Devoir_5

EL_Hadrami

27/12/2020

```r
library("FactoMineR")
library("factoextra")
```

```
## Loading required package: ggplot2
```

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

```r
library("corrplot")
```

```
## corrplot 0.84 loaded
```

```r
library("ca")
library("readxl")
```

Chargement du jeux de données

```r
load("data/chiens.rda")
# chiendata <- read_xlsx("data/chiens.xlsx",col_names = FALSE)
# chiendata <- as.matrix(chiendata)
# chiendata[,1]
# chiendataextr <- as.matrix(chiendata[,2:8])
# rownames(chiendataextr) <- chiendata[,1]
# colnames(chiendataextr) <- c("taille","Poids","Veloc","Intell","Affec","Agress","Role")
# chiendataextr <- as.data.frame(chiendataextr)
head(chiens,4)
```

```
##            taille poids velocite intellig affect agress  fonction
## beauceron     T++    P+      V++       I+    Af+    Ag+   Utilite
## basset         T-    P-       V-       I-    Af-    Ag+    Chasse
## ber_allem     T++    P+      V++      I++    Af+    Ag+   Utilite
## boxer          T+    P+       V+       I+    Af+    Ag+ Compagnie
```

```r
class(chiens)
```

```
## [1] "data.frame"
```

Realisation d'une ACM

```r
H <- chiens[,1:6]
# tableau disjonctif
tabd <- tab.disjonctif(H)
tabd <- as.matrix(tabd)
f <- tabd / sum(tabd)
r <- apply(f,1,sum)
c <- apply(f,2,sum)
# matrice de Z
Z <- diag(1/r)%*%(f-r%*%t(c))%*%diag(1/c)
```

```r
source("GSVD.R")
U <- gsvd(Z,r,c)$U
V <- gsvd(Z,r,c)$V
d <- gsvd(Z,r,c)$d
```

3.b

```r
# inertie totale
it <- sum(d^2)
it
```

```
## [1] 1.666667
```

```r
m <- ncol(tabd)
p <- ncol(H)
(m / p) - 1
```

```
## [1] 1.666667
```

Ce qui montre que l'inertie total vaut $\frac{m}{p} - 1$ avec m le nombre de modalité et p le nombre de variable qualitative

3.c

```r
d # les valeurs propres sur chaque dimension
```

```
##  [1] 0.69397850 0.62027195 0.45929734 0.39693076 0.38746957 0.35113432
##  [7] 0.28541629 0.21370484 0.15343373 0.08782388
```

```r
length(d)
```

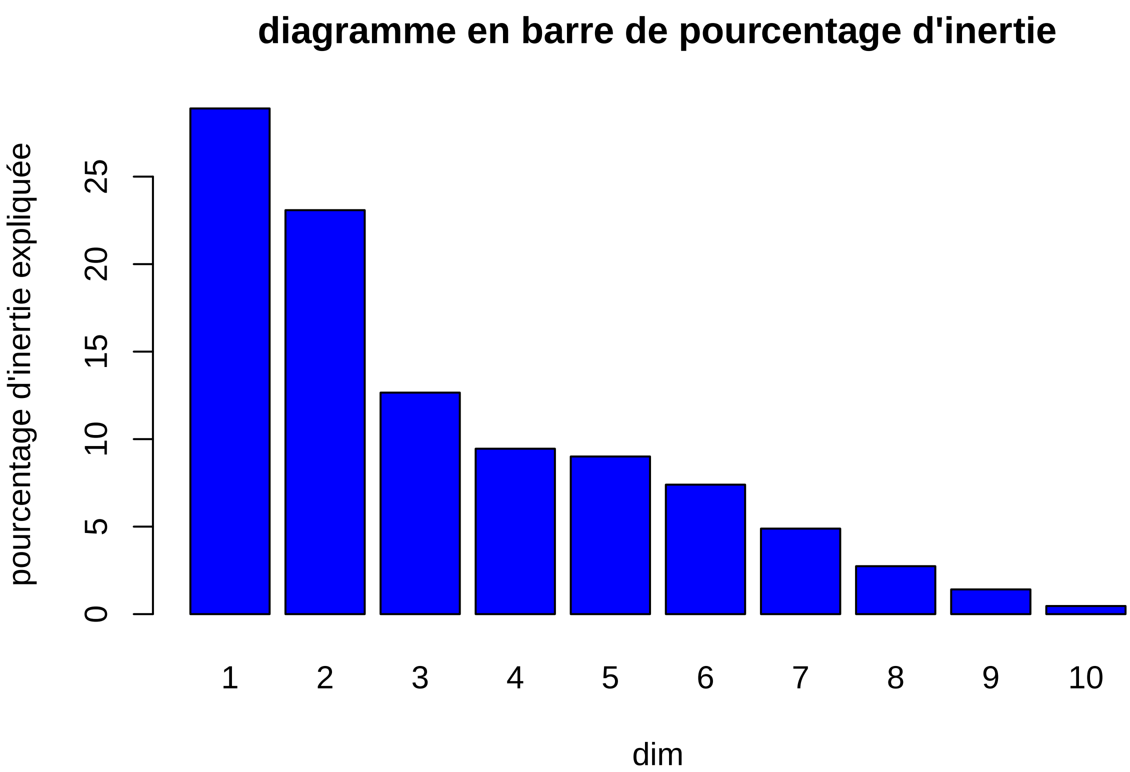
```
## [1] 10
```

```r
n <- nrow(H)
min(n - 1, m - p)
```

```
## [1] 10
```

