Fiche TD avec le logiciel : tdr82

Analyse en coordonnées principales

A.B. Dufour & D. Chessel

La fiche introduit à la manipulation des matrices de distances. Une configuration de points dans un tableau donne une matrice de distances. Une matrice de distances donne une configuration de points. Différents exemples biologiques sont analysés.

Table des matières

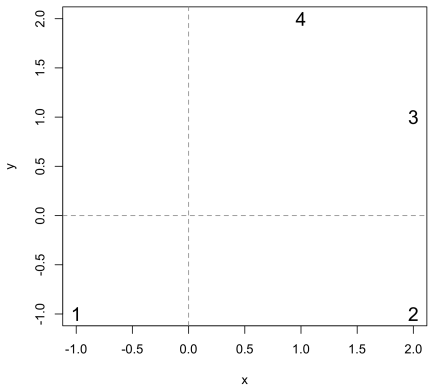
1 Le principe de l'analyse par l'exemple

On considère un nuage de quatre points dans $\mathbb{R}^2$ dont les coordonnées, pour l'individu i sont x_i et y_i.

```
dpoints <- matrix(c(-1,-1,2,-1,2,1,1,2), byrow=T, ncol=2)
colnames(dpoints) <- c("x","y")
dpoints <- data.frame(dpoints)
dpoints
```

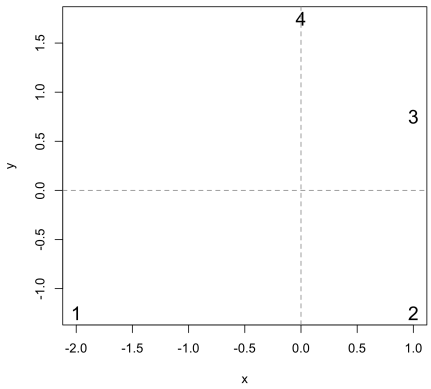
```
  x  y
1 -1 -1
2  2 -1
3  2  1
4  1  2
```

```
plot(dpoints, type="n")
text(dpoints,lab=rownames(dpoints),cex=1.5)
abline(v=0, col=grey(0.6), lty=2)
abline(h=0, col=grey(0.6), lty=2)
```



1. On représente le nuage de points en plaçant l'origine au centre de gravité.

```
dptscent <- scale(dpoints, center=TRUE, scale=FALSE)
plot(dptscent, type="n")
text(dptscent,lab=rownames(dpoints),cex=1.5)
abline(v=0, col=grey(0.6), lty=2)
abline(h=0, col=grey(0.6), lty=2)
```



2. On calcule, à l'aide de la fonction `dist`, la matrice des distances canoniques entre ces points notée `dt`. Dans $\mathbb{R}^2$, la distance entre deux points i et j est donnée par

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

```
       1      2      3      4
1 0.0000 3.0000 3.6056 3.6056
2 3.0000 0.0000 2.0000 3.1623
3 3.6056 2.0000 0.0000 1.4142
4 3.6056 3.1623 1.4142 0.0000
```

3. On calcule la matrice $\mathbf{H} = \left[-\frac{1}{2}d_{ij}^2\right]_{\bullet\bullet}$ où les points désignent le double centrage par ligne et par colonne.

```
matdt <- as.matrix(dt)
matH <- -matdt*matdt/2
matH <- t(matH - colMeans(matH)) - rowMeans(matH) + mean(matH)
matH
```

```
        1       2       3       4
1  5.5625 -0.4375 -2.9375 -2.1875
2 -0.4375  2.5625  0.0625 -2.1875
3 -2.9375  0.0625  1.5625  1.3125
4 -2.1875 -2.1875  1.3125  3.0625
```

Noter que l'on retrouve le même résultat avec :

```
library(ade4)
bicenter.wt(-0.5*matdt*matdt)
```

```
        1       2       3       4
1  5.5625 -0.4375 -2.9375 -2.1875
2 -0.4375  2.5625  0.0625 -2.1875
3 -2.9375  0.0625  1.5625  1.3125
4 -2.1875 -2.1875  1.3125  3.0625
```

4. On calcule les valeurs propres associées à la matrice $\mathbf{H}$.

```
eigen(matH)$values
```

```
[1]  8.409853e+00  4.340147e+00 -4.440892e-16 -1.776357e-15
```

```
eigen(matH)$values[1:2] -> valpro
```

5. On calcule les deux premiers vecteurs propres de $\mathbf{H}$ de normes les racines des valeurs propres.

```
eigen(matH)$vectors[,1:2] -> vecpro
norme <- function(x) sqrt(sum(x*x))
apply(vecpro,2,norme)
```

```
[1] 1 1
```

```
t(vecpro)*sqrt(valpro)
```

```
           [,1]      [,2]       [,3]      [,4]
[1,]  2.2391581 0.3232505 -1.2157677 -1.346641
[2,] -0.7407234 1.5678039  0.2905322 -1.117613
```

