

Analyse en composantes principales - Eaux minérales

Sarah Ouadah

Module DYST, M2 BEE, Septembre 2023

```
knitr::opts_chunk$set(echo = TRUE)
library(FactoMineR)
library(ggplot2)
library(factoextra) #pour extraire et visualiser les résultats issus de FactoMineR

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

## corrplot 0.88 loaded
library(gridExtra)
```

Données

On souhaite créer une typologie des eaux pour regrouper des eaux ayant des caractéristiques communes. On a pour cela noté la teneur en HCO₃, SO₄, Cl, Ca, Mg, Na de 21 bouteilles d'eau minérale. La table de données se trouve dans le fichier `eaux.csv`. On utilisera le package `FactoMineR`.

Extrait des données récoltées :

```
Table <- read.table("eaux.csv",header=TRUE, sep=";", row.names=7)
head(Table)
```

```
##           HC03 S04 CL   CA MG  NA.
## Aix           341  27  3 84.0 23  2.0
## Beckerish     263  23  9 91.0  5  3.0
## Cayranne      287   3  5 44.0 24 23.0
## Chambon       298   9 23 96.0  6 11.0
## Cristal       200  15  8 70.0  2  4.0
## Cristaline    228  11 15 67.5  7  8.4
```

```
str(Table)
```

```
## 'data.frame':    21 obs. of  6 variables:
## $ HC03: int  341 263 287 298 200 228 357 311 186 183 ...
## $ S04 : int  27 23 3 9 15 11 10 14 10 16 ...
## $ CL  : int  3 9 5 23 8 15 2 18 16 44 ...
## $ CA  : num  84 91 44 96 70 67.5 78 73 64 48 ...
## $ MG  : int  23 5 24 6 2 7 24 18 4 11 ...
## $ NA. : num  2 3 23 11 4 8.4 5 13 9 31 ...
```

Statistiques descriptives

On commence par regarder des statistiques simples sur les différentes variables :

```
m.Table=colMeans(Table)
sd.Table=apply(Table, 2, sd)
print(m.Table)
```

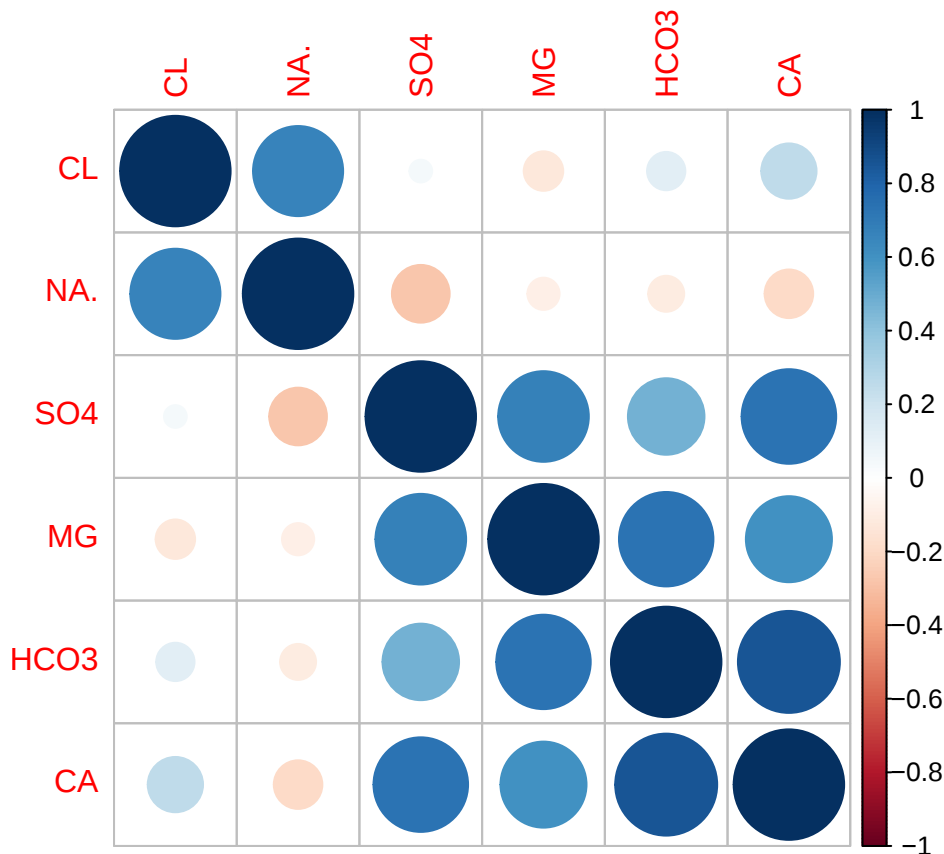
```
##      HCO3      SO4      CL      CA      MG      NA.
## 249.38095 40.90476 13.71429 77.02381 11.61905 10.01905
```

```
head(cor(Table))
```

```
##      HCO3      SO4      CL      CA      MG      NA.
## HCO3  1.0000000  0.4798519  0.1201849  0.8520840  0.73099594 -0.10600737
## SO4   0.4798519  1.0000000  0.0421654  0.7330726  0.67349251 -0.27261905
## CL    0.1201849  0.0421654  1.0000000  0.2503334 -0.12759491  0.66535937
## CA    0.8520840  0.7330726  0.2503334  1.0000000  0.60667932 -0.19330280
## MG    0.7309959  0.6734925 -0.1275949  0.6066793  1.00000000 -0.08493976
## NA.   -0.1060074 -0.2726190  0.6653594 -0.1933028 -0.08493976  1.00000000
```

```
correlation=cor(Table)
```

```
# h=heatmap(abs(correlation),symm=T)
corrplot(correlation[h$rowInd,h$colInd])
```



