TD4

Colin Coërchon

2023-11-20

Contents

Exercice 1

```
X0<-read.table(text="
math scie fran lati d-m
jean 6.0 6.0 5.0 5.5 8.0
aline 8.0 8.0 8.0 8.0 9.0
annie 6.0 7.0 11.0 9.5 11.0
monique 14.5 14.5 15.5 15.0 8.0
didier 14.0 14.0 12.0 12.5 10.0
andr´e 11.0 10.0 5.5 7.0 13.0
pierre 5.5 7.0 14.0 11.5 10.0
brigitte 13.0 12.5 8.5 9.5 12.0
evelyne 9.0 9.5 12.5 12.0 18.0
")
```

```
X<- scale(X0,center=TRUE,scale = FALSE)
X
```

```
##               math       scie       fran       lati d.m
## jean     -3.6666667 -3.8333333 -5.2222222 -4.5555556  -3
## aline    -1.6666667 -1.8333333 -2.2222222 -2.0555556  -2
## annie    -3.6666667 -2.8333333  0.7777778 -0.5555556   0
## monique   4.8333333  4.6666667  5.2777778  4.9444444  -3
## didier    4.3333333  4.1666667  1.7777778  2.4444444  -1
## andr´e    1.3333333  0.1666667 -4.7222222 -3.0555556   2
## pierre   -4.1666667 -2.8333333  3.7777778  1.4444444  -1
## brigitte  3.3333333  2.6666667 -1.7222222 -0.5555556   1
## evelyne  -0.6666667 -0.3333333  2.2777778  1.9444444   7
## attr(,"scaled:center")
##      math      scie      fran      lati       d.m
##  9.666667  9.833333 10.222222 10.055556 11.000000
```

```
n<-nrow(X)
p<-ncol(X)
S<-var(X)*(n-1)/n

U<-eigen(S)$vectors ; Lambda<-eigen(S)$values ; C = X%*%U

Lambda
```

```
## [1] 28.253249801 12.074723274  8.615733579  0.021732182  0.009869805
```

Significativité des différentes valeurs propres :

```
cumsum(Lambda/sum(Lambda))
```

```
## [1] 0.5768876 0.8234348 0.9993547 0.9997985 1.0000000
```

On remarque qu'on a clairement 3 valeurs prépondérantes qui correspondent à 99,9% de la somme totale.

Matrice de rotation

```
rot_map <- sqrt(Lambda) * U
rot_map
```

```
##              [,1]       [,2]         [,3]         [,4]         [,5]
## [1,] -2.734950189  3.0135494 -0.272801593 -1.534807732  3.043314229
## [2,] -1.761719352  1.2926370 -0.050222155  1.921799481 -1.898505282
## [3,] -1.445188052 -1.9089548  0.317200445  1.155714453  1.202815235
## [4,] -0.071443132 -0.0476513  0.003323298 -0.099388800 -0.066844869
## [5,] -0.003042775 -0.0112156 -0.098597506  0.003421169  0.001258566
```

Tout ce qui est corrélation

On calcule maintenant la corrélation. C'est-à-dire la contribution relative des axes aux individus :

```
COR<- C^2 / rowSums(X^2)
COR
```

```
##                [,1]         [,2]        [,3]         [,4]         [,5]
## jean     0.88894548 0.0340427538 0.076587880 2.622775e-04 1.616070e-04
## aline    0.80365559 0.0260088320 0.169811511 4.260371e-04 9.802799e-05
## annie    0.46012197 0.5344933508 0.004038159 1.329907e-03 1.660880e-05
## monique  0.89414954 0.0004372797 0.105035734 2.827126e-04 9.472899e-05
## didier   0.87726184 0.1020725256 0.019794006 1.079034e-04 7.637245e-04
## andr´e   0.23623016 0.5777346580 0.185435015 1.439185e-05 5.857764e-04
## pierre   0.02583884 0.9071461473 0.066661161 3.109728e-04 4.287757e-05
## brigitte 0.17435766 0.7357964730 0.087211720 2.629342e-03 4.805383e-06
## evelyne  0.05330060 0.1977673369 0.748407627 4.645728e-04 5.986834e-05
```

Et maintenant, la contribution relative des individus aux axes :

```
CTR<- 1/n* C^2 / matrix(eigen(S)$values,n,p,byrow = TRUE)
CTR
```

```
##                [,1]         [,2]        [,3]        [,4]        [,5]
## jean     0.297726930 0.0266783459 0.084116112 0.114200796 0.154939841
## aline    0.061005911 0.0046197026 0.042271263 0.042045012 0.021301584
## annie    0.040507433 0.1101020178 0.001165793 0.152211729 0.004185620
## monique  0.374291227 0.0004283019 0.144182553 0.153854355 0.113512117
## didier   0.159647400 0.0434643006 0.011812517 0.025528947 0.397858978
## andr´e   0.034783719 0.1990491457 0.089538242 0.002755005 0.246906420
## pierre   0.004343771 0.3568307951 0.036748775 0.067964365 0.020634020
## brigitte 0.015425990 0.1523214081 0.025302476 0.302429457 0.001217026
## evelyne  0.012267619 0.1065059822 0.564862269 0.139010335 0.039444393
```

