Positionnement multidimensionnel

Multidimensional Scaling (MDS)

Ricco RAKOTOMALALA

Université Lumière Lyon 2

PLAN

1. MDS – Position du problème
2. MDS classique
3. MDS classique et ACP (analyse en composantes principales)
4. Plus loin avec le MDS
5. Conclusion
6. Bibliographie

Position du problème

Construction d'un système de représentation des individus à partir d'une matrice de distances (ou de dissimilarités, ou de similarités)

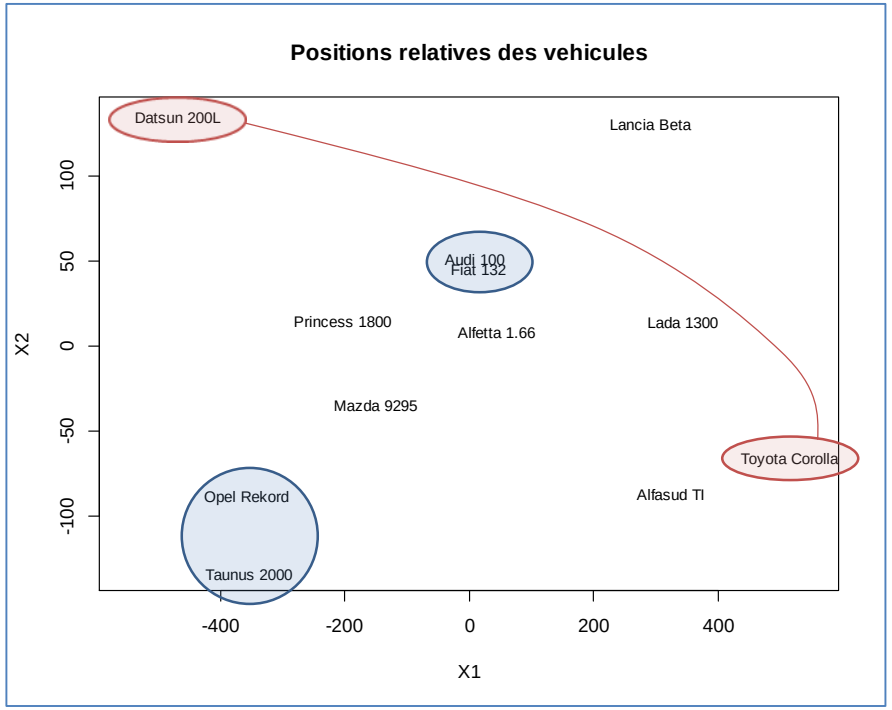
Positionnement multidimensionnel (MDS)

Objectif : A partir d'une matrice de distances entre individus, rendre compte de leurs positions relatives dans un repère euclidien (p, nombre de dimensions à choisir, *p = 2 ou p= 3 souvent*)

Saporta, 2006, page 428 ; restreint aux véhicules étrangers

Lecture : bleu, proches ; rouge : éloignés

	Toyota Corolla	Lada 1300	Alfasud TI	Lancia Beta	Mazda 9295	Fiat 132	Alfetta 1.66	Princess 1800	Audi 100	Taunus 2000	Opel Rekord	Datsun 200L
Toyota Corolla												
Lada 1300	190.2											
Alfasud TI	195.3	106.0										
Lancia Beta	299.3	130.1	219.8									
Mazda 9295	667.2	497.4	477.9	472.4								
Fiat 132	513.6	331.6	336.1	289.8	184.9							
Alfetta 1.66	477.8	301.1	294.5	275.6	204.3	51.0						
Princess 1800	722.4	546.2	536.4	507.6	72.0	220.9	251.8					
Audi 100	521.1	339.9	346.7	295.3	184.2	35.2	73.8	217.2				
Taunus 2000	870.9	712.1	678.2	696.3	225.1	408.8	423.9	211.7	407.5			
Opel Rekord	872.3	708.3	680.5	684.2	213.3	395.0	414.8	187.4	391.8	47.8		
Datsun 200L	1004.6	821.4	822.8	760.4	360.5	492.0	530.4	292.9	486.5	292.3	251.7	



Distance entre deux points quelconques $\delta_{ii'}$

Position des individus $[x_i = (x_{i,1}, x_{i,2})]$ dans un repère centré et orthogonal (p = 2 ici), avec : $\hat{\delta}_{ii'} = \|x_i - x_{i'}\|$

On souhaite que : $\delta_{ii'} \approx \hat{\delta}_{ii'}$

MDS – Quel intérêt ?

Contexte

Parfois les données s'expriment sous la forme de similarités ou dissimilarités entre objets (ex. proximités dans un graphe de réseau social, préférences [classements] de produits par des expérimentateurs, distance ''routières'' entre villes, etc.).

Finalité

Dataviz : rendre compte visuellement des proximités, en disposant d'un critère de qualité de la représentation.

Identifier / interpréter les dimensions (choix de p crucial) qui permettent de discerner les objets.

Se placer dans un espace de représentation (individus x variables) qui permet l'utilisation de techniques de machine learning inapplicables directement sur des matrices de distances.

Positionnement multidimensionnel ‘’classique’’

Analyse factorielle sur tableau de distances

Dissimilarité, distance, distance euclidienne

Dissimilarité

$$\delta_{ii} = 0$$

$$\delta_{ii'} \geq 0, \forall i \neq i'$$

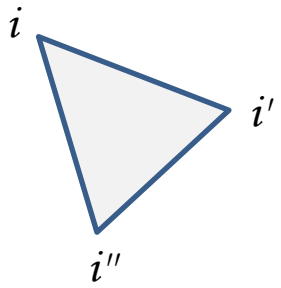
$$\delta_{ii'} = \delta_{i'i}$$

Symétrie

Distance

$$\delta_{ii'} \leq \delta_{ii''} + \delta_{i'i''}$$

Inégalité triangulaire



Distance euclidienne

$$\delta^2_{ii'} = \langle x_i - x_{i'}, x_i - x_{i'} \rangle$$

La distance peut s'exprimer à travers un produit scalaire de l'écart entre les vecteurs individus

''Classical'' MDS – Analyse factorielle sur tableau de distance (1)

La représentation dans l'espace factoriel (X) est centrée c.-à-d.

$$\frac{1}{n}\sum_{i=1}^{n} x_{i,j} = 0, \forall j = 1, \cdots, p$$

Le MDS classique consiste à identifier les coordonnées des n individus (x_i) qui minimise la quantité.

