TP2 : Analyse des Correspondences Multiples

1 ACM sur données fictives

Récupérer les jeux de données `chiens.rda`. Il s'agit de données fictives où 27 races de chiens sont décrites avec 7 variables qualitatives.

1. Charger le jeu de données `chiens` dans R avec la commande `load`. Afficher les données. Quelle est la classe de cet objet ?

```
load(file="chiens.rda")
head(chiens)

##           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
## bull-dog      T-    P-       V-       I+    Af+    Ag- Compagnie
## bull-mass    T++   P++       V-      I++    Af-    Ag+   Utilite

dim(chiens)

## [1] 27  7

class(chiens)

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

2. Créez une matrice H contenant la description des $n = 27$ races canines sur uniquement les $p = 6$ premières variables.

```
H <- subset(chiens,select=-fonction)
```

3. On veut effectuer l'ACM de cette matrice H.
   (a) Quelle décomposition en valeurs singulières généralisée (GSVD) faut-il faire ? Réaliser cette DSVG avec R.

```
library(FactoMineR)
K <- tab.disjonctif(H)

#--------------Calcul de la matrice R ------------------
F <- K/sum(K) #matrice des frequences
r<-apply(F,1,sum) #poids des lignes
c<-apply(F,2,sum) #poids des colonnes
R <- diag(1/r)%*%(F-r%*%t(c))%*%diag(1/c)

source("gsvd.R")

U<-gsvd(R,r,c)$U
V<-gsvd(R,r,c)$V
d<-gsvd(R,r,c)$d
```

(b) Montrer qu'en ACM, l'inertie totale des données vaut toujours $\frac{m}{p} - 1$ où m est le nombre total de modalités et p le nombre de variables qualitatives. Vérifiez ensuite avec R que la somme des valeurs singulières trouvées à la question précédente vaut bien $\frac{m}{p} - 1$.

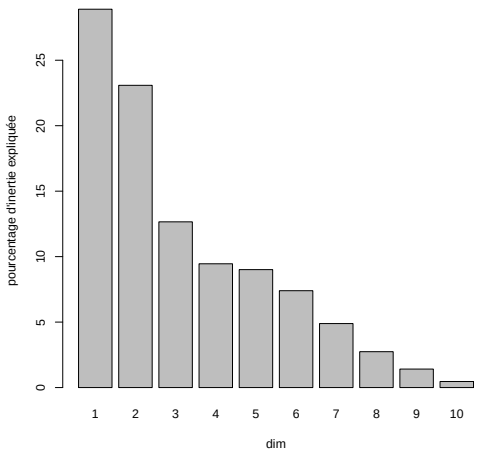
```
sum(d^2) #somme des valeurs singulieres

## [1] 1.7

p <- ncol(H) #nb de variables qualitatives
q <- ncol(K) #nb de modalites
q/p-1

## [1] 1.7
```

(c) Vérifiez également que le nombre maximum de dimension de cette ACM vaut bien $\min(n - 1, m - p)$.

```
round(d^2,digit=3)

##  [1] 0.482 0.385 0.211 0.158 0.150 0.123 0.081 0.046 0.024 0.008

length(d) #10 valeurs propres non nulles

## [1] 10

q-p #min(n-1,q-p)

## [1] 10
```

(d) Représenter dans un diagramme en barre les pourcentages d'inertie expliquée par les dimensions de l'ACM.

```
barplot(d^2/sum(d^2)*100,names.arg=1:length(d),xlab="dim",ylab="pourcentage d'inertie expliquée")
```



(e) Determiner les matrices X et Y des coordonnées factorielles des races de chiens et des modalités des variables qualitatives sur les $k = 3$ premières dimensions. Modifier les noms des lignes et des colonnes dans X et Y afin qu'ils soient parlants.