ACP

L'ACP est une méthode pour visualiser en deux dimensions, avec un minimum de déformation, un nuage d'individus vivant dans une dimension supérieure, ici 6.

```
library(FactoMineR)
acp.res=PCA(Table, scale.unit = TRUE, graph=F, ncp = 6)
names(acp.res)
```

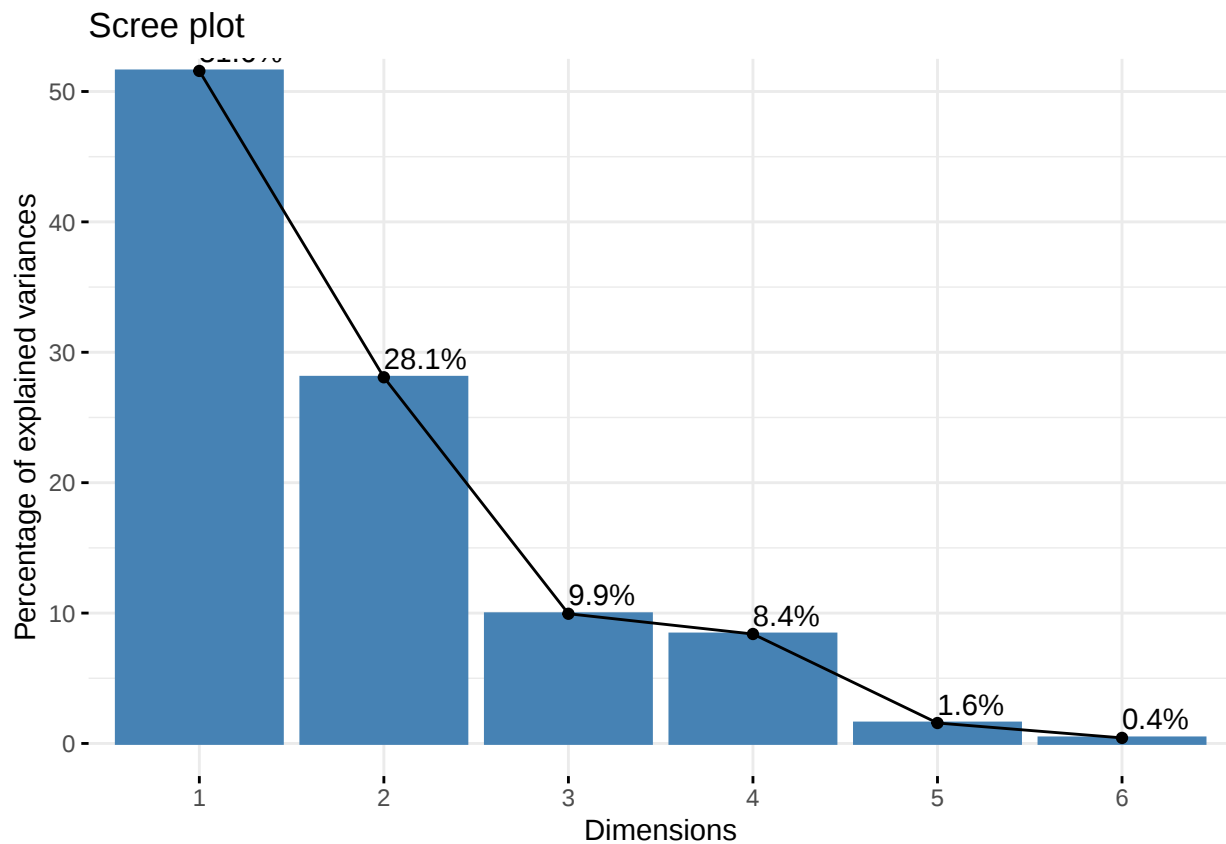
```
## [1] "eig" "var" "ind" "svd" "call"
```