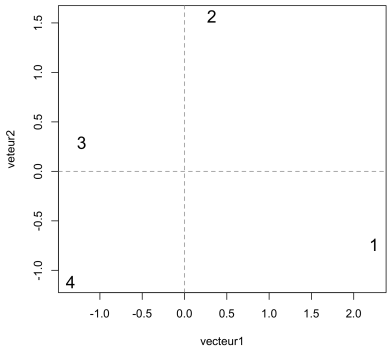
```
vecpro <- t( t(vecpro)*sqrt(valpro))
colnames(vecpro) <- c("vecteur1","veteur2")
vecpro
```

```
      vecteur1    veteur2
[1,]  2.2391581 -0.7407234
[2,]  0.3232505  1.5678039
[3,] -1.2157677  0.2905322
[4,] -1.3466409 -1.1176127
```

6. On représente le nouveau nuage de points et on donne la matrice des distances correspondantes.

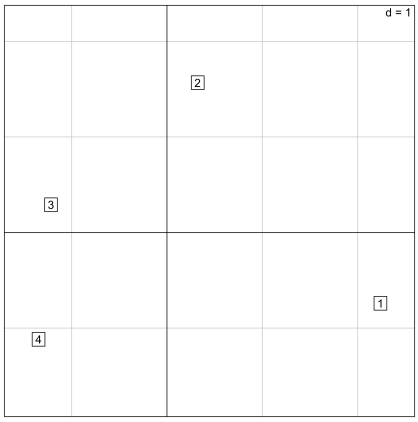
```
plot(vecpro,type="n")
text(vecpro,lab=rownames(dpoints),cex=1.5)
abline(v=0, col=grey(0.6), lty=2)
abline(h=0, col=grey(0.6), lty=2)
dist(vecpro, method = "euclidean", diag = T, upper = T)
```

```
         1        2        3        4
1 0.000000 3.000000 3.605551 3.605551
2 3.000000 0.000000 2.000000 3.162278
3 3.605551 2.000000 0.000000 1.414214
4 3.605551 3.162278 1.414214 0.000000
```



Conclusion. Une matrice de distances donne une configuration de points. On vient de réaliser une analyse en coordonnées principales "à la main".
Tous ces calculs s'opèrent dans la fonction `dudi.pco` de la librairie `ade4`.

```
pco1 <- dudi.pco(dt, scannf=FALSE, nf=2)
scatter(pco1, posieig="none")
```



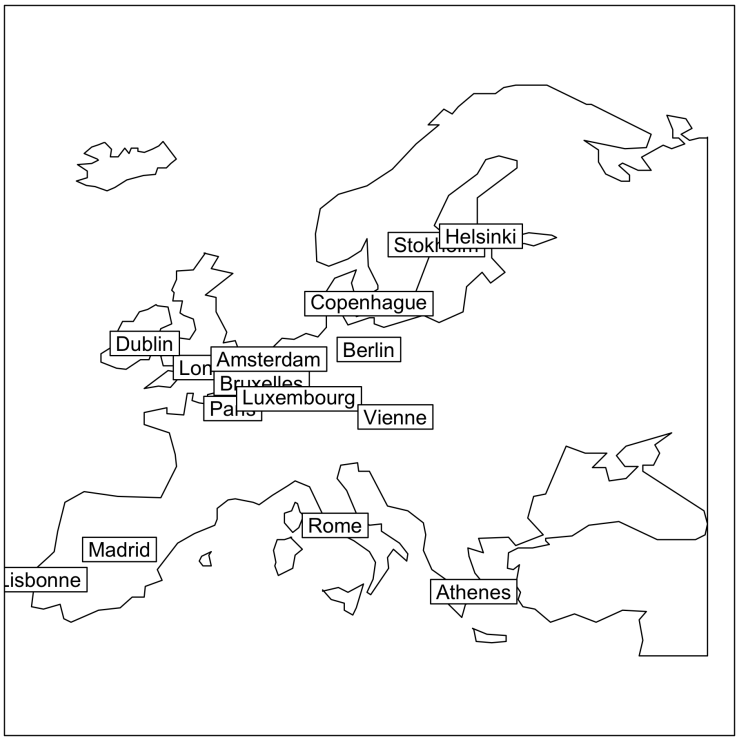
2 Distances entre les capitales Européennes

On représente les capitales de l'Europe occidentale.

```
data(capitales)
names(capitales)
```

```
[1] "df"   "xy"   "area" "logo"
```

```
temp <- capitales$area
area.plot(temp)
s.label(capitales$xy, lab = names(capitales$df), add.plot=TRUE)
```



2.1 Distances à vol d'oiseaux

1. Quelle est la nature de l'objet `capitales$xy`
2. Réaliser l'analyse en composantes principales centrée des positions géographiques des capitales.
3. A l'aide de la fonction `dist`, calculer la matrice des distances euclidiennes (en pixels) entre les capitales. Donner par exemple la distance entre Paris et Londres.
4. Réaliser l'analyse en coordonnées principales de cette matrice de distances.
5. Commenter.