$$\sum_{i,i'} (\langle x_i, x_{i'} \rangle - b_{ii'})^2$$

Où B=($b_{ii'}$) est la matrice des produits scalaires déduite de la matrice des distances via la *formule de Torgerson* (Saporta, 2006, page 182 ; Diday et al., 1982, page 212). L'objectif est donc d'approximer dans l'espace (X) les produits scalaires déduits des distances.

$$b_{ii'} = -\frac{1}{2}\left(\delta^2_{ii'} - \delta^2_{i.} - \delta^2_{.i'} + \delta^2_{..}\right)$$

Où $\delta^2_{i.} = \frac{1}{n}\sum_{i'=1}^{n} \delta^2_{ii'}$ $\delta^2_{..} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i'=1}^{n} \delta^2_{ii'}$

$\delta^2_{.i'} = \frac{1}{n}\sum_{i=1}^{n} \delta^2_{ii'}$

''Classical'' MDS – Analyse factorielle sur tableau de distance (2)

Solution : diagonalisation de la matrice B ($n \times n$), nous obtenons

Les valeurs propres $\lambda_j , j = 1, \cdots, p$

Les vecteurs propres $v_j , j = 1, \cdots, p$

2 à 2 orthogonales

Les coordonnées des individus dans l'espace factoriel pour le j[ème] facteur

$$x_{i,j} = \sqrt{\lambda_j} \times v_{i,j}$$

Si B est semi-définie positive, alors :

$$\lambda_j \geq 0 , j = 1, \cdots, n$$

Problème sinon, il faut trouver une solution pour s'en sortir. Cf. plus loin.

''Classical'' MDS – Algorithme (3)

Desbois, 2005 (section 4.6, page 6)

```
#delta est la matrice initiale des distances
print(delta)

#effectifs
n <- nrow(autos)
print(n)

#sommes delta_i., delta_.iprim, delta_..
d1 <- colSums(delta^2)/n
d2 <- rowSums(delta^2)/n
d3 <- sum(delta^2)/(n^2)

#matrice des produits scalaires - initialisation
b <- delta^2

#boucler
for (i in 1:n){
  for (iprim in 1:n){
    b[i,iprim] <- b[i,iprim] - d1[i] - d2[iprim] + d3
  }
}

#puis
b <- -0.5*b
print(b)

#diagonalisation
vp <- eigen(b)

#valeurs propres
print(vp$values)

#vecteurs propres
print(vp$vectors[,1:2])

#coordonnées factorielles
coord <- vp$vectors[,1:2]
for (j in 1:2){
  coord[,j] <- coord[,j]*sqrt(vp$values[j])
}
#nom des autos +  affichage des coordonnées
rownames(coord) <- rownames(autos)[index]
print(coord)
```

Matrice des distances, points de départ de l'algorithme

	Toyota Corolla	Lada 1300	Alfasud TI	Lancia Beta	Mazda 9295	Fiat 132	Alfetta 1.66	Princess 1800	Audi 100	Taunus 2000	Opel Rekord	Datsun 200L
Toyota Corolla	0.0	190.2	195.3	299.3	667.2	513.6	477.8	722.4	521.1	870.9	872.3	1004.6
Lada 1300	190.2	0.0	106.0	130.1	497.4	331.6	301.1	546.2	339.9	712.1	708.3	821.4
Alfasud TI	195.3	106.0	0.0	219.8	477.9	336.1	294.5	536.4	346.7	678.2	680.5	822.8
Lancia Beta	299.3	130.1	219.8	0.0	472.4	289.8	275.6	507.6	295.3	696.3	684.2	760.4
Mazda 9295	667.2	497.4	477.9	472.4	0.0	184.9	204.3	72.0	184.2	225.1	213.3	360.5
Fiat 132	513.6	331.6	336.1	289.8	184.9	0.0	51.0	220.9	35.2	408.8	395.0	492.0
Alfetta 1.66	477.8	301.1	294.5	275.6	204.3	51.0	0.0	251.8	73.8	423.9	414.8	530.4
Princess 1800	722.4	546.2	536.4	507.6	72.0	220.9	251.8	0.0	217.2	211.7	187.4	292.9
Audi 100	521.1	339.9	346.7	295.3	184.2	35.2	73.8	217.2	0.0	407.5	391.8	486.5
Taunus 2000	870.9	712.1	678.2	696.3	225.1	408.8	423.9	211.7	407.5	0.0	47.8	292.3
Opel Rekord	872.3	708.3	680.5	684.2	213.3	395.0	414.8	187.4	391.8	47.8	0.0	251.7
Datsun 200L	1004.6	821.4	822.8	760.4	360.5	492.0	530.4	292.9	486.5	292.3	251.7	0.0

Matrice des produits scalaires, à diagonaliser

	Toyota Corolla	Lada 1300	Alfasud TI	Lancia Beta	Mazda 9295	Fiat 132	Alfetta 1.66	Princess 1800	Audi 100	Taunus 2000	Opel Rekord	Datsun 200L
Toyota Corolla	269144.1	175489.0	171355.6	140642.0	-75846.6	3895.8	21878.8	-105417.6	662.7	-173264.9	-178063.0	-250476.1
Lada 1300	175489.0	118027.9	109250.5	101403.9	-52516.7	5259.8	15102.8	-69235.7	3138.6	-123158.0	-124005.1	-158757.2
Alfasud TI	171355.6	109250.5	111701.1	82548.5	-46200.1	598.3	13918.3	-67061.1	-2366.8	-102785.4	-107875.5	-163083.6
Lancia Beta	140642.0	101403.9	82548.5	101698.9	-48573.7	10077.8	14313.8	-57053.7	9137.6	-120178.0	-115370.1	-118647.2
Mazda 9295	-75846.6	-52516.7	-46200.1	-48573.7	24325.8	-3712.8	-7283.8	30499.8	-2933.0	58207.4	57258.3	66775.3
Fiat 132	3895.8	5259.8	598.3	10077.8	-3712.8	2431.6	1348.6	-2251.8	2472.4	-10975.2	-8942.2	-202.3
Alfetta 1.66	21878.8	15102.8	13918.3	14313.8	-7283.8	1348.6	2861.6	-9332.8	581.4	-17034.2	-16746.2	-19608.3
Princess 1800	-105417.6	-69235.7	-67061.1	-57053.7	30499.8	-2251.8	-9332.8	41863.8	-776.0	69898.4	71225.3	97641.3
Audi 100	662.7	3138.6	-2366.8	9137.6	-2933.0	2472.4	581.4	-776.0	3751.3	-9796.3	-7013.4	3141.5
Taunus 2000	-173264.9	-123158.0	-102785.4	-120178.0	58207.4	-10975.2	-17034.2	69898.4	-9796.3	142748.1	138078.0	148259.9
Opel Rekord	-178063.0	-124005.1	-107875.5	-115370.1	57258.3	-8942.2	-16746.2	71225.3	-7013.4	138078.0	135693.9	155759.8
Datsun 200L	-250476.1	-158757.2	-163083.6	-118647.2	66775.3	-202.3	-19608.3	97641.3	3141.5	148259.9	155759.8	239196.8

```
[1]  1.110688e+06  7.921646e+04  1.823007e+03  1.440375e+03  2.127499e+02  6.452439e+01
[7]  9.358190e-11  2.486619e-11  1.558165e-11  6.889424e-12  4.709744e-13 -1.368087e-11
```