3.d

```r
pi <- d^2/it*100 #pourcentage d'inertie des axes
barplot(pi,names.arg=1:length(d),xlab="dim",ylab="pourcentage d'inertie expliquée",col="blue",main="diag
```

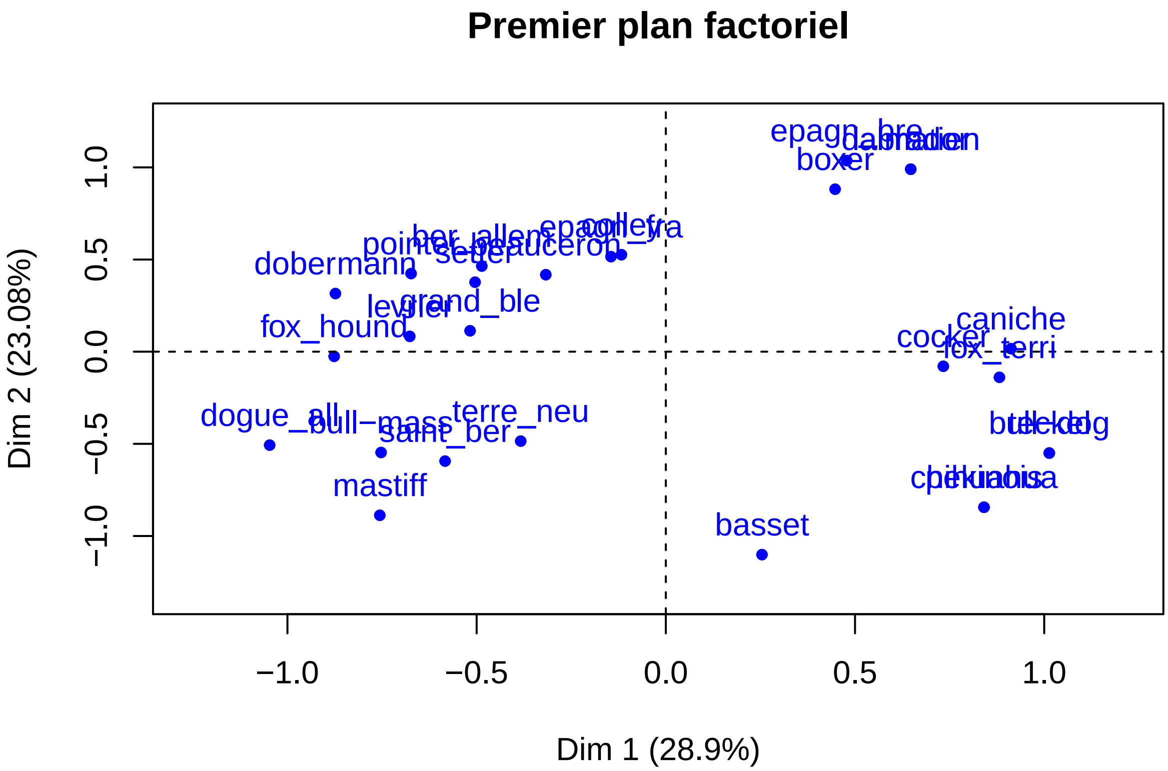


3.e

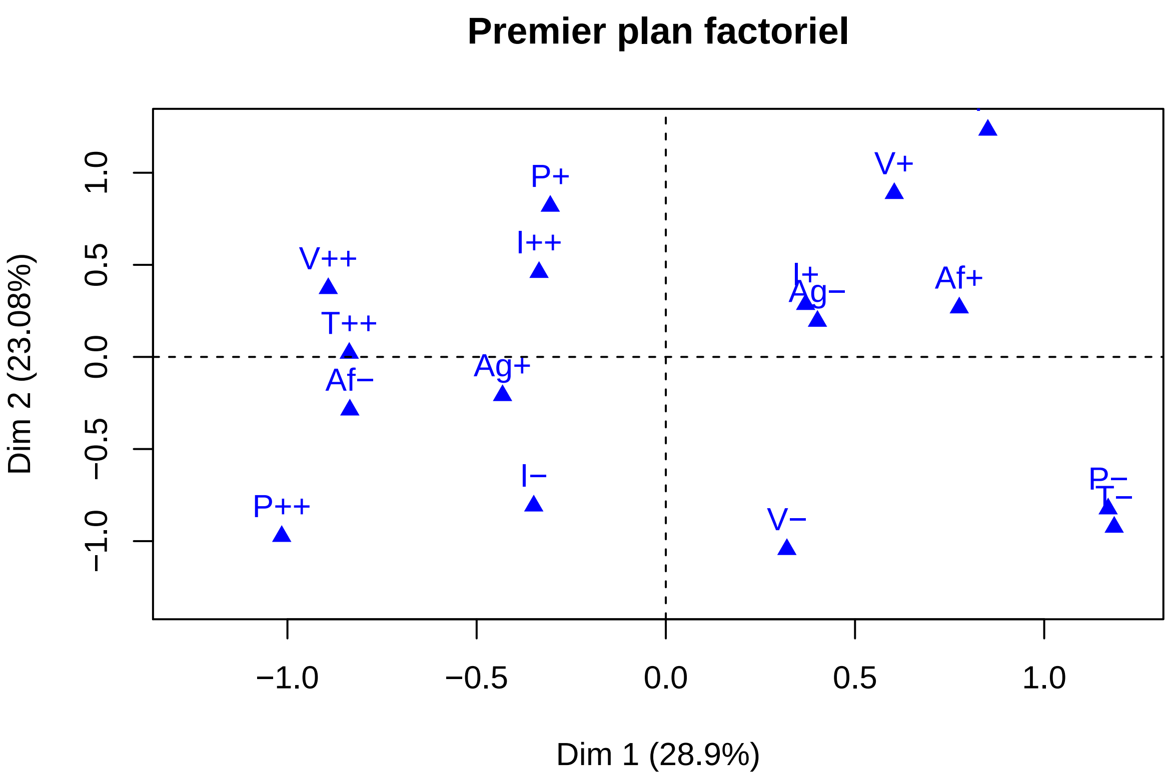
```
X <- sweep(U,2,d,'*')
X <- X[,1:3]
Y <- sweep(V,2,d,'*')
Y <- Y[,1:3]
rownames(X) <- as.matrix(rownames(chiens))
rownames(Y) <- as.matrix(colnames(tabd))
```

3.f plot des individus et des modalités dans le premier plan factoriel

```
xmin <-min(X[,1])
xmax <-max(X[,1])
xlim <-c(xmin, xmax)* 1.2
ymin <-min(X[,2])
ymax <-max(X[,2])
ylim <- c(ymin, ymax)* 1.2
pi2dim <- d[1:2]^2/it*100
pi2dim <- round(pi2dim,2)
xlab <- paste("Dim ", 1, " (", pi2dim[1], "%)", sep = "")
ylab <- paste("Dim ", 2, " (", pi2dim[2], "%)", sep = "")
plot(X[,1:2],xlab=xlab,ylab= ylab,xlim=xlim,ylim=ylim,col="blue",pch=20,main="Premier plan factoriel")
abline(v = 0, lty = 2)
abline(h = 0, lty = 2)
text(X[,1:2],rownames(chiens),col="blue",pos=3)
```




```r
plot(Y[,1:2],xlab=xlab,ylab= ylab,xlim=xlim,ylim=ylim,col="blue",pch=17,main="Premier plan factoriel")
abline(v = 0, lty = 2)
abline(h = 0, lty = 2)
text(Y[,1:2],colnames(tabd),col="blue",pos=3)
```



3.g

```r
moy <- apply(X[which(tabd[,3]==1),],2,mean)
1/d[1:3] * moy # coordonné de Y
```

```
##        dim1        dim2        dim3
## -0.83667535  0.02057846  0.05121744
```

```r
Y[3,]
```

```
##        dim1        dim2        dim3
## -0.83667535  0.02057846  0.05121744
```