2.2 Distances routières

L'objet `capitales$df` contient les distances routières entre les capitales d'Europe.

1. Quelle est la nature de cet objet ?
2. Transformer `capitales$df` en une matrice de distances.
3. Réaliser l'analyse en coordonnées principales de cette matrice de distance. Commenter le "non résultat" obtenu.

3 Distances euclidiennes

Quand il existe une configuration de n points dans un espace euclidien dont les distances deux à deux sont celles de la matrice de distances, on dit que cette matrice de distances est euclidienne.
Pour qu'il en soit ainsi, il faut et il suffit que $\mathbf{H}$ ait des valeurs propres positives ou nulles.

```
is.euclid(distcap, print=TRUE)
```

```
 [1]  8.672165e+04  3.508285e+04  1.255213e-11  8.322316e-12  5.776558e-12
 [6]  3.707984e-12  1.112836e-12  1.216155e-15 -5.990375e-13 -1.298682e-12
[11] -1.510669e-12 -3.200594e-12 -4.910066e-12 -7.519561e-12 -9.300155e-12
[1] TRUE
```

```
is.euclid(distrout, print=TRUE)
```

```
 [1]  1.771252e+07  1.043099e+07  2.856106e+06  1.311331e+06  7.901565e+05
 [6]  3.974890e+05  1.154366e+05  2.355395e+04 -2.886788e-09 -2.908168e+04
[11] -1.654115e+05 -2.966759e+05 -8.083921e+05 -1.438154e+06 -2.120982e+06
[1] FALSE
```

Définitions
On considère $\mathcal{E}$ un ensemble d'objets comme ici, les capitales européennes. Sur chaque objet, p variables ont été mesurées. La distance entre deux objets i et j de $\mathcal{E}$ est définie par :

$$d_{ij} = \sqrt{\sum_{\ell=1}^{p} (x_{i\ell} - x_{j\ell})^2}$$

Propriétés d'une distance euclidienne.
- Pour tout objet i de $\mathcal{E}$, $d_{ii} = 0$
- Pour tout i et tout j de $\mathcal{E}$, $d_{ij} = d_{ji}$
- Pour tout i, tout j et tout k de $\mathcal{E}$, $d_{ij} \leq d_{ik} + d_{kj}$

4 Rendre une distance euclidienne

Il existe plusieurs méthodes pour rendre une distance euclidienne. Les fonctions associées s'appliquent à des objets de la classe `dist` (librairies `MASS` et `mva` à télécharger).
Pour discuter de la nature euclidienne ou non d'une distance, nous nous appuierons sur les données `yanomama`, liste à trois composantes :
`$geo`, distances géographiques (en km),
`$gen`, distances génétiques,

`$ant`, distances anthropométriques.
La source de ces données se trouve dans Spielman [4] et Manly [3].

```
data(yanomama)
names(yanomama)
```

```
[1] "geo" "gen" "ant"
```

```
yanomama$geo
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
 [1,]    0    9   28  152  149  169  172  253  244    82    99   100    97    52
 [2,]    9    0   20  161  158  175  178  259  250    90   108   108   106    47
 [3,]   28   20    0  178  175  184  187  276  267   110   127   126   124    34
 [4,]  152  161  178    0    7  102   98  232  222    87    57    53    55   181
 [5,]  149  158  175    7    0   95   91  238  228    88    57    52    54   177
 [6,]  169  175  184  102   95    0    7  330  320   151   126   115   114   168
 [7,]  172  178  187   98   91    7    0  327  317   150   124   113   113   172
 [8,]  253  259  276  232  238  330  327    0   10   195   210   220   222   306
 [9,]  244  250  267  222  228  320  317   10    0   185   200   210   212   296
[10,]   82   90  110   87   88  151  150  195  185     0    30    39    38   127
[11,]   99  108  127   57   57  126  124  210  200    30     0    11    12   137
[12,]  100  108  126   53   52  115  113  220  210    39    11     0     3   134
[13,]   97  106  124   55   54  114  113  222  212    38    12     3     0   131
[14,]   52   47   34  181  177  168  172  306  296   127   137   134   131     0
[15,]   35   39   56  153  152  190  192  221  212    70    96   100    98    86
[16,]   98  107  127   77   79  152  151  186  176    16    27    37    38   143
[17,]   82   89  102   86   81   90   92  266  256    72    59    50    48    98
[18,]   86   94  107   80   75   88   89  264  253    71    55    46    44   103
[19,]   88   97  117   91   92  161  160  184  174    11    37    47    47   136
      [,15] [,16] [,17] [,18] [,19]
 [1,]    35    98    82    86    88
 [2,]    39   107    89    94    97
 [3,]    56   127   102   107   117
 [4,]   153    77    86    80    91
 [5,]   152    79    81    75    92
 [6,]   190   152    90    88   161
 [7,]   192   151    92    89   160
 [8,]   221   186   266   264   184
 [9,]   212   176   256   253   174
[10,]    70    16    72    71    11
[11,]    96    27    59    55    37
[12,]   100    37    50    46    47
[13,]    98    38    48    44    47
[14,]    86   143    98   103   136
[15,]     0    85   100   103    73
[16,]    85     0    80    78    14
[17,]   100    80     0     6    83
[18,]   103    78     6     0    81
[19,]    73    14    83    81     0
```