```
#coordonnees factorielles des races de chiens
X <- data.frame(U[,1:3]%*%diag(d[1:3]))
rownames(X) <- rownames(H)
colnames(X) <- paste("dim", 1:3, sep = "")
round(X,digit=2)

##            dim1  dim2  dim3
## beauceron -0.32  0.42  0.10
## basset     0.25 -1.10  0.19
## ber_allem -0.49  0.46  0.50
## boxer      0.45  0.88 -0.69
## bull-dog   1.01 -0.55  0.16
## bull-mass -0.75 -0.55 -0.50
```

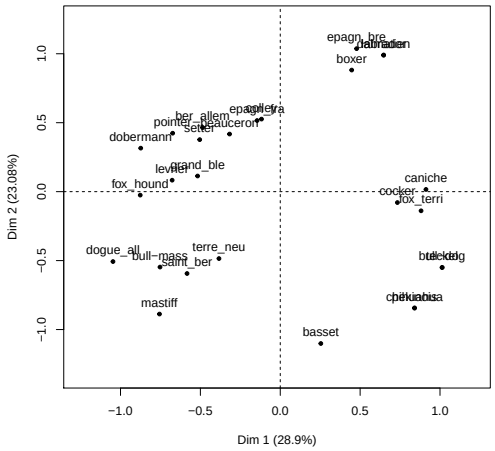
```
## caniche    0.91  0.02  0.58
## chihuahua  0.84 -0.84  0.47
## cocker     0.73 -0.08 -0.66
## colley    -0.12  0.53  0.33
## dalmatien  0.65  0.99 -0.46
## dobermann -0.87  0.32  0.45
## dogue_all -1.05 -0.51 -0.17
## epagn_bre  0.48  1.04 -0.06
## epagn_fra -0.14  0.52 -0.12
## fox_hound -0.88 -0.03  0.36
## fox_terri  0.88 -0.14 -0.05
## grand_ble -0.52  0.11 -0.04
## labrador   0.65  0.99 -0.46
## levrier   -0.68  0.08  0.60
## mastiff   -0.76 -0.89 -0.59
## pekinois   0.84 -0.84  0.47
## pointer   -0.67  0.42  0.69
## saint_ber -0.58 -0.59 -0.89
## setter    -0.50  0.38  0.29
## teckel     1.01 -0.55  0.16
## terre_neu -0.38 -0.49 -0.66
```

```
#coordonnees factorielles des modalités
Y <- data.frame(V[,1:3]%*%diag(d[1:3]))
rownames(Y) <- colnames(K)
colnames(Y) <- paste("dim", 1:3, sep = "")
round(Y,digit=2)
```

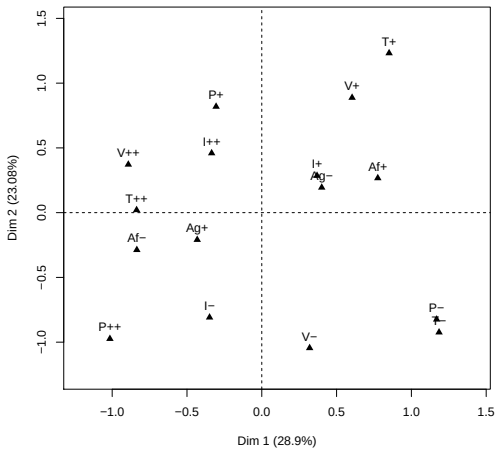
```
##      dim1  dim2  dim3
## T-   1.18 -0.92  0.62
## T+   0.85  1.23 -1.02
## T++ -0.84  0.02  0.05
## P-   1.17 -0.82  0.36
## P+  -0.31  0.82  0.23
## P++ -1.02 -0.97 -1.22
## V-   0.32 -1.04 -0.40
## V+   0.60  0.89 -0.36
## V++ -0.89  0.37  0.76
## I-  -0.35 -0.81  0.35
## I+   0.37  0.29 -0.49
## I++ -0.34  0.46  0.60
## Af- -0.84 -0.29 -0.07
## Af+  0.78  0.27  0.06
## Ag-  0.40  0.19  0.31
## Ag+ -0.43 -0.21 -0.33
```

(f) Faire un plot des individus et des modalités dans le premier plan factoriel.