Correction du prof

```r
mon_ACP <- function(X){
  n = nrow(X)
  X <- scale(X, scale = FALSE)  # On peut mettre TRUE aussi paraît-il
  S <- cov(X)  # On peut aussi faire t(X) %*% X / n
  eigen_res <- eigen(S)
  U <- eigen_res$vectors
  Lambda <- eigen_res$values
  C <- X %*% U

  rot_map <- sqrt(Lambda) * U

}
```

Exercice 2

```
library(MASS)
```

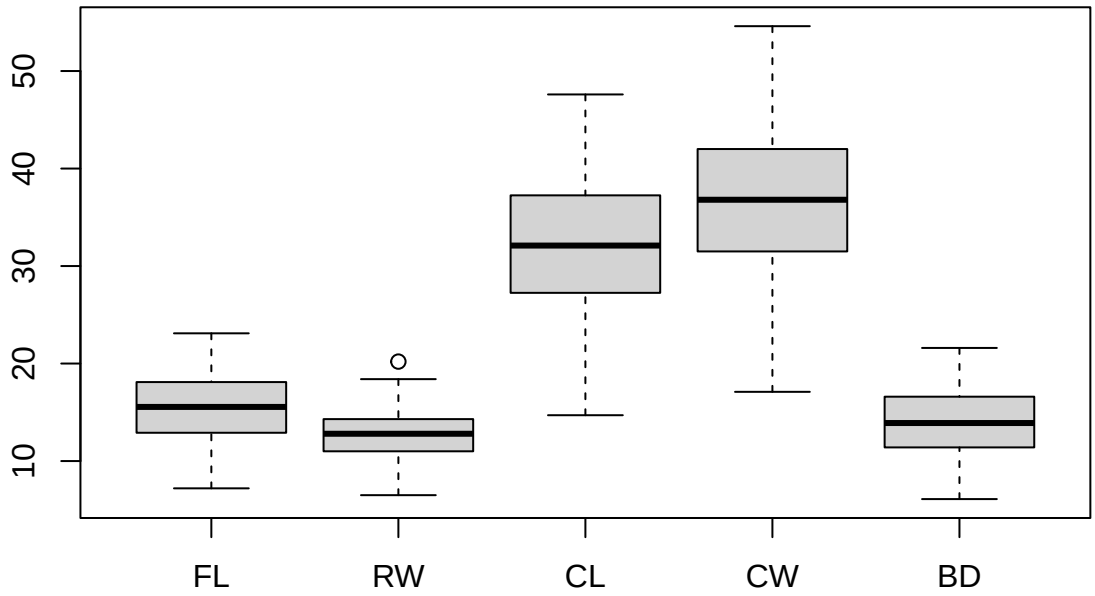
```
##
## Attachement du package : 'MASS'
```

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

```
data(crabs)
crabsquant<-crabs[,4:8]
```

```
# ACP0 <- prcomp(crabs)
# summary(ACP)
# plot(ACP0$sdev)
```