C'est une matrice de distances complète (matrice inférieure, diagonale et matrice supérieure).

```
inherits(yanomama$geo,"dist")
```

```
[1] FALSE
```

```
geo <- as.dist(yanomama$geo)
inherits(geo,"dist")
```

```
[1] TRUE
```

```
geo
```

```
     1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
2    9
3   28  20
4  152 161 178
5  149 158 175   7
6  169 175 184 102  95
7  172 178 187  98  91   7
8  253 259 276 232 238 330 327
9  244 250 267 222 228 320 317  10
10  82  90 110  87  88 151 150 195 185
11  99 108 127  57  57 126 124 210 200  30
12 100 108 126  53  52 115 113 220 210  39  11
13  97 106 124  55  54 114 113 222 212  38  12   3
14  52  47  34 181 177 168 172 306 296 127 137 134 131
15  35  39  56 153 152 190 192 221 212  70  96 100  98  86
16  98 107 127  77  79 152 151 186 176  16  27  37  38 143  85
17  82  89 102  86  81  90  92 266 256  72  59  50  48  98 100  80
18  86  94 107  80  75  88  89 264 253  71  55  46  44 103 103  78   6
19  88  97 117  91  92 161 160 184 174  11  37  47  47 136  73  14  83  81
```

```
unclass(geo)
```

```
  [1]   9  28 152 149 169 172 253 244  82  99 100  97  52  35  98  82  86  88  20 161
 [21] 158 175 178 259 250  90 108 108 106  47  39 107  89  94  97 178 175 184 187 276
 [41] 267 110 127 126 124  34  56 127 102 107 117   7 102  98 232 222  87  57  53  55
 [61] 181 153  77  86  80  91  95  91 238 228  88  57  52  54 177 152  79  81  75  92
 [81]   7 330 320 151 126 115 114 168 190 152  90  88 161 327 317 150 124 113 113 172
[101] 192 151  92  89 160  10 195 210 220 222 306 221 186 266 264 184 185 200 210 212
[121] 296 212 176 256 253 174  30  39  38 127  70  16  72  71  11  11  12 137  96  27
[141]  59  55  37   3 134 100  37  50  46  47 131  98  38  48  44  47  86 143  98 103
[161] 136  85 100 103  73  80  78  14   6  83  81
attr(,"Size")
[1] 19
attr(,"call")
as.dist.default(m = yanomama$geo)
attr(,"Diag")
[1] FALSE
attr(,"Upper")
[1] FALSE
```

C'est le jeu de données sur les fonctions génériques qui crée cette différence. `geo` est un objet `dist`. `geo` appelle `print(geo)`. `unclass(geo)` est un objet sans classe qui appelle `print.default`.

```
is.euclid(geo, print=TRUE)
```

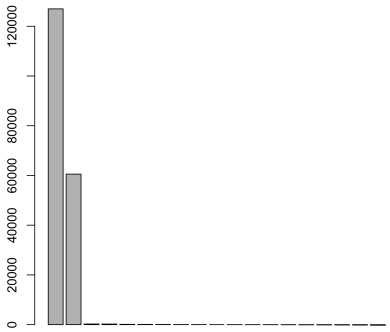
```
 [1]  1.270026e+05  6.054361e+04  2.596671e+02  2.420916e+02  1.291763e+02
 [6]  9.500950e+01  6.269167e+01  3.382556e+01  2.016325e+01 -9.041212e-13
[11] -1.318136e+01 -2.167582e+01 -2.942247e+01 -4.815153e+01 -7.937122e+01
[16] -1.178871e+02 -1.489993e+02 -1.755983e+02 -2.411742e+02
[1] FALSE
```

`geo` n'est pas euclidienne.