On obtient la liste des valeurs propres en affichant la partie de l'objet `acp.res` correspondante :

```
acp.res$eig
```

```
##          eigenvalue percentage of variance cumulative percentage of variance
## comp 1 3.09465248          51.5775414          51.57754
## comp 2 1.68496073          28.0826788          79.66022
## comp 3 0.59686615           9.9477692          89.60799
## comp 4 0.50368595           8.3947658          98.00276
## comp 5 0.09414994           1.5691656          99.57192
## comp 6 0.02568475           0.4280792         100.00000
```

```
fviz_eig(acp.res, addlabels = TRUE, ylim = c(0,50))
```



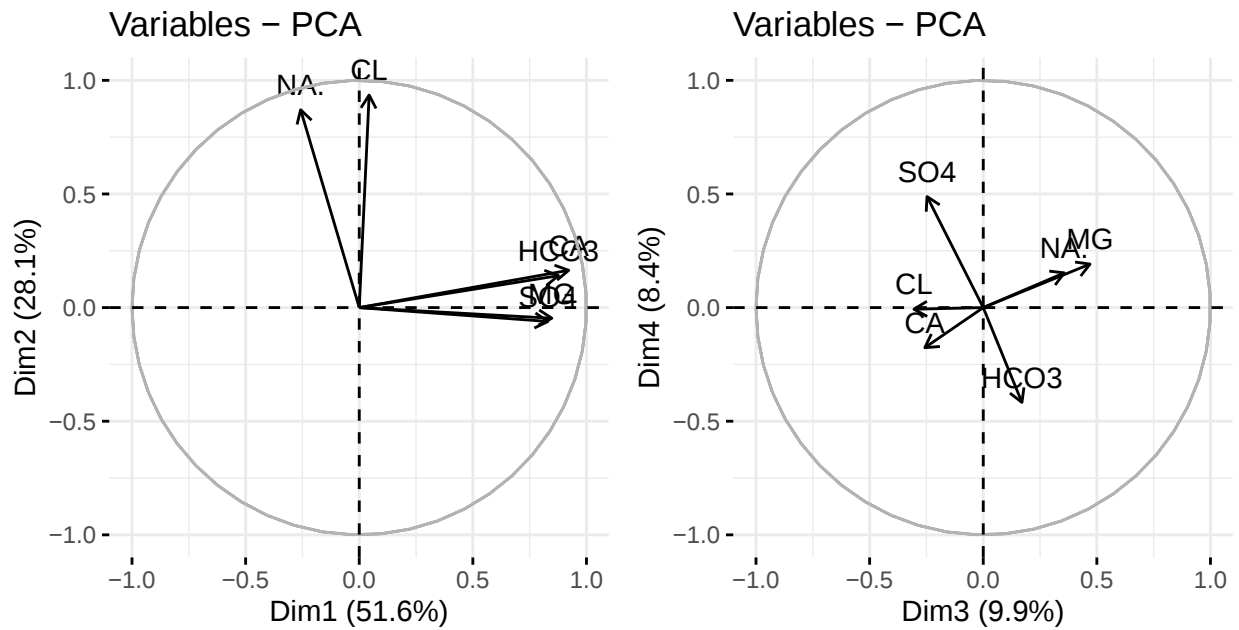
Les deux premiers axes expliquent quasiment 80% de l'inertie.

Le cercle des corrélations permet d'analyser les relations entre les composantes principales et les anciennes variables. On obtient son graphique ainsi que les informations associées grâce aux commandes suivantes :

```
# plot(acp.res, choix="var")
names(acp.res$var)
```

```
## [1] "coord" "cor" "cos2" "contrib"
```

```
p1=fviz_pca_var(acp.res, axes = 1:2)
p2=fviz_pca_var(acp.res, axes = 3:4)
grid.arrange(p1,p2,nrow=1)
```



La projection des individus dans le premier plan, constitué des deux premiers axes principaux, ainsi que les informations associées sont obtenues grâce aux commandes suivantes :

```
# plot(acp.res, choix="ind")
names(acp.res$ind)

## [1] "coord" "cos2" "contrib" "dist"

p1=fviz_pca_ind(acp.res, axes = 1:2,col.ind="cos2")
p1
```