```
dim1 <- 1
dim2 <- 2
dim <- c(dim1,dim2)
pourc <- round(d[1:3]^2/sum(d^2)*100, digit = 2)
lab.x <- paste("Dim ", dim1, " (", pourc[dim1], "%)", sep = "")
lab.y <- paste("Dim ", dim2, " (", pourc[dim2], "%)", sep = "")
#Plan factoriel des individus
xmin <- min(X[,dim1])
xmax <- max(X[,dim1])
xlim <- c(xmin, xmax)* 1.2
ymin <- min(X[,dim2])
ymax <- max(X[,dim2])
ylim <- c(ymin, ymax)* 1.2
plot(X[,dim],xlab=lab.x,ylab=lab.y,xlim=xlim,ylim=ylim,pch=20)
abline(v = 0, lty = 2)
abline(h = 0, lty = 2)
text(X[,dim],labels=rownames(X),pos=3)
```




```
#Plan factoriel des modalites
xmin <- min(Y[,dim1])
xmax <- max(Y[,dim1])
xlim <- c(xmin, xmax)* 1.2
ymin <- min(Y[,dim2])
ymax <- max(Y[,dim2])
ylim <- c(ymin, ymax)* 1.2
plot(Y[,dim],xlab=lab.x,ylab=lab.y,xlim=xlim,ylim=ylim,pch=17)
abline(v = 0, lty = 2)
abline(h = 0, lty = 2)
text(Y[,dim],labels=rownames(Y),pos=3)
```



(g) Utiliser la relation quasi-barycentrique pour retouver les coordonnées factorielles de la modalité T++ à partir des coordonnées factorielles des races de chiens.

```
which(K[,1]==1) #indice des lignes des chiens T++

##    basset  bull-dog   caniche chihuahua fox_terri  pekinois    teckel
##         2         5         7         8        17        22        26

moy <- apply(X[which(K[,1]==1),],2,mean) #moyenne des coord. fact. des chiens T++
moy*(1/d[1:3]) #relation quasi-barycentrique

##  dim1  dim2  dim3
##  1.18 -0.92  0.62

Y[1,] #coord. fact. de T++

##    dim1  dim2 dim3
## T-  1.2 -0.92 0.62
```

(h) Quels est le rapport de corrélation entre la variable `taille` avec la première composante principale ? Entre la variable `taille` et la seconde composante principale ?

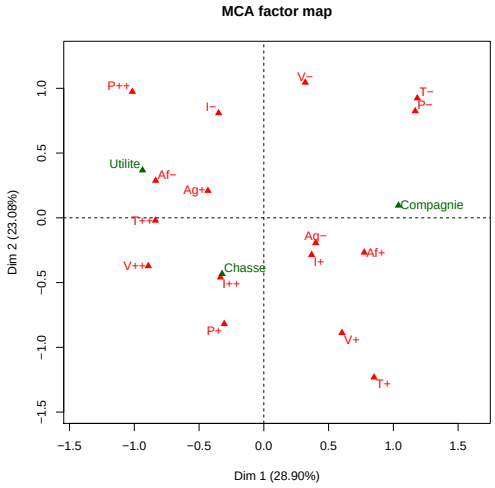
```
eta2 <- function(x, gpe) {
  moyennes <- tapply(x, gpe, mean)
  effectifs <- tapply(x, gpe, length)
  varinter <- (sum(effectifs * (moyennes - mean(x)) ^ 2))
  vartot <- (var(x) * (length(x) - 1))
  res <- varinter / vartot
  return(res)
}
```

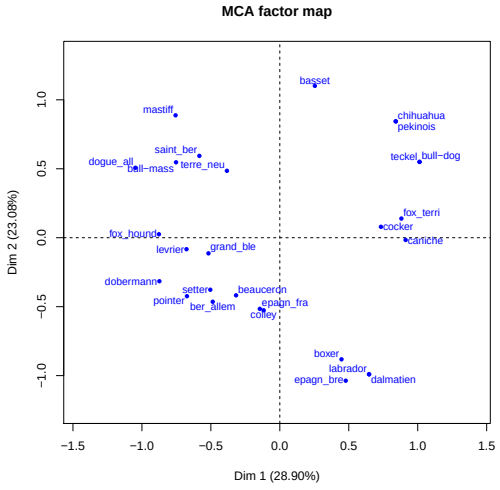
```
eta2(X$dim1,chiens$taille)