3.h:Rapport de correlation entre la variable taille et les deux premieres composantes principale

```r
eta <- function(x) {
  # taille de l'echantillon du premier composante pour chaque modalité de la varoable taille
  ns <- tapply(x, chiens$taille, "length")
  xbarres <-  tapply(x, chiens$taille, "mean")
  denom1 <- sum(ns * (xbarres - mean(x)) ^ 2)
  denom2 <- var(x) * (length(x) - 1)
  rappcorr <- denom1 / denom2
  return (rappcorr)
}
xc1 <- as.data.frame(X)$dim1
xc2 <- as.data.frame(X)$dim2
# rapport de correlation pour la premiere composante avec la variable taille
eta(xc1)
```

```
## [1] 0.8870733
```

```r
# rapport de correlation pour la deuxieme composante avec la variable taille
eta(xc2)
```

```
## [1] 0.5024857
```

4

4.a:Realisation d'une ACM

```r
acmchiens <- MCA(chiens,quali.sup = 7,graph = FALSE)
```

4.b

```r
head(acmchiens$ind$coord,4)
```

```
##                 Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## beauceron -0.3172001 -0.4177013 -0.1014677 -0.2114363 -0.1185095
## basset     0.2541098  1.1012270 -0.1907010  0.2926373 -0.5240085
## ber_allem -0.4863955 -0.4644496 -0.4981339  0.5774253  0.2759021
## boxer      0.4473649 -0.8817779  0.6920158  0.2600018 -0.4555898
```

```r
head(X,4)
```

```
##                 dim1       dim2       dim3
## beauceron -0.3172001  0.4177013  0.1014677
## basset     0.2541098 -1.1012270  0.1907010
## ber_allem -0.4863955  0.4644496  0.4981339
## boxer      0.4473649  0.8817779 -0.6920158
```

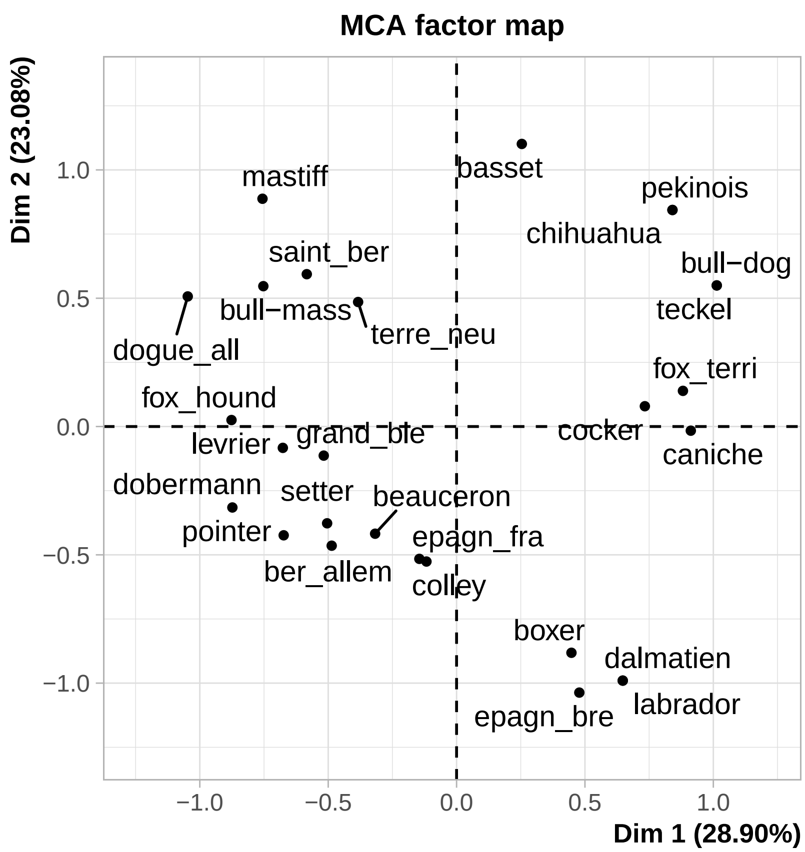
```r
head(acmchiens$var$coord,4)
```

```
##           Dim 1       Dim 2       Dim 3      Dim 4       Dim 5
## T-    1.1849557  0.92389650 -0.61599962  0.1201492 -0.01996350
## T+    0.8510880 -1.23171972  1.01605178  0.3424564 -0.31004022
## T++ -0.8366753 -0.02057846 -0.05121744 -0.1702218  0.11266304
## P-    1.1689180  0.82434462 -0.35877044  0.1648838 -0.05122143
```