4.1 La plus empirique des méthodes consiste à ne garder que les valeurs propres positives.

```
is.euclid(geo, plot=TRUE)
```

```
[1] FALSE
```

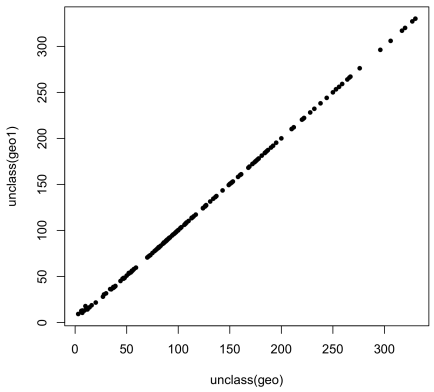


Cette matrice n'est pas euclidienne uniquement à cause de la précision de l'édition.

```
geo1 <- quasieuclid(geo)
is.euclid(geo1,print=TRUE)
```

```
 [1]  1.270026e+05  6.054361e+04  2.596671e+02  2.420916e+02  1.291763e+02
 [6]  9.500950e+01  6.269167e+01  3.382556e+01  2.016325e+01  1.179628e-11
[11]  6.494348e-12  3.646085e-12  2.905322e-12  2.470958e-12 -3.573130e-13
[16] -1.323521e-12 -4.465886e-12 -9.410897e-12 -2.745745e-11
[1] TRUE
```

```
plot(unclass(geo),unclass(geo1),pch=20)
```



Les distances sont pratiquement inchangées et la distance est devenue euclidienne.

4.2 La seconde transformation est celle de Lingoes

Elle est du type :

$$\delta_{ij} = \sqrt{d_{ij}^2 + 2c}$$

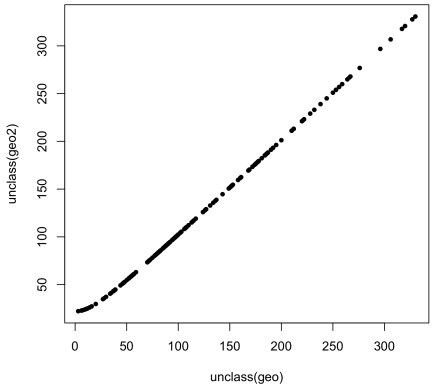
[2].
Quand c est suffisamment grande, la distance devient euclidienne. La plus petite constante qui convient est la valeur absolue de la dernière valeur propre.

```
geo2 <- lingoes(geo, print=TRUE)
```

```
Lingoes constant = 241.1742
```

```
plot(unclass(geo), unclass(geo2), pch=20)
is.euclid(geo2, print=TRUE)
```

```
 [1]  1.272438e+05  6.078478e+04  5.008413e+02  4.832658e+02  3.703505e+02
 [6]  3.361837e+02  3.038659e+02  2.749997e+02  2.613374e+02  2.279928e+02
[11]  2.194984e+02  2.117517e+02  1.930227e+02  1.618030e+02  1.232871e+02
[16]  9.217490e+01  6.557585e+01 -1.391331e-12 -1.186700e-11
[1] TRUE
```



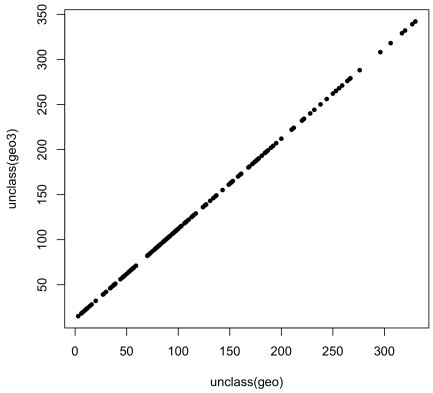
4.3 La troisième transformation est celle de Cailliez

Elle est du type :

$$\delta_{ij} = d_{ij} + c$$

[1].
Quand c est suffisamment grande, la distance devient euclidienne. La plus petite constante qui convient est la plus grande valeur propre d'une matrice non symétrique de taille $2n$.

```
 geo3 <- cailliez(geo,print=TRUE)
Cailliez constant = 12.08705
 plot(unclass(geo), unclass(geo3), pch=20)
 geo4 <- cailliez(geo2)
```