Valeurs propres, toutes positives (*attention, le dernier, erreur de précision des calculs simplement*)

	[,1]	[,2]
Toyota Corolla	-514.37	-65.80
Lada 1300	-342.77	14.88
Alfasud TI	-322.67	-85.48
Lancia Beta	-290.43	131.45
Mazda 9295	151.93	-33.80
Fiat 132	-13.72	46.16
Alfetta 1.66	-44.53	9.76
Princess 1800	203.30	15.42
Audi 100	-7.65	52.42
Taunus 2000	353.77	-132.49
Opel Rekord	357.45	-88.16
Datsun 200L	469.67	135.64

Coordonnées factorielles. *Attention, signes inversées sur le 1er facteur (cf. page 4). Artefact de calcul simplement, ce sont les positions relatives qui comptent !*

Qualité relative de la représentation

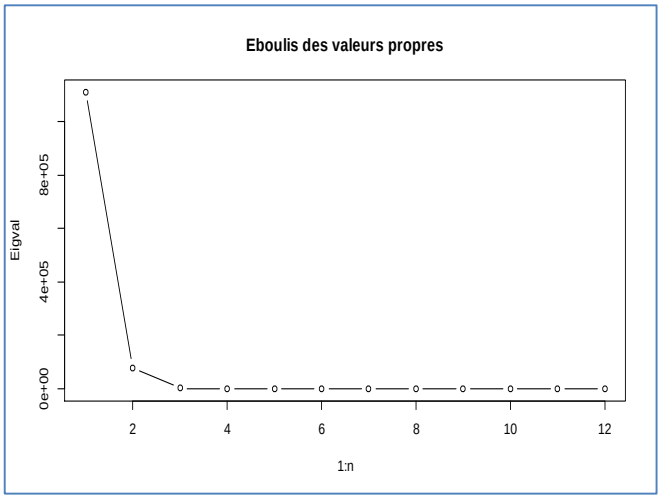
Quantité totale d'information véhiculée par B ➔ trace de la matrice B

$$tr(B) = \sum_{j=1}^{n} \lambda_j$$

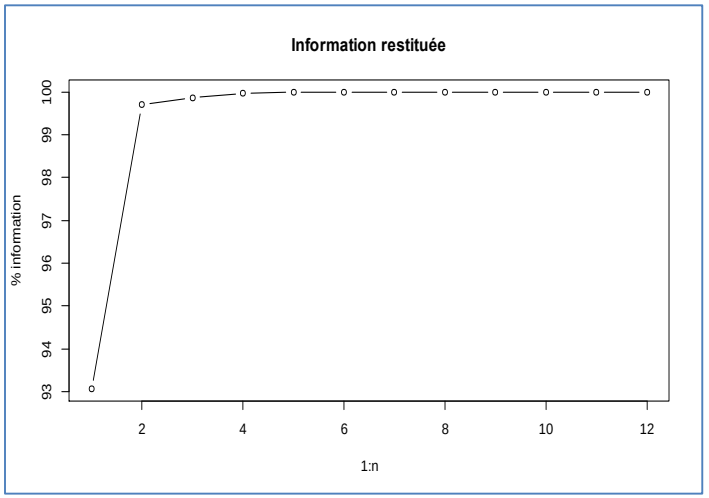
Qualité de l'approximation avec les p premiers facteurs

$$\tau_p = \frac{\sum_{j=1}^{p} \lambda_p}{tr(B)}$$

Un graphique de type « scree plot » permet d'identifier la « bonne » valeur de p



Ou un graphique en proportion de variance restituée



$\tau_2 = 99.7\%$

Matrice des distances restituées

$$\hat{\delta}_{ii'} = \sqrt{\sum_{j=1}^{p} \left(x_{i,j} - x_{i',j}\right)^2}$$

Matrice des distances initiales ($\delta_{ii'}$)

	Toyota Corolla	Lada 1300	Alfasud TI	Lancia Beta	Mazda 9295	Fiat 132	Alfetta 1.66	Princess 1800	Audi 100	Taunus 2000	Opel Rekord	Datsun 200L
Toyota Corolla												
Lada 1300	190.2											
Alfasud TI	195.3	106.0										
Lancia Beta	299.3	130.1	219.8									
Mazda 9295	667.2	497.4	477.9	472.4								
Fiat 132	513.6	331.6	336.1	289.8	184.9							
Alfetta 1.66	477.8	301.1	294.5	275.6	204.3	51.0						
Princess 1800	722.4	546.2	536.4	507.6	72.0	220.9	251.8					
Audi 100	521.1	339.9	346.7	295.3	184.2	35.2	73.8	217.2				
Taunus 2000	870.9	712.1	678.2	696.3	225.1	408.8	423.9	211.7	407.5			
Opel Rekord	872.3	708.3	680.5	684.2	213.3	395.0	414.8	187.4	391.8	47.8		
Datsun 200L	1004.6	821.4	822.8	760.4	360.5	492.0	530.4	292.9	486.5	292.3	251.7	

Matrice des distances restituées ($\hat{\delta}_{ii'}$)

	Toyota Corolla	Lada 1300	Alfasud TI	Lancia Beta	Mazda 9295	Fiat 132	Alfetta 1.66	Princess 1800	Audi 100	Taunus 2000	Opel Rekord	Datsun 200L
Toyota Corolla												
Lada 1300	189.6											
Alfasud TI	192.7	102.4										
Lancia Beta	298.4	127.8	219.3									
Mazda 9295	667.1	497.1	477.4	472.2								
Fiat 132	513.0	330.5	335.8	289.6	183.9							
Alfetta 1.66	475.9	298.3	294.0	274.4	201.2	47.7						
Princess 1800	722.3	546.1	535.6	507.2	71.1	219.2	247.9					
Audi 100	520.3	337.2	343.9	293.6	181.4	8.7	56.4	214.2				
Taunus 2000	870.7	712.0	678.1	696.2	224.7	408.6	422.9	211.0	406.0			
Opel Rekord	872.1	707.8	680.1	684.1	212.6	394.7	413.7	185.7	391.2	44.5		
Datsun 200L	1004.4	821.4	822.6	760.1	360.1	491.6	529.4	292.2	484.5	292.1	250.4	

