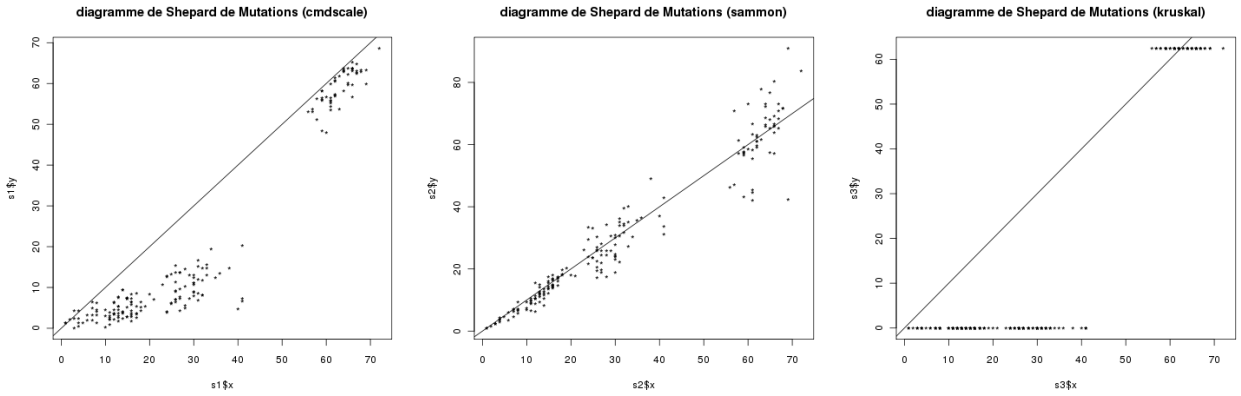
Les représentations de notre fonction *aftd()* et de la fonction *cmdscale()* sont similaires, on peut observer 3 groupes d'espèces bien différenciés entre eux.

Question 2 :



La représentation de *Kruskal* nous montre les 3 groupes présents sur les 2 représentations de l'*AFTD* précédentes, cependant, les individus de chaque groupe sont beaucoup plus concentrés sur un même point, les distances entre les individus sont minimes.
Concernant la méthode de *Sammon*, nous pouvons observer au contraire que les individus sont plus dispersés par rapport aux autres représentations. Nous n'avons pas 3 groupes d'individus bien définis. La projection par cette méthode ne tend pas à minimiser les distances entre chaque individus du nuage de points initial.
L'*AFTD* semble être une méthode pour donner une représentation intermédiaire entre les méthodes de *Kruskal* et de *Sammon*.

Question 3 :