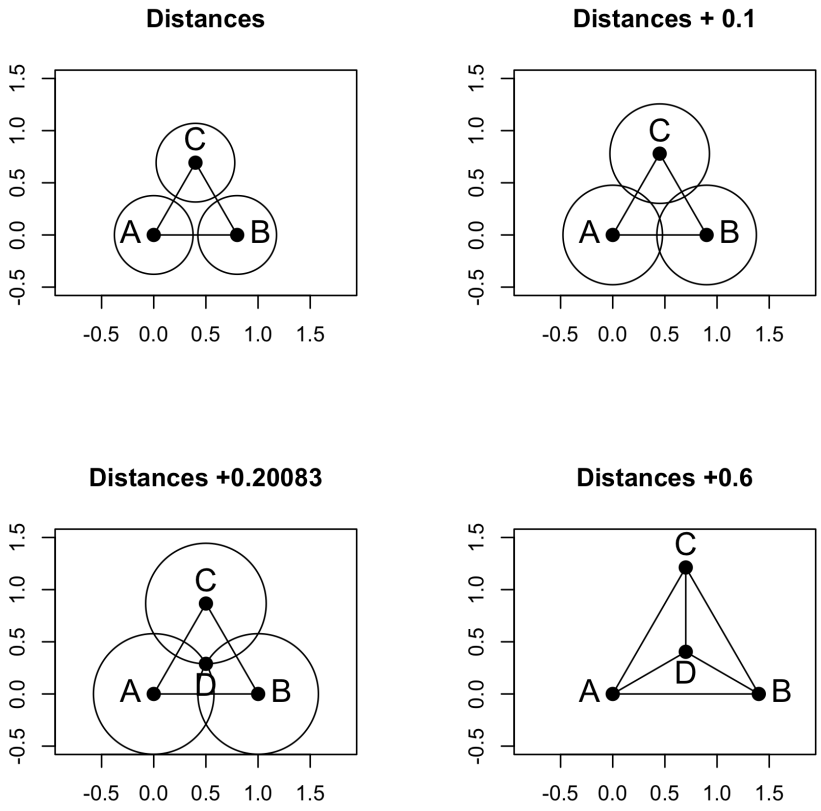
Exercice proposé par Legendre et Legendre (p 435).

1. Essayer de représenter quatre points sur un plan qui expriment la matrice de distances ci-dessous.

$$\begin{pmatrix} 0 & 0.8 & 0.8 & 0.377 \\ 0.8 & 0 & 0.8 & 0.377 \\ 0.8 & 0.8 & 0 & 0.377 \\ 0.377 & 0.377 & 0.377 & 0 \end{pmatrix}$$

2. Expliquer pourquoi la matrice n'est pas euclidienne (s'aider du graphe ci-dessous).
3. Trouver la constante de Cailliez et dessiner la représentation euclidienne.
4. Que se passe-t-il avec une constante plus grande ?

En bref, on peut toujours rendre euclidienne une matrice qui ne l'est pas.



Après avoir étudié les propriétés des distances euclidiennes, trois procédures pour rendre euclidienne une distance qui ne l'était pas, nous nous proposons de présenter ici d'autres distances associées à la nature des variables à analyser.

5 Distances associées à des données binaires

On considère deux listes d'objets sous forme de 0 et de 1 :

011000011010...
101011010111...

Ces listes peuvent être visualisées comme des tables de contingence binaires :

	1	0	total
1	a	b	$a+b$
0	c	d	$c+d$
total	$a+c$	$b+d$	n

Les quatre nombres de la table définissent une similarité entre les deux objet. TOUS les indices définis ci-dessous donnent des distances euclidiennes. Ils sont tous inférieurs ou égaux à 1 et la distance associée est définie par :

$$D_k = \sqrt{1 - S_k}$$

5.1 Distances

La fonction `dist.binary` d'ade4 propose 10 indices de similarité permettant de construire des distances.

1. Indice de Communauté de Jaccard :
$$S_1 = \frac{a}{a+b+c}$$
2. Indice de Sokal et Michener :
$$S_2 = \frac{a+d}{n}$$
3. Indice de Sokal et Sneath :
$$S_3 = \frac{a}{a+2(b+c)}$$
4. Indice de Rogers et Tanimoto :
$$S_4 = \frac{a+d}{a+2(b+c)+d}$$
5. Indice de Sorensen :
$$S_5 = \frac{2a}{2a+b+c}$$
6. Indice de Gower et Legendre :
$$S_6 = \frac{a-(b+c)+d}{n}$$