Un critère possible d'évaluation de la représentation est

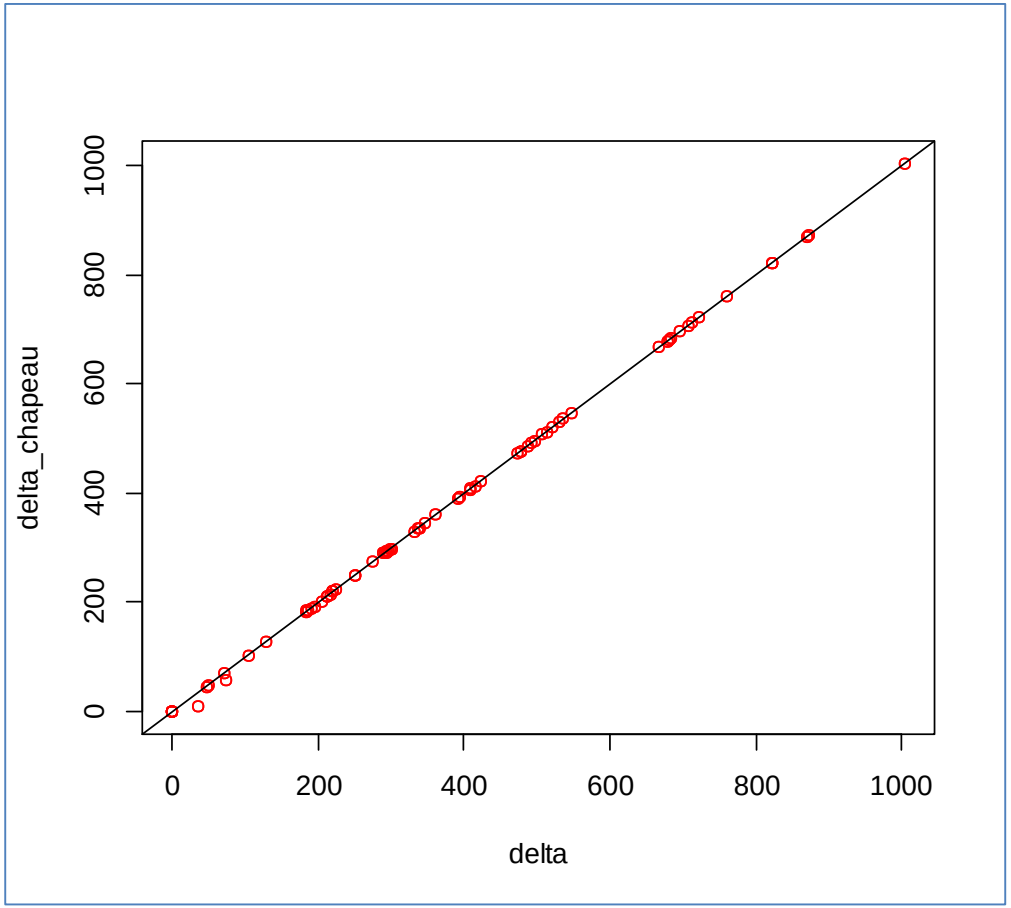
$$Stress\ 1 = \sqrt{\frac{\sum_{i,i'}\left(\hat{\delta}_{ii'} - \delta_{ii'}\right)^2}{\sum_{i,i'} \delta_{ii'}{}^2}} = 0.00896$$

Excellent sur notre exemple

Diagramme de Shepard

Rend compte visuellement de la qualité de la représentation

$\hat{\delta}_{ii'}$



$\delta_{ii'}$

Points alignés sur la première bissectrice, excellente qualité de restitution des distances.

Travailler à partir d'une matrice de similarité

Les données peuvent se présenter initialement sous la forme d'une matrice de similarité ($s_{ii'}$)

$$\begin{cases} s_{ii} = max_s \\ s_{ii'} \leq max_s \end{cases}$$

On peut se ramener à une matrice de dissimilarité via des transformations adaptées. Ex. entres autres…

Sol.1 $$\delta_{ii'} = max_s - s_{ii'}$$

Sol.2 $$\delta_{ii'} = (s_{ii} + s_{i'i'} - 2 \times s_{ii'})^{\frac{1}{2}}$$

On peut dès lors appliquer l'algorithme MDS

Cas de B non-définie positive

Problème : B peut être non-définie positive c.-à-d. certaines valeurs propres sont négatives. C'est le cas lorsque ($\delta_{ii'}$) n'est pas une distance euclidienne ou, pire, est une simple dissimilarité (sans l'inégalité triangulaire)

Stratégie 1

Prendre en compte uniquement les facteurs associés aux valeurs propres positives. Solution très simple mais/et opérationnelle.

Stratégie 2

Ajouter une constante positive c aux éléments hors diagonale principale de la matrice B ($B^* : b^*_{ii'} = b_{ii'} + c \, , \forall \, i \neq i'$) de manière à ce que toutes les valeurs propres de B* soient positives. Attention, si c est trop grand, l'information est détériorée, la représentation sera déformée.

Caillez (1983)

Propose une solution analytique permettant de trouver **le plus petit** c assurant ($\lambda^*_j \geq 0 \, , \forall j$) (Saporta, 2006 ; page 183).

Implémentée dans cmdscale() de R par ex.