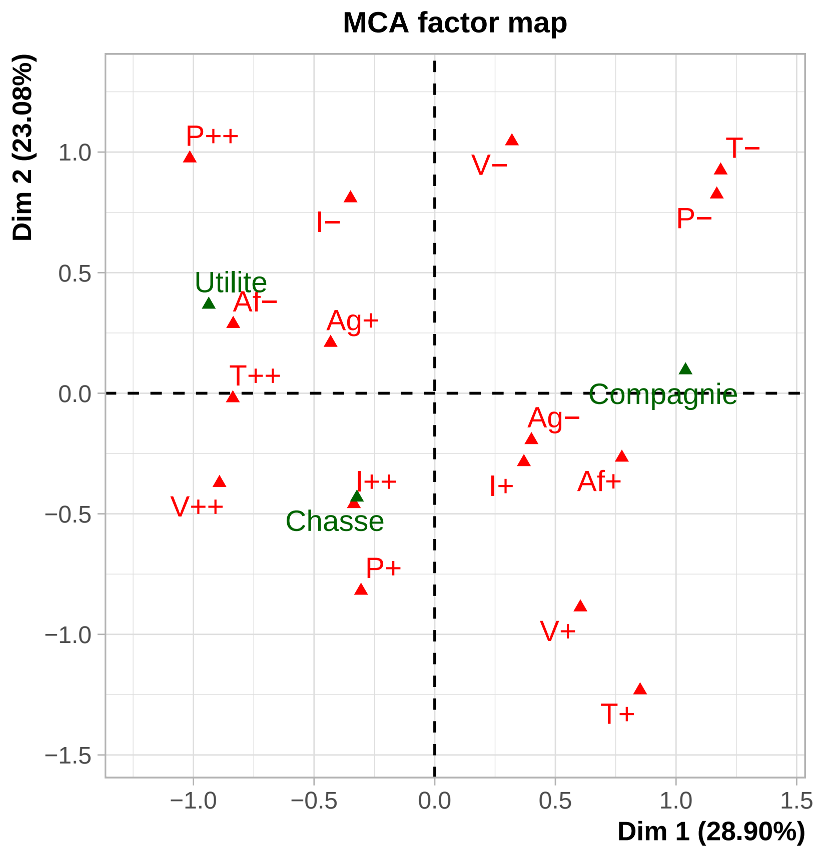
```r
head(Y,4)
```

```
##            dim1        dim2        dim3
## T-    1.1849557 -0.92389650  0.61599962
## T+    0.8510880  1.23171972 -1.01605178
## T++ -0.8366753  0.02057846  0.05121744
## P-    1.1689180 -0.82434462  0.35877044
```

```r
plot(acmchiens,choix="ind",invisible = c("var","quali.sup"))
```




```r
plot(acmchiens,choix="ind",invisible="ind")
```