7. Indice de Ochiai :
$$S_7 = \frac{a}{\sqrt{(a+b)(a+c)}}$$

8. Indice de Sokal et Sneath :
$$S_8 = \frac{ad}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$$

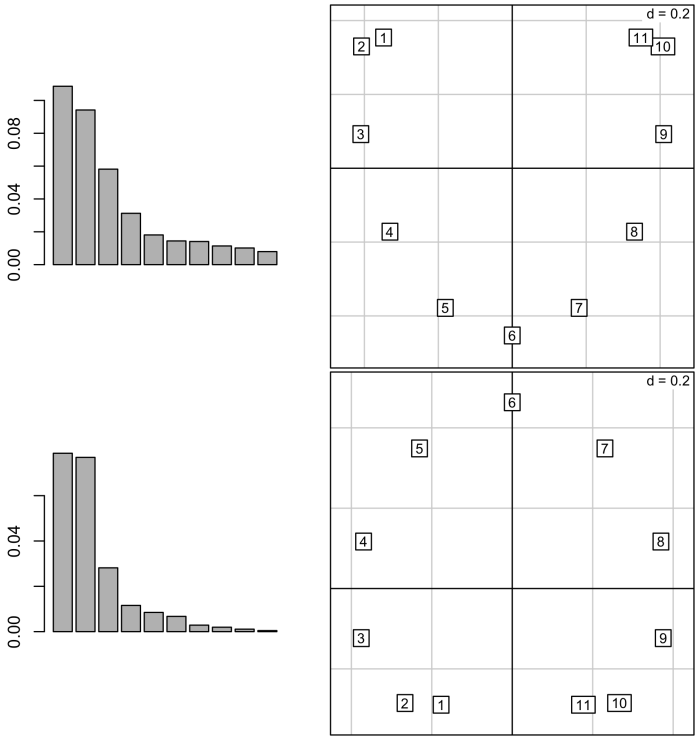
9. Phi de pearson :
$$S_9 = \frac{ad-bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$$

10. Indice spécifique :
$$S_{10} = \frac{a}{n}$$
avec l'unité si les deux objets sont identiques

5.2 Exercice

On considère un tableau de données artificielles représentant des présence-absence de 16 espèces dans 11 sites (`DK1`).
Réaliser une analyse en coordonnées principales sur ces données pour chacune des distances proposées. Représenter les sites sur les premiers plans de l'analyse. Discuter.
A titre d'exemple, on a réalisé l'étude sur les deux premières distances créées à partir de S_1 et S_2.

```
DK1 <- read.table("http://pbil.univ-lyon1.fr/R/donnees/DK1.txt")
library(ade4)
dist1 <- dist.binary(DK1,method=1)
pco1 <- dudi.pco(dist1, scannf=FALSE, nf=2)
dist2 <- dist.binary(DK1,method=2)
pco2 <- dudi.pco(dist2, scannf=FALSE, nf=2)
#
par(mfrow=c(2,2))
barplot(pco1$eig)
scatter(pco1,posieig="none")
barplot(pco2$eig)
scatter(pco2,posieig="none")
```



La procédure peut être optimisée.

6 Distances associées à des pourcentages

Un tableau contient des nombres positifs ou nuls et est considéré comme définissant des distributions de fréquences par ligne ou par colonne. On note p_i (resp q_i) une ligne du tableau contenant les proportions telles que $\sum_{i=1}^{n} p_i = 1$ ($\sum_{i=1}^{n} q_i = 1$).

p_1	p_2	$\cdots$	p_n
q_1	q_2	$\cdots$	q_n

6.1 Distances

La fonction `dist.prop` d'ade4 en propose 5.

1. Distance 1 :
$$d_1 = \sum_{i=1}^{n} |p_i - q_i|/2$$
2. Distance 2 :
$$d_2 = 1 - \sum_{i=1}^{n} p_i q_i / \left(\sqrt{\sum_{i=1}^{n} p_i^2 \sum_{i=1}^{n} q_i^2} \right)$$

3. Distance 3 de Rogers :

$$d_3 = \sum_{i=1}^{n} (p_i - q_i)^2/2$$

4. Distance 4 de Nei :

$$d_4 = -Ln\left(\frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\sqrt{\sum_{i=1}^{n} q_i^2}}\right)$$

5. Distance 5 de Edwards :

$$d_5 = \sqrt{(1 - \sum_{i=1}^{n} \sqrt{p_i q_i})}$$

6.2 Exemple

On étudie la répartition des fréquences alléliques du microsatellite INRA32 dans des populations de bovins.

```
data(microsatt)
names(microsatt)
```

```
[1] "tab"           "loci.names"    "loci.eff"      "alleles.names"
```

```
microsatt$loci.names
```

```
[1] "INRA5"  "INRA32" "INRA35" "INRA63" "INRA72" "ETH152" "ETH225" "INRA16" "INRAK"
```

```
microsatt$loci.eff
```

```
[1]  8 15 11 10 17 10 14 15 12
```

```
inra32 <- microsatt$tab[microsatt$loci.eff[1]+1:microsatt$loci.eff[2]]
```

1. Utiliser une des deux distances génétiques associées à l'étude d'un microsatellite (4. Nei ou 5. Edwards) pour construire la matrice des distances entre populations de bovins.
2. Réaliser une analyse en coordonnées principales.
3. A l'aide des informations contenues dans la fiche `pps055.pdf` que l'on trouve sur le site pédagogique `pbil`, interpréter les résultats.