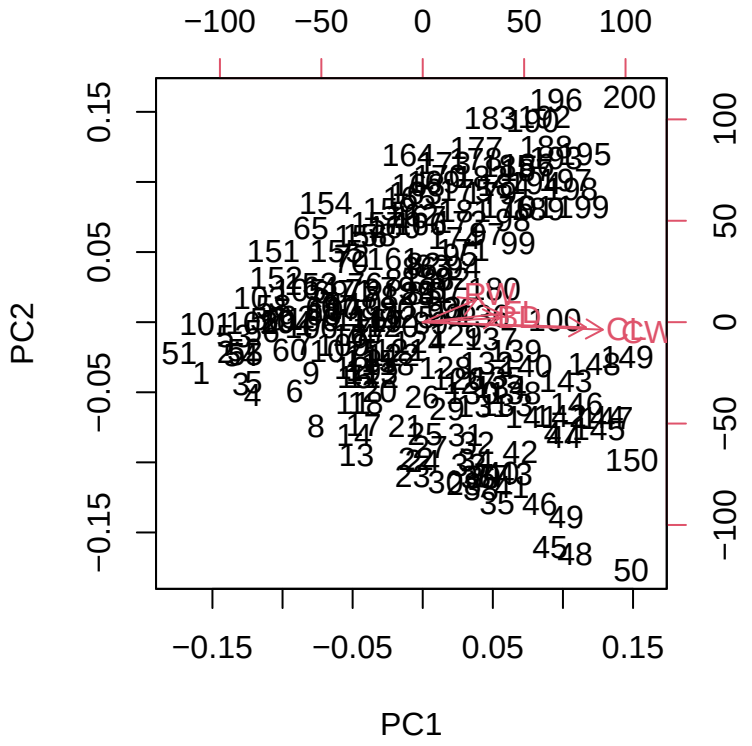
```
boxplot(crabsquant)
```



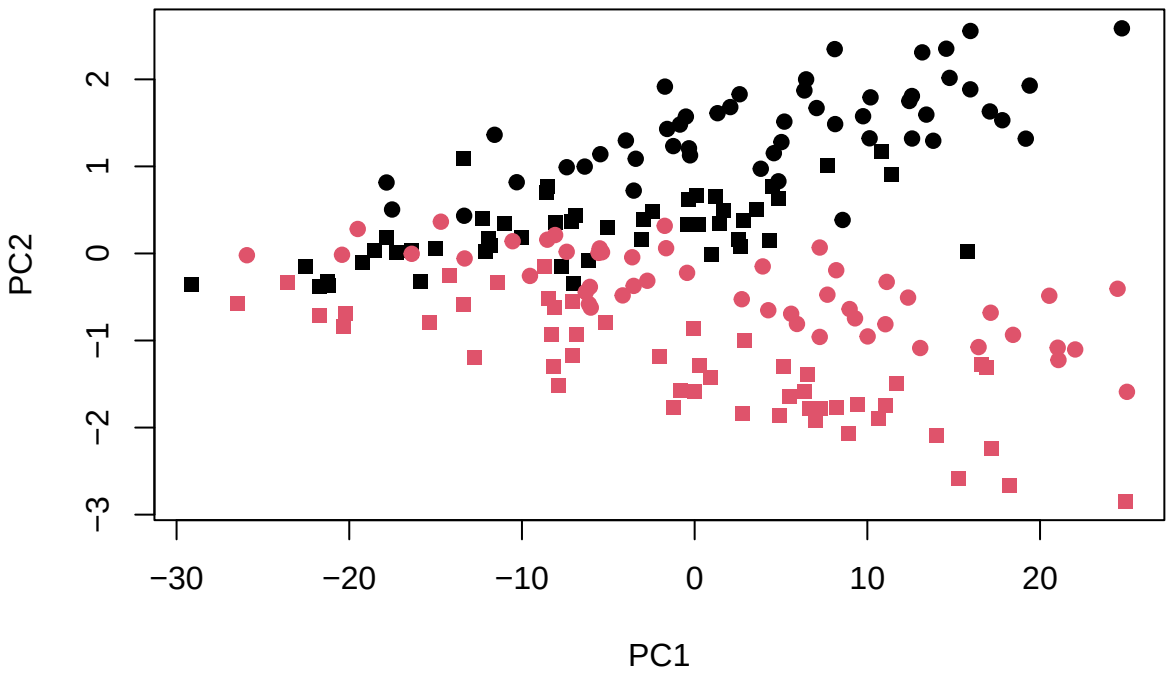
```
res <- prcomp(crabsquant)
res
```

```
## Standard deviations (1, .., p=5):
## [1] 11.8619441  1.1387874  1.0001346  0.3678306  0.2791312
##
## Rotation (n x k) = (5 x 5):
##           PC1        PC2        PC3        PC4        PC5
## FL 0.2889810  0.3232500 -0.5071698  0.7342907  0.1248816
## RW 0.1972824  0.8647159  0.4141356 -0.1483092 -0.1408623
## CL 0.5993986 -0.1982263 -0.1753299 -0.1435941 -0.7416656
## CW 0.6616550 -0.2879790  0.4913755  0.1256282  0.4712202
## BD 0.2837317  0.1598447 -0.5468821 -0.6343657  0.4386868
```

```
biplot(res)
```




```
plot(res$x, col=crabs$sex, pch = c(15,19)[crabs$sp])
```



On remarque qu'il y a qu'une seule variable vraiment significative.

On va reconstruire notre X en enlevant la principale composante 1 (trop significative) :

```
u1 <- prcomp(crabsquant)$rotation[,1, drop=FALSE]

newX <- as.matrix(crabsquant) %*% u1 %*% t(u1)

head(newX)
```

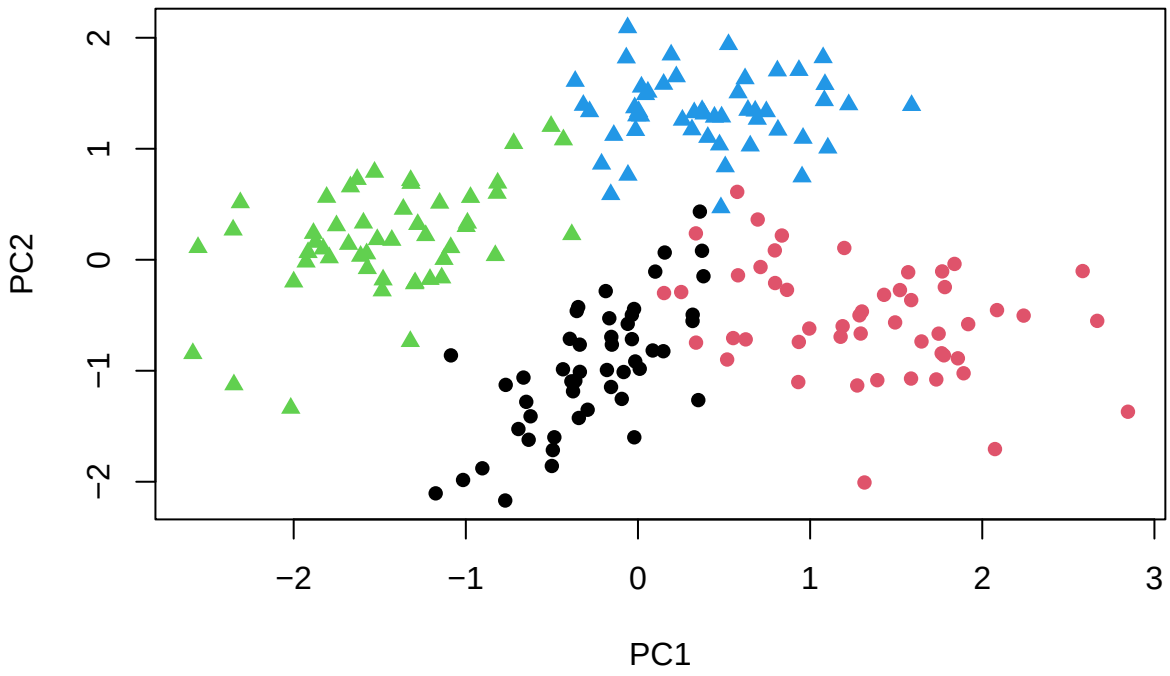
```
##          FL       RW       CL       CW        BD
## 1  8.054021 5.498343 16.70549 18.44060  7.907722
## 2  8.892886 6.071021 18.44545 20.36128  8.731350
## 3  9.418411 6.429789 19.53548 21.56453  9.247329
## 4  9.822893 6.705922 20.37445 22.49064  9.644463
## 5  9.860819 6.731813 20.45312 22.57747  9.681700
## 6 11.269428 7.693446 23.37483 25.80265 11.064723
```

```
# Soustraction correcte pour obtenir les données ajustées
adjustedData <- crabsquant - newX

# Application de l'ACP sur les données ajustées
res2 <- prcomp(adjustedData)
res2
```

```
## Standard deviations (1, .., p=5):
```