MDS et ACP

Equivalence lorsque l'on utilise la distance euclidienne

Analyse en composantes principales

L'ACP est une technique de visualisation qui travaille à partir d'un tableau (individus x variables) et permet de projeter les observations dans un espace de dimensionnalité réduite en minimant la perte d'information. Dans le cas où ($\delta_{ii'}$) est une distance euclidienne, elle est totalement équivalente au MDS.

autos

Modele	CYL	PUISS	LONG	LARG	POIDS	VMAX
Alfasud TI	1350	79	393	161	870	165
Audi 100	1588	85	468	177	1110	160
Fiat 132	1585	98	439	164	1105	165
Lancia Beta	1297	82	429	169	1080	160
Toyota Corolla	1166	55	399	157	815	140
Alfetta 1.66	1570	109	428	162	1060	175
Princess 1800	1798	82	445	172	1160	158
Datsun 200L	1998	115	469	169	1370	160
Taunus 2000	1993	98	438	170	1080	167
Mazda 9295	1769	83	440	165	1095	165
Opel Rekord	1979	100	459	173	1120	173
Lada 1300	1294	68	404	161	955	140

```
#ACP non normée (matrice de covariance)
acp <- princomp(autos,cor=FALSE,scores=TRUE)
print(acp$scores[,1:2])

#matrice des distances euclidiennes
dautos <- dist(autos,method="euclidean")

#positionnement dimensionnel sur dautos
mds <- cmdscale(dautos,k=2,eig=TRUE)
print(mds$points[,1:2])
```

	Comp.1	Comp.2
Alfasud TI	-322.67	85.48
Audi 100	-7.65	-52.42
Fiat 132	-13.72	-46.16
Lancia Beta	-290.43	-131.45
Toyota Corolla	-514.37	65.80
Alfetta 1.66	-44.53	-9.76
Princess 1800	203.30	-15.42
Datsun 200L	469.67	-135.64
Taunus 2000	353.77	132.49
Mazda 9295	151.93	33.80
Opel Rekord	357.45	88.16
Lada 1300	-342.77	-14.88

Coordonnées factorielles de l'ACP

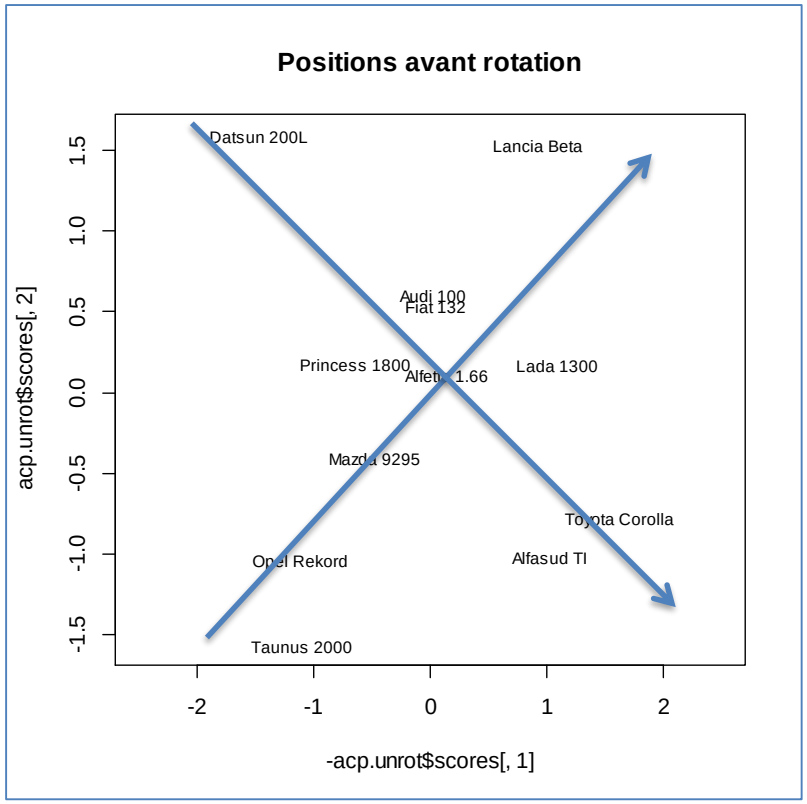
	[,1]	[,2]
Alfasud TI	322.67	85.48
Audi 100	7.65	-52.42
Fiat 132	13.72	-46.16
Lancia Beta	290.43	-131.45
Toyota Corolla	514.37	65.80
Alfetta 1.66	44.53	-9.76
Princess 1800	-203.30	-15.42
Datsun 200L	-469.67	-135.64
Taunus 2000	-353.77	132.49
Mazda 9295	-151.93	33.80
Opel Rekord	-357.45	88.16
Lada 1300	342.77	-14.88

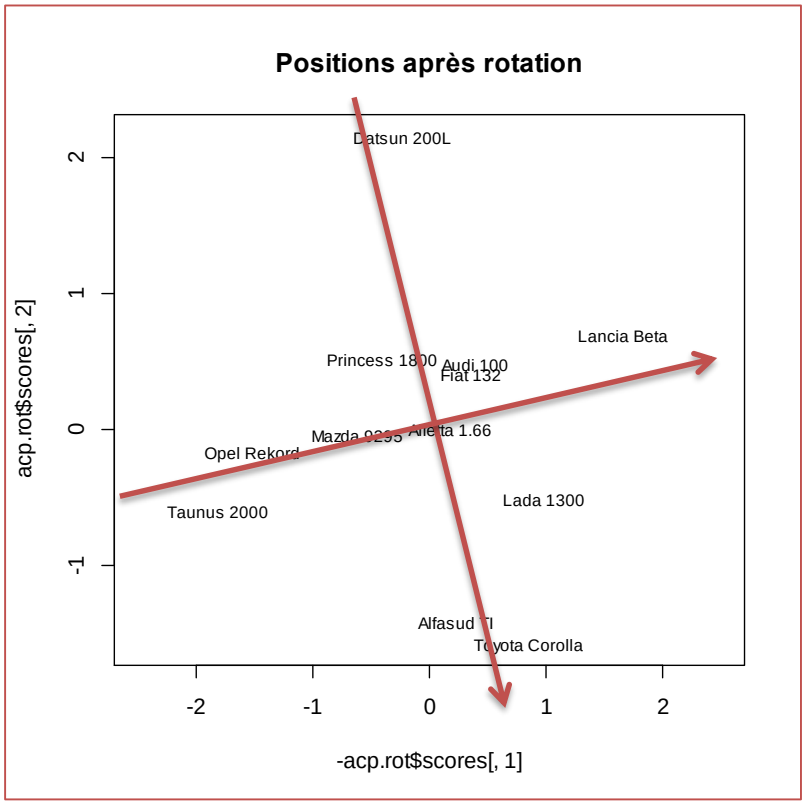
Coordonnées factorielles du MDS

On peut montrer formellement l'équivalence (Desbois, 2005 ; page 17).

Rotation des facteurs

Comme en ACP, il est possible de faire pivoter les axes (rotation orthogonale) pour améliorer l'interprétation des résultats, en situant mieux les directions des oppositions et concomitances. Attention néanmoins, chaque méthode a son rôle, le MDS est focalisé sur la restitution des proximités, pas par la construction des axes.