7 Distances associées à des données quantitatives

On considère un tableau $\mathbf{X} = [x_{ij}]$ contenant p variables quantitatives mesurées sur n individus.

7.1 Distances

La fonction `dist.quant` d'ade4 en propose 3 dont la distance canonique. les distances sont de la forme :

$$\sqrt{(\mathbf{x}-\mathbf{y})^T}\mathbf{A}(\mathbf{x}-\mathbf{y})$$

1. Distance canonique :
$$d_{ij} = \sum_{k=1}^{p}(x_{ik} - x_{jk})^2$$
Dans ce cas, $\mathbf{A} = \mathbf{I}$

2. Distance de Joreskog :
$$\mathbf{A} = 1/diag(\mathbf{C}) \text{ où } \mathbf{C} \text{ est la matrice de variances-covariances}$$

3. Distance de Mahalanobis :
$$\mathbf{A} = inv(\mathbf{C})$$

7.2 Exemple

On étudie 8 variables morphologiques chez 129 oiseaux provenant de Bourgogne, Provence, Californie et Chili.

```
data(ecomor)
names(ecomor)
```

```
[1] "forsub"  "diet"    "habitat" "morpho"  "taxo"    "labels"  "categ"
```

```
morpho <- ecomor$morpho
```

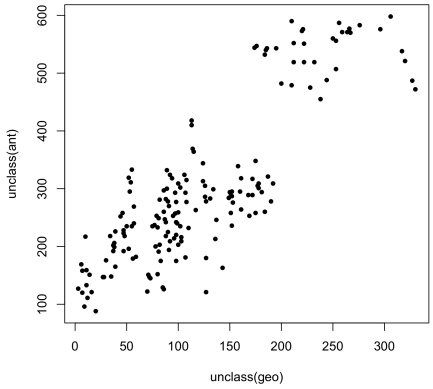
1. Utiliser une des distances associées à des variables quantitatives pour construire la matrice des distances entre la morphologie des oiseaux.
2. Réaliser une analyse en coordonnées principales.
3. A l'aide des informations contenues dans la fiche `pps023.pdf` que l'on trouve sur le site pédagogique `pbil`, interpréter les résultats.

8 Corrélation entre matrices de distances

La demi matrice de distances inférieure est traitée comme une variable.

```
ant <- as.dist(yanomama$ant)
plot(unclass(geo),unclass(ant), pch=20)
cor(unclass(geo),unclass(ant))
```

```
[1] 0.8428053
```



La signification statistique se mesure par le test de Mantel :

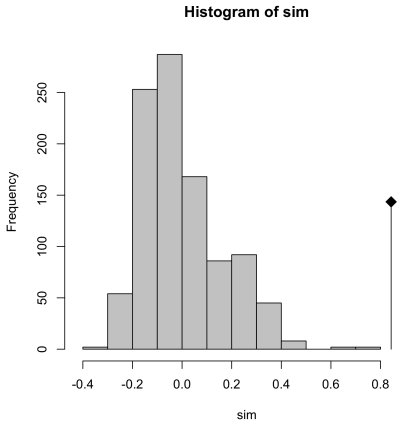
```
extest <- mantel.randtest(geo,ant,999)
extest
```

```
Monte-Carlo test
Call: mantel.randtest(m1 = geo, m2 = ant, nrepet = 999)
Observation: 0.8428053

Based on 999 replicates
Simulated p-value: 0.001
Alternative hypothesis: greater

     Std.Obs  Expectation     Variance
5.253945895 -0.001675966  0.025835022
```

```
class(extest)
```

```
[1] "randtest"
```

```
plot(extest)
```



Références

[1] F. Cailliez. The analytical solution of the additive constant problem. *Psychometrika*, 48 :305–310, 1983.

[2] J.C. Lingoes. Somme boundary conditions for a monotone analysis of symmetric matrices. *Psychometrika*, 36 :195–203, 1971.

[3] B.F.J. Manly. Randomization and regression methods for testing for associations with geographical environmental and biological distances between populations. *Researches on Population Ecology*, 28 :201–218, 1986.

[4] R.S. Spielman. Differences among yanomama indian villages : do the patterns of allele frequencies, anthropometrics and map locations correspond ? *American Journal of Physical Anthropology*, 39 :461–480, 1973.