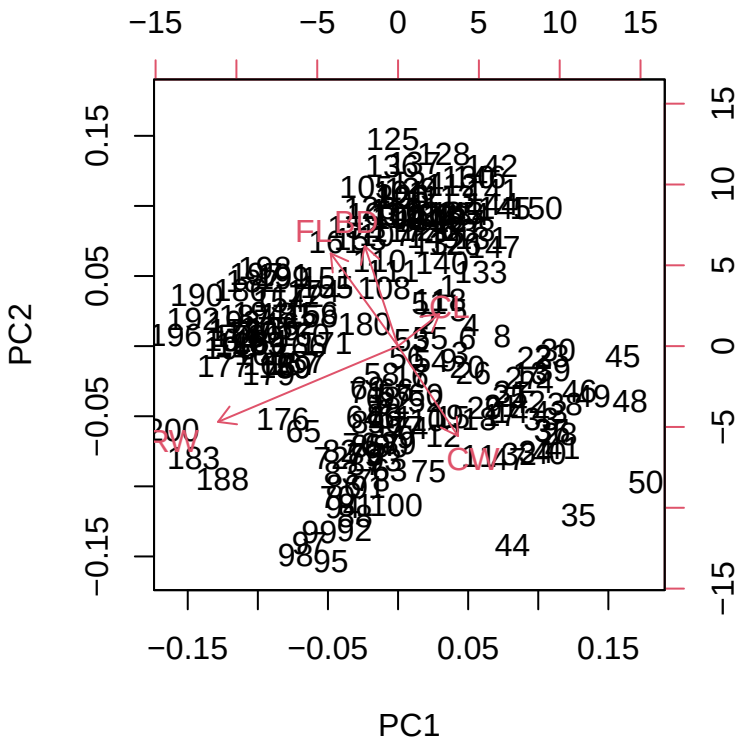
```
## [1] 1.138787e+00 1.000135e+00 3.678306e-01 2.791312e-01 4.960538e-15
##
## Rotation (n x k) = (5 x 5):
##           PC1        PC2        PC3        PC4       PC5
## FL -0.3232500  0.5071698  0.7342907 -0.1248816 0.2889810
## RW -0.8647159 -0.4141356 -0.1483092  0.1408623 0.1972824
## CL  0.1982263  0.1753299 -0.1435941  0.7416656 0.5993986
## CW  0.2879790 -0.4913755  0.1256282 -0.4712202 0.6616550
## BD -0.1598447  0.5468821 -0.6343657 -0.4386868 0.2837317
```

```
plot(res2$x, col=interaction(crabs$sex,crabs$sp), pch = c(16,17)[crabs$sp])
```



4 axes à garder j'ai l'impression

```
biplot(res2)
```



Effectivement, sans la première CP trop significative, cette nouvelle ACP est beaucoup plus intéressante !

Exercice 3

```
data <- read.table("neighbor_globin.txt", header=FALSE)
X <- as.matrix(data[, -1])
head(X)
```