Rotation VARIMAX ici, contraster plus fortement les contributions des individus aux axes.

Plus loin avec le MDS

MDS métrique vs. MDS non-métrique, Individus supplémentaires, MDS sur corrélations

MDS Métrique (mMDS) vs. MDS non-métrique (nMDS)

MDS métrique

Généralisation du MDS classique où l'on essaie d'optimiser explicitement une fonction de perte (*stress*) via une heuristique.

Où f() est une fonction affine de la distance f(δ) = a + b x δ

$$stress = \frac{\sum_{i,i'}\left[\hat{\delta}_{ii'} - f(\delta_{ii'})\right]^2}{\sum_{i,i'} {\delta_{ii'}}^2}$$

Remarque : si f(δ) = δ, la solution est différente du MDS classique !

MDS non-métrique

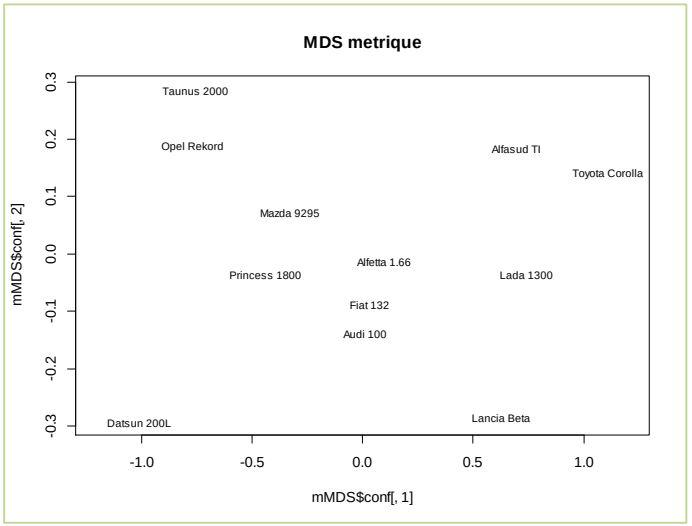
Lorsque ce n'est pas tant les valeurs et les écarts entre les $\delta_{ii'}$ qui nous intéressent, mais plutôt leur ordre (l'objet A est proche de B, éloigné de C, plus éloigné encore de D, etc. – on souhaite restituer cet ordonnancement dans le repère factoriel).

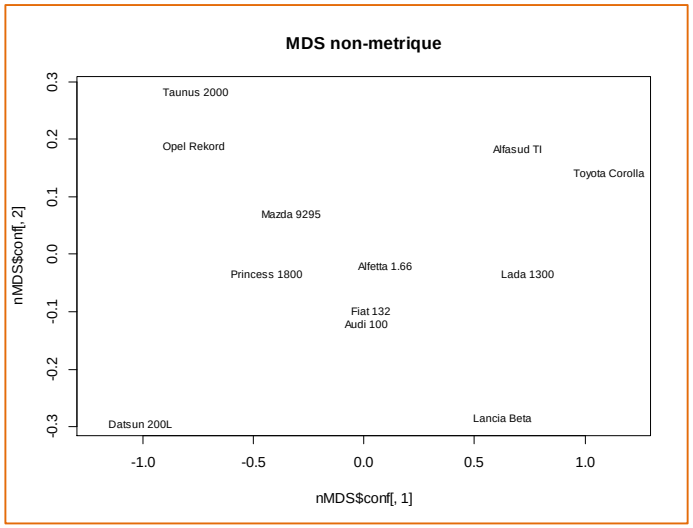
On optimise le même *stress*, mais f() est une fonction monotone qui a pour propriété de préserver l'ordre

$$\delta_{ii'} < \delta_{kk'} \Leftrightarrow f(\delta_{ii'}) < f(\delta_{kk'})$$

mMDS vs. nMDS - Exemple

```
#smacof package
library(smacof)

#métrique - interval - f = a + b x delta
mMDS <- smacof::smacofSym(dautos,ndim=2,type="interval")

plot(mMDS$conf[,1],mMDS$conf[,2],type="n",main="MDS metrique")
text(mMDS$conf[,1],mMDS$conf[,2],label=rownames(autos))

#non-métrique
nMDS <- smacof::smacofSym(dautos,ndim=2,type="ordinal")

plot(nMDS$conf[,1],nMDS$conf[,2],type="n",main="MDS non-metrique")
text(nMDS$conf[,1],nMDS$conf[,2],label=rownames(autos))
```





Sur notre matrice où les $\delta_{ii'}$ sont des distances euclidiennes calculées à partir de données tabulaires, les deux approches ne se démarquent pas vraiment.

Traitement d'un individu supplémentaire (1)

A partir du vecteur de distance $\delta_k=(\delta_{k1},\ldots, \delta_{ki},\ldots, \delta_{kn})$ d'un individu supplémentaire k avec chaque objet de la base (i = 1, …, n), il est possible de calculer ses coordonnées factorielles à partir des résultats de la MDS.

Transformer la distance en produit scalaire

$$b_{ki} = -\frac{1}{2}\left(\delta_{ki}^2 - \delta_{k.}^2 - \delta_{.i}^2 + \delta_{..}^2\right)$$

Calculés une fois pour toutes à partir de la matrice initiale des distances ayant servi pour la construction du repère MDS.

Calculer les coordonnées de l'individu supplémentaire

$$x_{kj} = \frac{1}{\lambda_j}\sum_{i=1}^{n} b_{ki} \times x_{i,j}$$

Où $x_{i,j}$ est la coordonnée de l'individu i (appartenant à la base d'apprentissage) pour le facteur j