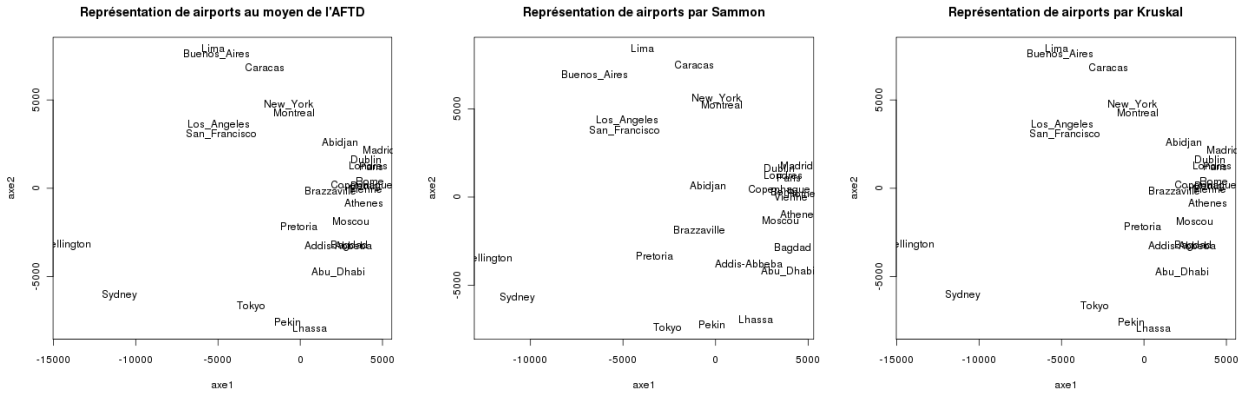


Ces diagrammes de *Shepard* nous permettent de visualiser la relation entre les dissimilarités initiales et les distances données par les différentes méthodes, donc d'apprécier la qualité des représentations de données par ces méthodes (*AFTD*, *Sammon* et *Kruskal*). Pour la méthode de l'*AFTD*, nous remarquons que la projection des individus est sous-estimée (les points sont un peu en-dessous de la diagonale). Pour la méthode de *Sammon*, la projection des individus a tendance à suivre la diagonale, ce qui nous indique que la méthode de *Sammon* nous donne une projection de meilleure qualité par rapport à la précédente. La qualité de représentation de la méthode de *Kruskal* est la plus faible des 3. Nous remarquons que *Kruskal* minimise au maximum la distance entre les individus, ce qui nous donne une vue globale plus claire, ce qui nous permet de dégager les groupes d'individus plus facilement alors que la méthode de *Sammon* nous permet d'étudier les distances entre chaque individus de manière plus fiable. L'*AFTD* nous permet une représentation intermédiaire.

Exercice 3 : Les données de distances entre aéroports

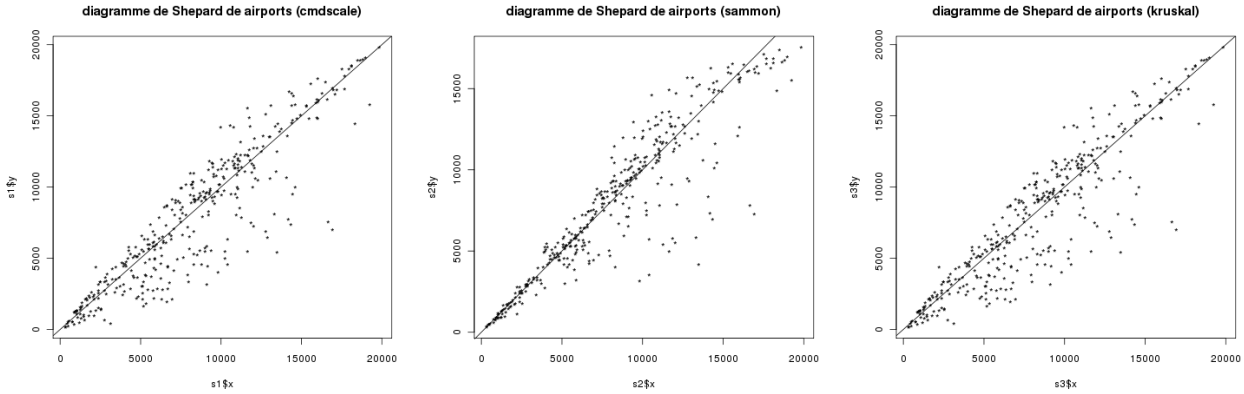
Dans cet exercice, nous travaillons sur les données contenues dans le fichier *airport2.txt* qui représentent les distances de vol entre des aéroports situés dans le monde. L'objectif de cet exercice est d'appliquer les différentes méthodes de positionnement multidimensionnel à ces données et de comparer les résultats obtenus.

L'exécution des méthodes de l'*AFTD*, de *Sammon* et de *Kruskal* sur le tableau de distance des données nous donne les graphiques suivants :



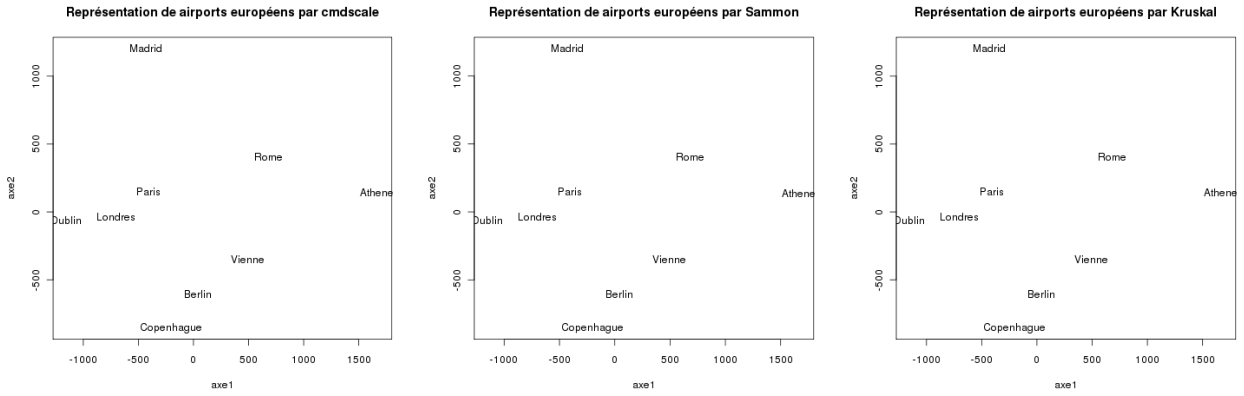
Question 1 :
À travers la représentation de l'*AFTD* (fonction *cmdscale*), nous pouvons observer que les différents continents ressortent plus ou moins entre eux. Nous avons l'Amérique en haut du graphique, l'Europe au centre et les pays orientaux en bas du graphique. Etant donné que les distances ont été calculées dans un référentiel en 3 dimensions, la représentation de ces distances dans un référentiel en 2 dimensions altère le positionnement des aéroports entre eux mais l'ensemble reste cohérent.