## [1] 0.89

eta2(X$dim2,chiens$taille)

## [1] 0.5
```

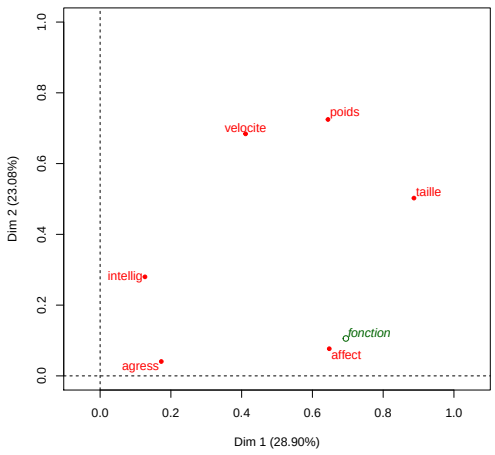
4. On veut maintenant utiliser la fonction `MCA` du package `FactoMineR`.

(a) Faire l'ACM des données sur les races canines en mettant la variable `fonction` en illustratif.

```
res <- MCA(chiens, quali.sup = 7)
```








```
#print(res)
```

(b) Retrouvez les résulats numériques et les graphiques de la question 2.

```
head(X)
```

```
##            dim1  dim2  dim3
## beauceron -0.32  0.42  0.10
## basset     0.25 -1.10  0.19
## ber_allem -0.49  0.46  0.50
## boxer      0.45  0.88 -0.69
## bull-dog   1.01 -0.55  0.16
## bull-mass -0.75 -0.55 -0.50
```

```
head(res$ind$coord)
```

```
##           Dim 1 Dim 2 Dim 3 Dim 4 Dim 5
## beauceron -0.32 -0.42 -0.10 -0.21 -0.12
## basset     0.25  1.10 -0.19  0.29 -0.52
## ber_allem -0.49 -0.46 -0.50  0.58  0.28
## boxer      0.45 -0.88  0.69  0.26 -0.46
## bull-dog   1.01  0.55 -0.16 -0.35  0.33
## bull-mass -0.75  0.55  0.50  0.66  0.72
```

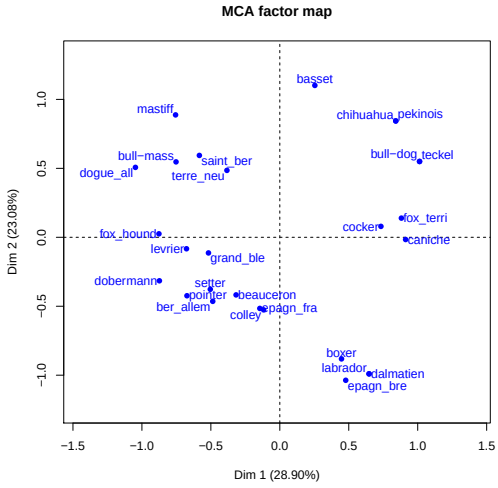
```
head(Y)
```

```
##      dim1   dim2   dim3
## T-   1.18 -0.924  0.616
## T+   0.85  1.232 -1.016
## T++ -0.84  0.021  0.051
## P-   1.17 -0.824  0.359
## P+  -0.31  0.819  0.231
## P++ -1.02 -0.974 -1.222
```

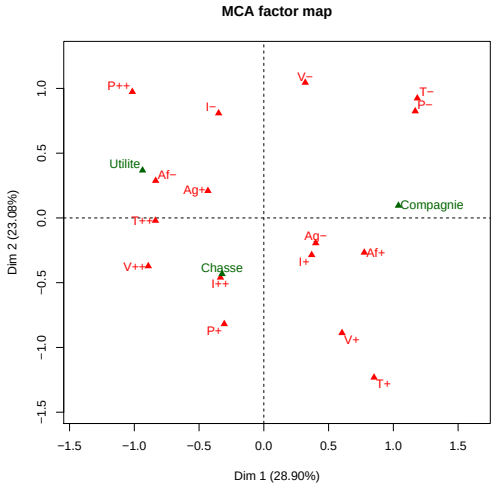
```
head(res$var$coord)
```