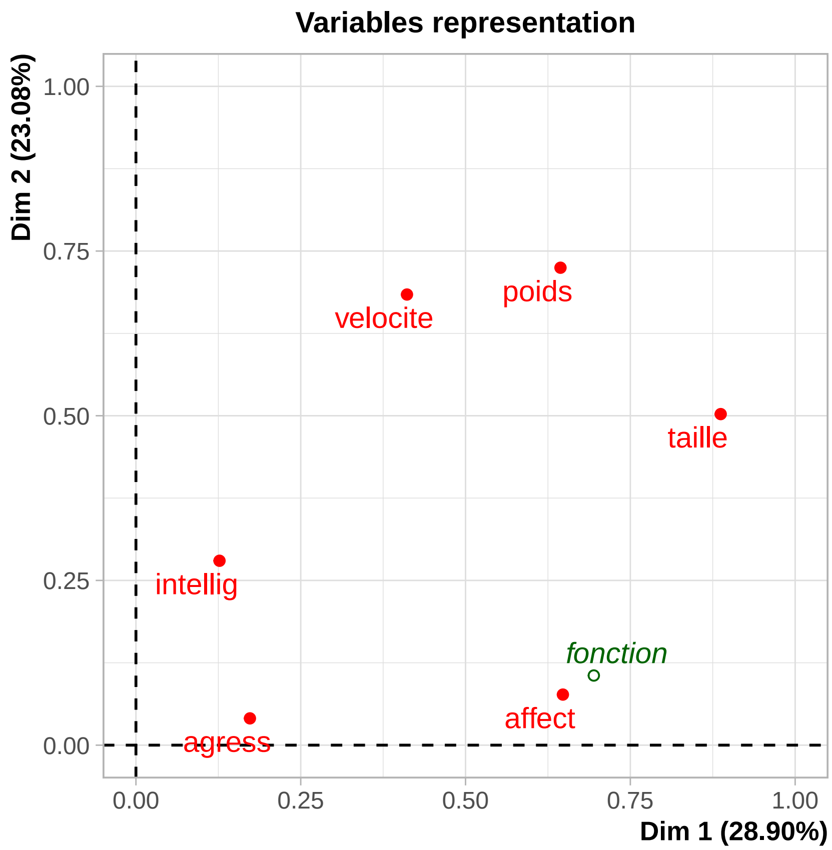


4.c

```
acmchiens$var$eta[,1:2]
```

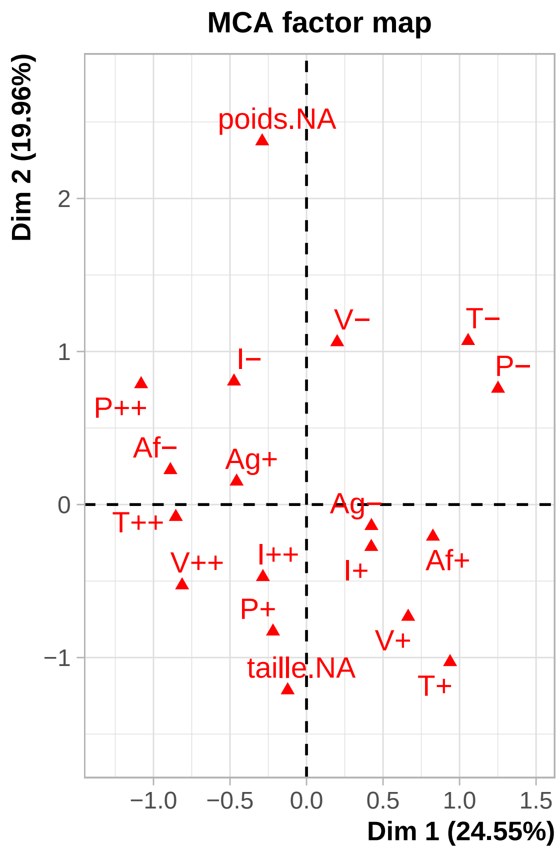
```
##                 Dim 1      Dim 2
## taille   0.8870733 0.50248565
## poids    0.6440465 0.72468773
## velocite 0.4111741 0.68400737
## intellig 0.1267635 0.27987008
## affect   0.6476559 0.07673604
## agress   0.1729238 0.04063686
```

```
plot(acmchiens,choix = "var")
```



4.d

```
chienNA <- H
chienNA[1,1] <- NA
chienNA[2,2] <- NA
mcachienna <- MCA(chienNA,graph = FALSE)
plot(mcachienna,choix = "ind",invisible = "ind")
```



5.

```
Htp <- subset(H,select = c(taille,poids))
Htptabc <- table(Htp)
# Realisation d'une AFC
afctp <- CA(Htptabc,graph = FALSE)
pt <- subset(chiens,select = c(taille,poids))
ptafc <- MCA(pt,graph = FALSE)
# valeur propre de l'ACF
vpafc <- afctp$eig
# valeur propre de l'ACM
vpamc <- ptafc$eig
vpafc
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## dim 1 0.86063286              91.742589                          91.74259
## dim 2 0.07746238               8.257411                         100.00000
```

```
vpamc
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## dim 1  0.9638515              48.192575                          48.19258
## dim 2  0.6391603              31.958016                          80.15059
## dim 3  0.3608397              18.041984                          98.19258
## dim 4  0.0361485               1.807425                         100.00000
```

```
(1 + sqrt(vpafc)) / 2
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
```

```
## dim 1  0.9638515               5.289118                          5.289118
## dim 2  0.6391603               1.936786                          5.500000
```

```r
(1 - sqrt(vpafc)) / 2
```

```
##       eigenvalue percentage of variance cumulative percentage of variance
## dim 1  0.0361485             -4.2891176                         -4.289118
## dim 2  0.3608397             -0.9367855                         -4.500000
```

Ce qui montre que chaque valeur propre de L'ACF correspond a deux valeurs propres de L'ACM