Question 2 :
La représentation de *Sammon* présente une répartition globale des éléments assez semblable à la représentation de l'*AFTD*. À l'intérieur de chaque groupe d'aéroports, nous notons cependant des différences sensibles (Tokyo, Pékin et Lhassa sont positionnées différemment). Quant à la représentation de *Kruskal*, nous remarquons une disposition des aéroports très similaire à la représentation de l'*AFTD*.
La différence de représentation entre celle de *Sammon* et celles de *Kruskal* et de l'*AFTD* provient vraisemblablement du changement de dimension appliqué pour obtenir une représentation sur 2 axes de distances obtenues dans 3 dimensions.
Pour comparer ces 3 représentations, nous observons les diagrammes de Shepard associés :

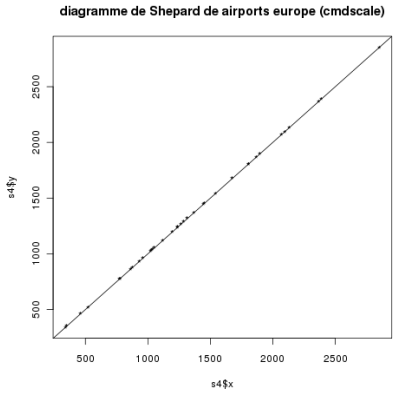


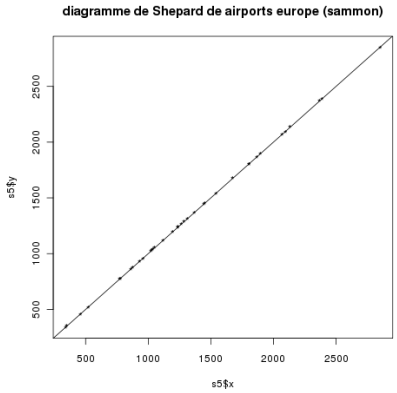
Nous pouvons remarquer que les 3 diagrammes observent la même tendance au niveau de la disgression pour des valeurs intermédiaires. Cependant, pour des faibles valeurs, l'utilisation de la méthode de *Sammon* semble plus précise que les 2 autres méthodes. Donc la méthode de *Sammon* serait plus appropriée pour les petites distances entre aéroports, par exemple les distances des aéroports à l'intérieur d'un continent.

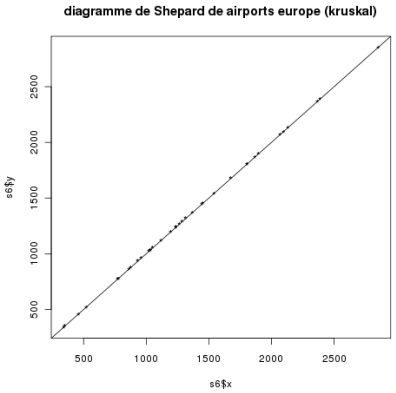
Question 3 :
Nous restreignons maintenant les aéroports à l'Europe :



Nous pouvons observer que les 3 représentations sont identiques au niveau des distances entre les aéroports européens. Vu qu'il s'agit de courtes distances où la représentation en 3 dimensions apporte peu de modifications des distances, ces distances dans un espace à 2 dimensions sont moins assujetties à des variations entre les méthodes utilisées.







Les diagrammes de *Shepard* pour chacune des méthodes nous montrent que les représentations sont identiques et fiables. Nous pouvons affirmer que la qualité des résultats dépend donc de la taille de l'échantillon et de la qualité initiale des données.

Conclusion

Tout au long de ce TP, nous avons pu voir que chaque méthode de positionnement multidimensionnel (*ACP*, *AFTD*, *Sammon* et *Kruskal*) possède ses avantages et ses inconvénients et donc qu'il peut être intéressant de ne pas se restreindre à l'application d'une seule méthode pour analyser un nuage d'individus selon le type de données initiales.
Nous avons pu observer l'utilité du diagramme de *Shepard* pour valider la qualité et la pertinence de l'application d'une méthode sur un type de données.