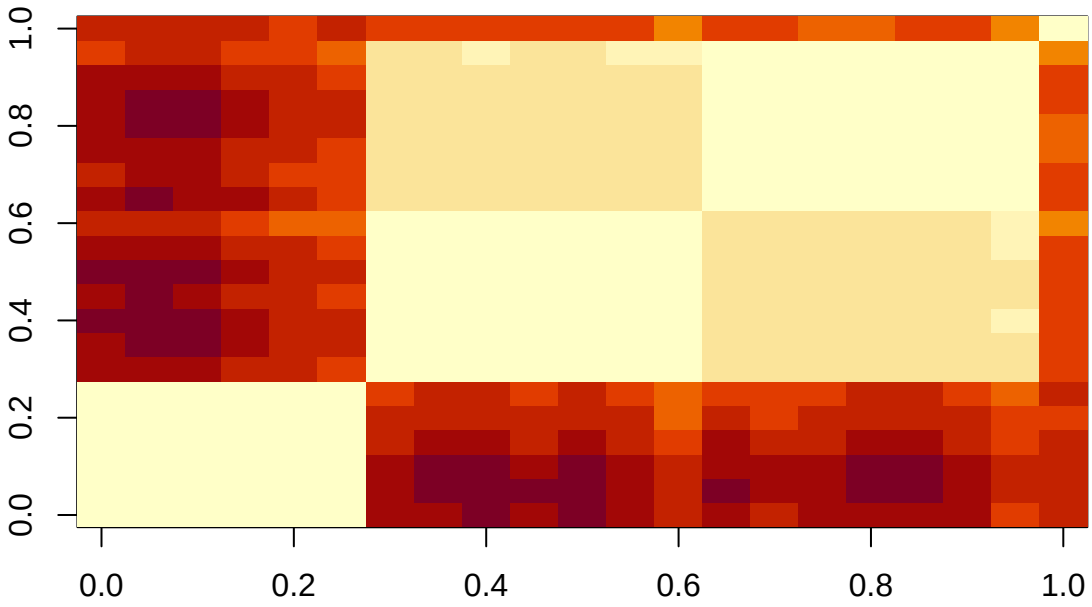
```
##          V2     V3     V4     V5     V6     V7     V8     V9    V10    V11
## [1,] 0.0000 0.1806 0.2434 0.3964 0.5656 0.4987 1.9654 2.1040 2.1278 2.0965
## [2,] 0.1806 0.0000 0.1929 0.2997 0.4852 0.4271 1.9675 2.0689 2.2427 2.1483
## [3,] 0.2434 0.1929 0.0000 0.3432 0.5312 0.4635 1.8727 2.1478 2.1478 2.1092
## [4,] 0.3964 0.2997 0.3432 0.0000 0.3657 0.3196 1.8520 2.0577 2.0649 1.8216
## [5,] 0.5656 0.4852 0.5312 0.3657 0.0000 0.2970 1.8912 2.0551 2.0572 1.7896
## [6,] 0.4987 0.4271 0.4635 0.3196 0.2970 0.0000 1.7142 1.9036 1.9751 1.6927
##         V12    V13    V14    V15    V16    V17    V18    V19    V20    V21
## [1,] 2.2725 2.0807 1.9645 1.9928 1.9195 2.0944 1.9867 1.9486 1.8515 1.9880
## [2,] 2.2753 2.0387 2.0941 2.1273 1.9495 2.0628 2.1114 1.9951 1.9200 2.0044
## [3,] 2.2318 1.9386 2.0581 2.0567 1.9920 2.1235 2.1776 2.0310 1.9519 2.0735
## [4,] 1.9345 2.0096 1.9935 2.0463 1.8520 1.9878 2.1320 1.9407 1.8823 2.0378
## [5,] 1.9478 1.9237 1.7647 1.9622 1.9429 1.9423 2.0500 1.9352 1.9823 2.0511
## [6,] 1.8907 1.8523 1.8770 1.8414 1.7849 1.8503 1.9604 1.9075 1.8643 1.7584
##         V22
## [1,] 2.6100
## [2,] 2.5663
## [3,] 2.6225
## [4,] 2.5424
## [5,] 2.3154
## [6,] 2.4536
```

```
n <- nrow(X)

sum.of.squares<- rowSums(X^2)
D2 <- cbind(sum.of.squares) %*% matrix(1,1,n) +
matrix(1,n,1) %*% rbind(sum.of.squares) -
2 * X%*%t(X)

print(D2,digits=2)
```

```
##        [,1]    [,2]     [,3]  [,4]     [,5]  [,6]  [,7]     [,8]  [,9] [,10]
##  [1,]  0.00 1.7e-01  2.4e-01  0.67  1.3e+00  1.36 41.47  4.3e+01 44.37 41.99
##  [2,]  0.17 2.8e-14  1.5e-01  0.58  1.3e+00  1.44 43.72  4.6e+01 46.83 44.16
##  [3,]  0.24 1.5e-01 -2.8e-14  0.58  1.2e+00  1.41 42.44  4.4e+01 45.47 42.89
##  [4,]  0.67 5.8e-01  5.8e-01  0.00  5.1e-01  0.54 38.88  4.1e+01 42.01 38.82
##  [5,]  1.27 1.3e+00  1.2e+00  0.51 -1.4e-14  0.46 36.34  3.8e+01 39.31 36.07
##  [6,]  1.36 1.4e+00  1.4e+00  0.54  4.6e-01  0.00 33.86  3.6e+01 37.06 33.91
##  [7,] 41.47 4.4e+01  4.2e+01 38.88  3.6e+01 33.86  0.00  4.2e-01  0.59  0.77
##  [8,] 43.44 4.6e+01  4.4e+01 41.00  3.8e+01 36.07  0.42 -1.4e-14  0.38  0.85
##  [9,] 44.37 4.7e+01  4.5e+01 42.01  3.9e+01 37.06  0.59  3.8e-01  0.00  0.92
## [10,] 41.99 4.4e+01  4.3e+01 38.82  3.6e+01 33.91  0.77  8.5e-01  0.92  0.00
## [11,] 45.71 4.8e+01  4.7e+01 42.55  4.0e+01 37.58  1.07  9.7e-01  0.84  0.28
## [12,] 41.66 4.4e+01  4.2e+01 39.07  3.6e+01 34.16  0.84  1.0e+00  0.94  0.49
## [13,] 36.38 3.9e+01  3.8e+01 34.53  3.1e+01 30.02  1.42  1.3e+00  1.57  1.91
## [14,] 41.69 4.4e+01  4.4e+01 40.82  3.8e+01 35.68  9.37  1.0e+01  9.18  9.99
```