```
##     Dim 1  Dim 2  Dim 3  Dim 4  Dim 5
## T-   1.18  0.924 -0.616  0.120 -0.020
## T+   0.85 -1.232  1.016  0.342 -0.310
## T++ -0.84 -0.021 -0.051 -0.170  0.113
## P-   1.17  0.824 -0.359  0.165 -0.051
## P+  -0.31 -0.819 -0.231 -0.118 -0.190
## P++ -1.02  0.974  1.222  0.068  0.615
```

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




```
plot(res,choix="ind",invisible="ind")
```

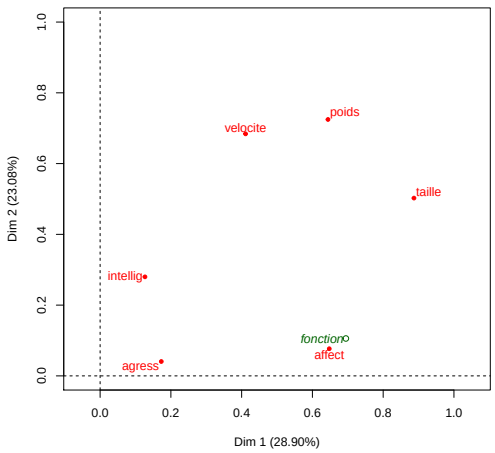


(c) Retrouver les rapports de corrélations entre les variables qualitatives et les deux premières composantes principales. Faire le plot des variables en fonction de ces rapports de corrélation en utilisant la fonction `plot.MCA`.

```
res$var$eta2[,1:2]
```

```
##          Dim 1 Dim 2
## taille    0.89 0.502
## poids     0.64 0.725
## velocite  0.41 0.684
## intellig  0.13 0.280
## affect    0.65 0.077
## agress    0.17 0.041
```

```
plot(res,choix="var",invisible=c("ind"))
```

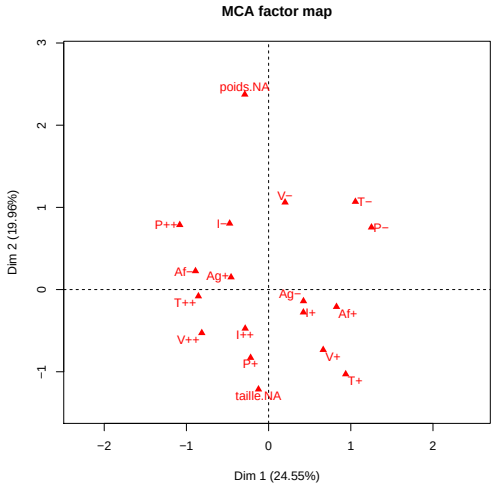


(d) Mettre des données manquantes dans les données avec le code suivant :

```
chiensNA <- H
chiensNA[1,1] <-NA
chiensNA[2,2] <-NA
```

(e) Faire l'ACM de `chiensNA`. Comment les données manquantes sont-elles prises en compte dans la fonction `MCA` du package `FactoMineR` ?

```
res2 <- MCA(chiensNA,graph=FALSE)
plot(res2,choix="ind",invisible="ind")
```



5. On veut maintenant comparer l'ACM et l'AFC dans le cas particulier de deux variables qualitatives.

(a) Avec la fonction `CA` de `FactoMineR`, effectuer l'AFC du tableau de contingence croisant les variables `taille` et `poids`.

```
N <- table(H[,1:2])
resca<-CA(N,ncp=2,graph=FALSE) #AFC
```

(b) Avec la fonction `MCA`, effectuer l'ACM des deux premières colonnes des données `chiens`.

```
resmca<-MCA(H[,1:2],graph=FALSE) #ACM
```

(c) Comparez les valeurs propres des deux analyses et vérifiez que vous retrouvez les relations du cours.