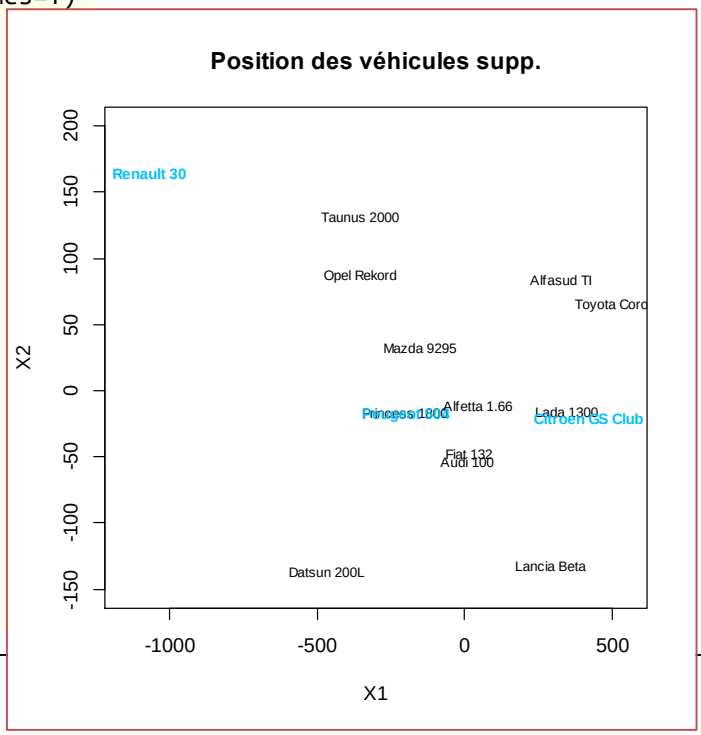
Traitement d'un individu supplémentaire (2)

Notre exemple est un peu spécifique, les distances ne sont pas fournies d'office, nous devons les calculer à partir d'un tableau « individus x variables ».

```
#library
library(openxlsx)
#chargement
autos <- openxlsx::read.xlsx("autos_mds.xlsx",sheet=1,colNames=T,rowNames=T)
#effectifs
n <- nrow(autos)
#distance entre les autos pris 2 à 2 (δ_ii')
dautos <- dist(autos,method="euclidean")
mautos <- as.matrix(dautos)
#moyenne des distances^2 par véhicule (δ²_i.)
moydist <- colMeans(mautos^2)
#moyenne globale des distances^2 (δ²_..)
moydistglob <- mean(mautos^2)
#MDS classique
mds <- cmdscale(dautos,k=2,eig=TRUE)
#ouvrir le fichier de l'individu supp.
autos.supp <- openxlsx::read.xlsx("autos_mds_supp.xlsx",sheet=1,colNames=T,rowNames=T)
#coord. pour chaque véhicule
coordSupp <- matrix(0,nrow=nrow(autos.supp),ncol=2)
#calculer les coordonnées
for (k in 1:nrow(autos.supp)){
  #distance euclidienne avec chaque point actif (δ_ki)
  d <- apply(as.matrix(autos),1,function(x){sqrt(sum((x-autos.supp[k,])^2))})
  #produit scalaire (b_ki)
  b <- -0.5*(d^2-mean(d^2)-moydist+moydistglob)
  #coordonnées (x_kj)
  coordSupp[k,1] <- sum(b*mds$points[,1])/mds$eig[1]
  coordSupp[k,2] <- sum(b*mds$points[,2])/mds$eig[2]
}
#intégrer dans le plan, avec les individus actifs
plot(mds$points[,1],mds$points[,2],type="n",main="Position des véhicules supp.",xlab="X1",ylab="X2",xlim=c(-1150,550),ylim=c(-150,200))
text(mds$points[,1],mds$points[,2],label=rownames(autos),cex=0.7)
text(coordSupp[,1],coordSupp[,2],label=rownames(autos.supp),cex=0.75,col="deepskyblue",font=2)
```

Pour chaque ind. supp., les plus proches et les plus éloignés.

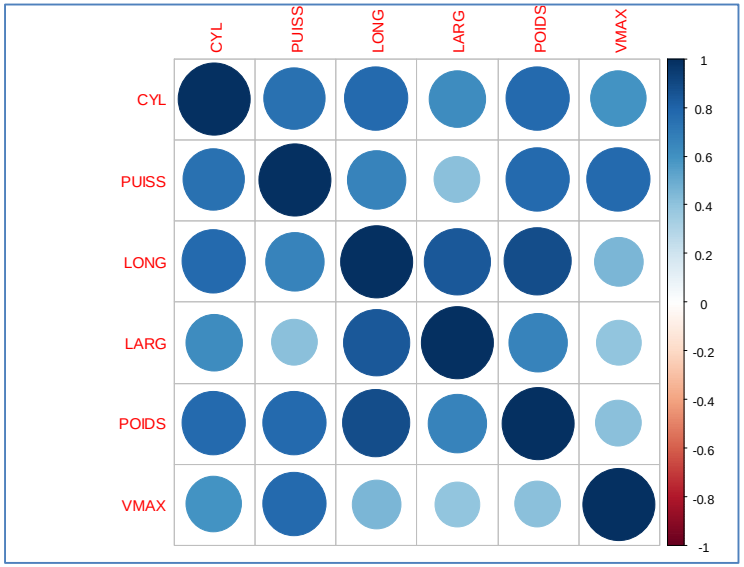
```
[1] "Citroen GS Club"
    Alfasud TI       Audi 100       Fiat 132     Lancia Beta
     144.71005      412.92251      406.01601      170.55204
Toyota Corolla   Alfetta 1.66 Princess 1800   Datsun 200L
     129.16269      375.94813      621.66229      895.71536
   Taunus 2000     Mazda 9295    Opel Rekord     Lada 1300
     787.06734      572.71808      783.36901       77.94229
[1] "Renault 30"
    Alfasud TI       Audi 100       Fiat 132     Lancia Beta
     1391.1675      1097.4502      1100.8365      1389.0104
Toyota Corolla   Alfetta 1.66 Princess 1800   Datsun 200L
     1583.9896      1124.9529       882.1598       668.5282
   Taunus 2000     Mazda 9295    Opel Rekord     Lada 1300
      713.5229       924.1796       714.2178      1420.4834
[1] "Peugeot 504"
    Alfasud TI       Audi 100       Fiat 132     Lancia Beta
     535.104663     215.083705     219.437918     505.812218
Toyota Corolla   Alfetta 1.66 Princess 1800   Datsun 200L
     720.653176     250.812679       7.348469     294.339940
   Taunus 2000     Mazda 9295    Opel Rekord     Lada 1300
     214.151815      72.027772     189.755105     544.458447
```



Travailler sur les corrélations (1)