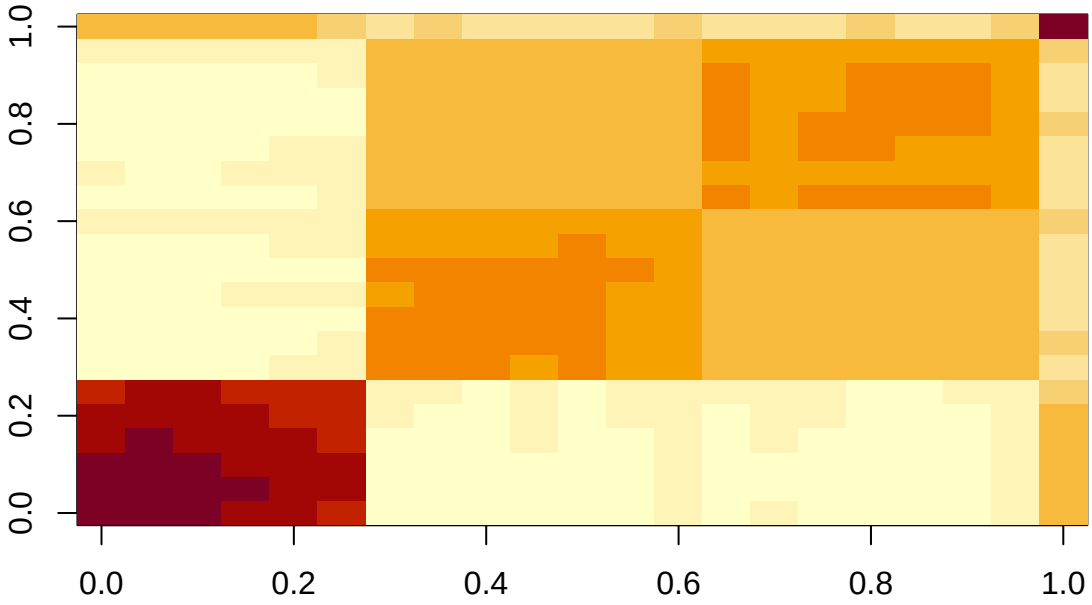
```
## [15,] 38.90 4.1e+01  4.1e+01 37.82  3.6e+01 32.98  8.76  9.5e+00  8.89  9.44
## [16,] 39.96 4.2e+01  4.2e+01 38.88  3.7e+01 34.05  9.80  1.0e+01  9.56 10.01
## [17,] 43.11 4.6e+01  4.6e+01 42.45  4.0e+01 37.41 10.03  1.1e+01  9.64 10.47
## [18,] 42.17 4.5e+01  4.4e+01 41.23  3.9e+01 36.27  9.71  1.1e+01  9.66 10.32
## [19,] 40.78 4.3e+01  4.3e+01 39.87  3.8e+01 35.03  9.35  1.0e+01  9.40  9.91
## [20,] 35.84 3.8e+01  3.8e+01 35.18  3.3e+01 30.32  8.23  8.6e+00  7.72  8.83
## [21,] 36.70 3.8e+01  3.7e+01 37.05  3.4e+01 35.96 35.06  3.2e+01 32.73 33.66
##         [,11] [,12]   [,13]   [,14] [,15]    [,16]   [,17]    [,18]    [,19]
##  [1,] 4.6e+01 41.66 3.6e+01 4.2e+01 38.90  4.0e+01 4.3e+01  4.2e+01  4.1e+01
##  [2,] 4.8e+01 43.80 3.9e+01 4.4e+01 41.39  4.2e+01 4.6e+01  4.5e+01  4.3e+01
##  [3,] 4.7e+01 42.48 3.8e+01 4.4e+01 41.02  4.2e+01 4.6e+01  4.4e+01  4.3e+01
##  [4,] 4.3e+01 39.07 3.5e+01 4.1e+01 37.82  3.9e+01 4.2e+01  4.1e+01  4.0e+01
##  [5,] 4.0e+01 36.34 3.1e+01 3.8e+01 35.72  3.7e+01 4.0e+01  3.9e+01  3.8e+01
##  [6,] 3.8e+01 34.16 3.0e+01 3.6e+01 32.98  3.4e+01 3.7e+01  3.6e+01  3.5e+01
##  [7,] 1.1e+00  0.84 1.4e+00 9.4e+00  8.76  9.8e+00 1.0e+01  9.7e+00  9.4e+00
##  [8,] 9.7e-01  1.01 1.3e+00 1.0e+01  9.52  1.0e+01 1.1e+01  1.1e+01  1.0e+01
##  [9,] 8.4e-01  0.94 1.6e+00 9.2e+00  8.89  9.6e+00 9.6e+00  9.7e+00  9.4e+00
## [10,] 2.8e-01  0.49 1.9e+00 1.0e+01  9.44  1.0e+01 1.0e+01  1.0e+01  9.9e+00
## [11,] 1.4e-14  0.41 2.5e+00 9.7e+00  9.37  9.7e+00 9.8e+00  9.7e+00  9.5e+00
## [12,] 4.1e-01  0.00 2.1e+00 8.7e+00  8.36  8.9e+00 9.0e+00  8.8e+00  8.6e+00
## [13,] 2.5e+00  2.10 1.4e-14 8.7e+00  8.28  8.8e+00 9.3e+00  9.6e+00  9.2e+00
## [14,] 9.7e+00  8.75 8.7e+00 1.4e-14  0.32  4.1e-01 7.6e-01  6.7e-01  6.7e-01
## [15,] 9.4e+00  8.36 8.3e+00 3.2e-01  0.00  4.7e-01 1.1e+00  8.4e-01  8.0e-01
## [16,] 9.7e+00  8.94 8.8e+00 4.1e-01  0.47 -1.4e-14 1.1e+00  1.0e+00  9.4e-01
## [17,] 9.8e+00  9.00 9.3e+00 7.6e-01  1.09  1.1e+00 1.4e-14  1.9e-01  3.4e-01
## [18,] 9.7e+00  8.82 9.6e+00 6.7e-01  0.84  1.0e+00 1.9e-01 -1.4e-14  1.1e-01
## [19,] 9.5e+00  8.56 9.2e+00 6.7e-01  0.80  9.4e-01 3.4e-01  1.1e-01 -1.4e-14
## [20,] 8.6e+00  7.54 7.2e+00 1.6e+00  1.66  1.7e+00 1.8e+00  1.9e+00  2.0e+00
## [21,] 3.3e+01 32.05 2.7e+01 3.3e+01 32.73  3.2e+01 3.1e+01  3.3e+01  3.4e+01
##         [,20] [,21]
##  [1,] 3.6e+01    37
##  [2,] 3.8e+01    38
##  [3,] 3.8e+01    37
##  [4,] 3.5e+01    37
##  [5,] 3.3e+01    34
##  [6,] 3.0e+01    36
##  [7,] 8.2e+00    35
##  [8,] 8.6e+00    32
##  [9,] 7.7e+00    33
## [10,] 8.8e+00    34
## [11,] 8.6e+00    33
## [12,] 7.5e+00    32
## [13,] 7.2e+00    27
## [14,] 1.6e+00    33
## [15,] 1.7e+00    33
## [16,] 1.7e+00    32
## [17,] 1.8e+00    31
## [18,] 1.9e+00    33
## [19,] 2.0e+00    34
## [20,] 1.4e-14    28
## [21,] 2.8e+01     0
```

```
image(D2)
```




```
J <- diag(n) - matrix(1/n, n, n)
```

```
B <- -(1/2) * J %*% D2 %*% J
image(B)
```



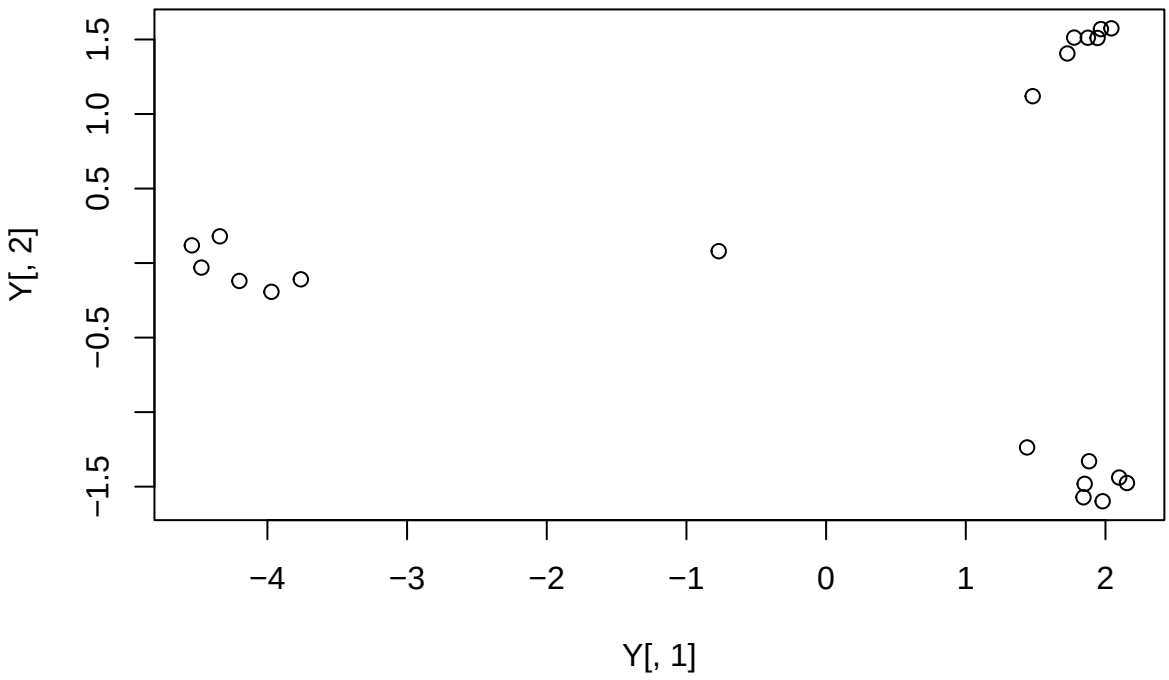
B correspond à la matrice XX^T lorsque X est centrée.

```
B_diag <- eigen(B)
U <- B_diag$vectors
Lambda <- diag(B_diag$values)
```