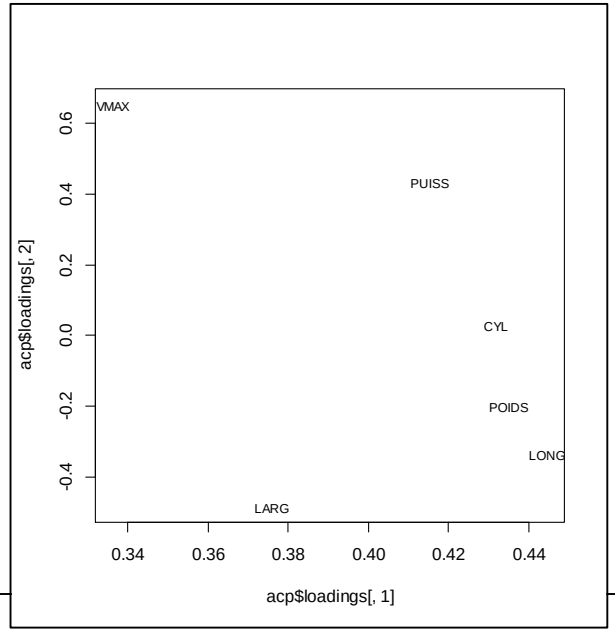
Le principe du MDS peut-être appliqué à l'étude des corrélations, en particulier pour identifier les structures de relations entre les variables, les redondances, les oppositions, en vue d'une sélection de variables par ex.

```
#library corrplot
library(corrplot)
corrplot(cor(autos))

#ACP avec autos - acp normée (matrice de corrélation)
acp <- princomp(autos,cor=TRUE,scores=TRUE)
plot(acp$loadings[,1],acp$loadings[,2],type="n")
text(acp$loadings[,1],acp$loadings[,2],label=colnames(autos),cex=0.75)
```



CYL est liée à peu près avec les autres variables, VMAX avec PUISS seulement, etc. Comment rendre compte de cela si on a des dizaines de variables ?

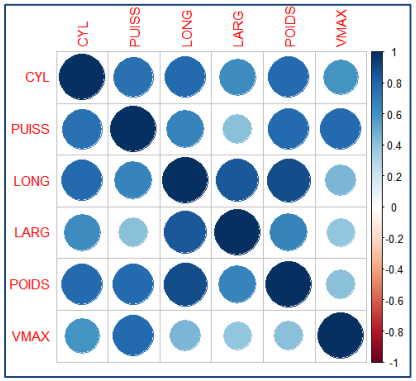
L'ACP est axé sur la création des composantes avec une logique intrinsèque, le cercle des corrélations ne rend pas toujours compte des liaisons entres variables(mais ça marche plutôt bien ici).

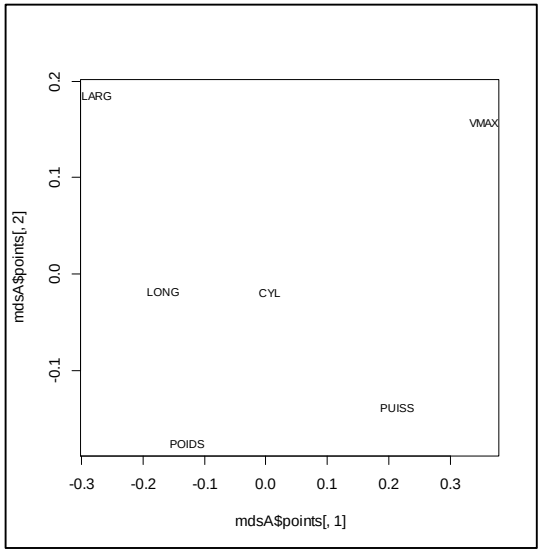


Travailler sur les corrélations (2) – Convertir la corrélation en distance

Solution.1 : $\delta_{jj'} = 1 - r_{jj'}$

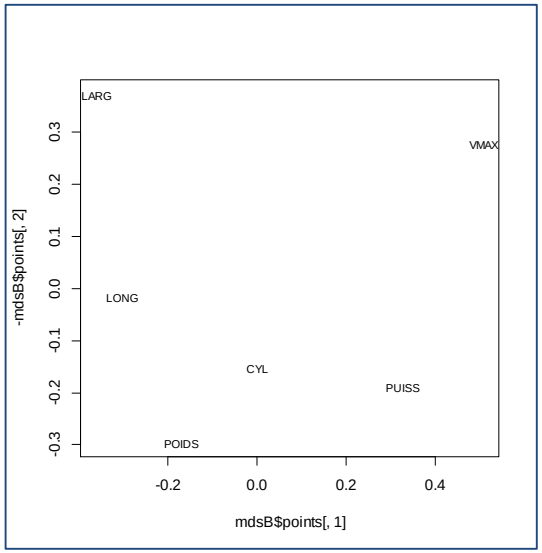
Tenir compte du sens de la relation





Solution.2 : $\delta_{jj'} = \sqrt{1 - r_{jj'}^2}$

Ne pas tenir compte du sens de la relation, qui peut être artificielle parfois. Ex. Véhicules : poids et accélération (a) en secondes pour 100 km/h D.A. ou (b) en vitesse aux 400 mètres ➔ (a) r > 0, (b) r < 0



Dans les deux cas, les proximités (corrélations) sont respectées.

Bon, dans notre exemple, toutes les corrélations sont positives, les deux solutions ne se démarquent pas !

Conclusion

Positionnement multidimensionnel

Conclusion

MDS permet de visualiser les proximités entre des objets dans un espace de dimension réduite. Avec un niveau de fidélité que l'on peut évaluer.

Au-delà de la visualisation, il permet également de comprendre les principales structures qui existent dans les données en mettant à jour les oppositions et les proximités.

MDS s'applique dès lors que nous ne disposons pas en entrée de données tabulaires « individus x variables » mais plutôt une matrice de similarité ou de dissimilarité.

Ca peut être le cas notamment des graphes sociaux (cf. cours web mining).

MDS peut préparer le terrain pour les techniques de machine learning qui ont absolument besoin de travailler à partir de tableaux « individus x variables ».

MDS classique est aussi appelé Principal Coordinates Analysis (PCoA) dans la littérature.

Bibliographie

Diday E., Lemaire J., Pouget J., Testu F., « Eléments d'analyse de données », Dunod, 1982 ; Section 2.3, « Analyse factorielle sur un tableau de distance », pages 207-221.

Saporta G., « Probabilités, Analyse de données et Statistique », Technip, 2006 ; Section 7.5, « Analyse factorielle sur tableaux de distance et de dissimilarités », pages 181-184.

Desbois D., « Une introduction au positionnement multidimensionnel », Revue MODULAD, n°32, pages 1-28, 2005.

Escoufier Y., « Le positionnement multidimensionnel », Revue de Statistique Appliquée, Tome 23, n°4, pages 5-14, 1975.

Abdi H., « Metric Multidimensional Scaling (MDS): Analyzing Distance Matrices », in Neil Salkind (Ed.), Encyclopedia of Measurement and Statistics, Sage, 2007.