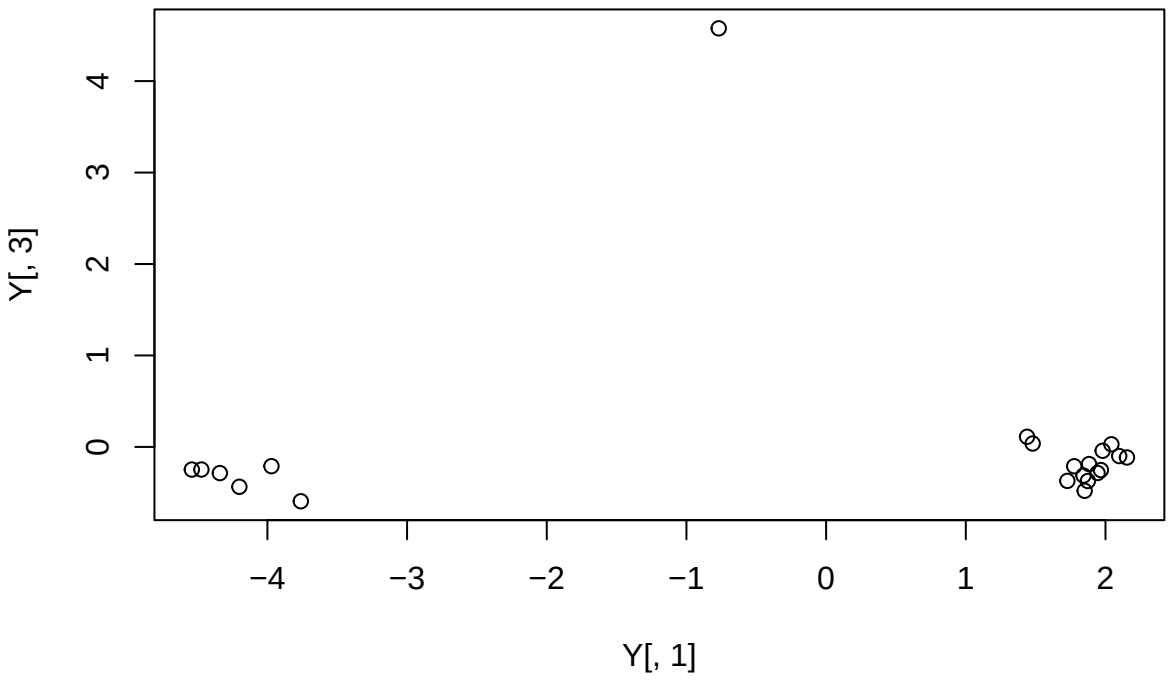
```
barplot(B_diag$values)
```

3 valeurs propres significatives.

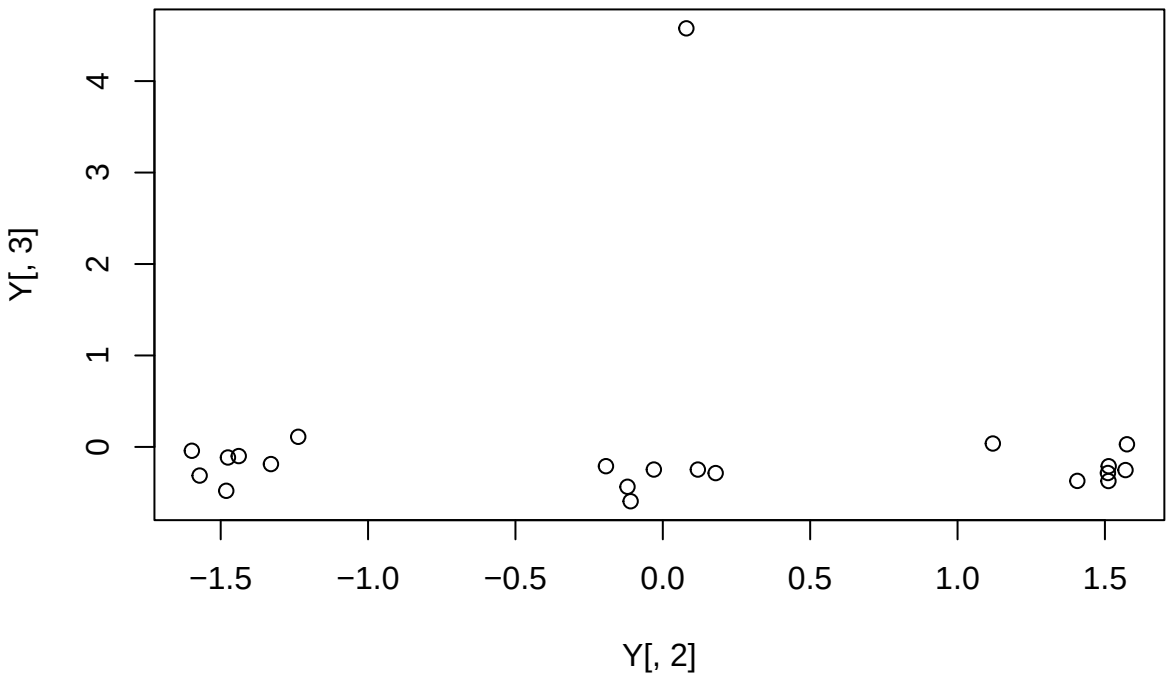
```
U_3 <- U[, 1:3] # les trois premiers vecteurs propres
Lambda_3 <- Lambda[1:3, 1:3] # les trois premières valeurs propres

Y <- U_3 %*% sqrt(Lambda_3)

plot(Y[,1], Y[,2])
```




```
plot(Y[,1], Y[,3])
```




```
plot(Y[,2], Y[,3])
```


Y[, 3]
0
1
2
3
4
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Y[, 2]