```
resca$eig #valeurs propres de CA

##       eigenvalue percentage of variance cumulative percentage of variance
## dim 1      0.861                  91.7                                92
## dim 2      0.077                   8.3                               100
```

```
resmca$eig # valeurs propres de MCA
```

```
##       eigenvalue percentage of variance cumulative percentage of variance
## dim 1      0.964                  48.2                                48
## dim 2      0.639                  32.0                                80
## dim 3      0.361                  18.0                                98
## dim 4      0.036                   1.8                               100
```

```
#relation entre les valeurs propres des deux analyses
mu <- resca$eig[,1]
(1+sqrt(mu))/2 #on retrouve des deux premieres valeurs propres de MCA
```

```
## [1] 0.96 0.64
```

```
(1-sqrt(mu))/2 #on retrouve des deux dernieres valeurs de MCA
```

```
## [1] 0.036 0.361
```

2 ACM avec données manquantes et choix du nombre de composantes

Le package R `missMDA` permet de gérer les données manquantes en ACP et en ACM, et de choisir le nombre de composantes par validation croisée. Ce travail est à réaliser en binôme et à me rendre.

1. Regarder les vidéos concernant ce package : `https://www.youtube.com/user/HussonFrancois`
2. Préparer un document avec `Rmardown` qui décrit les principales fonctionnalités de ce package, avec à chaque fois une explication de la méthode, des exemples et du code.

3 ACM et clustering

1. On reprend ici l'exercice 3 du devoir surveillé de décembre 2013.
   (a) Répondre aux questions.
   (b) Retrouver le code R de cet exercice.

```
load("credit.Rdata")
library(FactoMineR)

#description des donnees
str(credit)
```

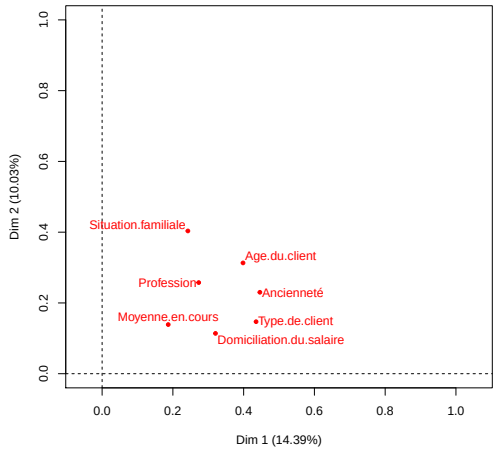
```
## 'data.frame': 468 obs. of  7 variables:
##  $ Type.de.client          : Factor w/ 2 levels "bon client","mauvais client": 1 1 2 1 1 1 1 1 1 1 ...
##  $ Age.du.client           : Factor w/ 4 levels "de 23 à 40 ans",..: 4 3 1 1 3 1 4 4 2 4 ...
##  $ Situation.familiale     : Factor w/ 4 levels "célibataire",..: 1 1 4 2 1 1 3 3 1 1 ...
##  $ Ancienneté              : Factor w/ 5 levels "anc. 1 an ou moins",..: 5 1 4 2 4 1 4 5 2 3 ...
##  $ Domiciliation.du.salaire: Factor w/ 2 levels "domicile salaire",..: 1 1 1 1 2 1 1 1 1 1 ...
##  $ Profession              : Factor w/ 3 levels "cadre","employé",..: 2 2 2 2 2 2 1 1 2 2 ...
##  $ Moyenne.en.cours        : Factor w/ 3 levels "de 2 à 5 KF encours",..: 1 1 1 3 1 1 1 1 2 1 ...
```

```
apply(credit,2,table)
```

```
## $Type.de.client
##
##     bon client mauvais client
##            237            231
##
## $Age.du.client
##
##  de 23 à 40 ans  de 40 à 50 ans moins de 23 ans  plus de 50 ans
##             150             122              88             108
##
## $Situation.familiale
##
## célibataire     divorcé       marié        veuf
##         170          61         221          16
##
```

```
## $Ancienneté
##
## anc. 1 an ou moins  anc. de 1 à 4 ans  anc. de 4 à 6 ans
##                199                  47                  69
## anc. de 6 à 12 ans   anc. plus 12 ans
##                 66                  87
##
## $Domiciliation.du.salaire
##
##     domicile salaire non domicile salaire
##                  316                  152
##
## $Profession
##
##            cadre          employé profession autre
##               77              237              154
##
## $Moyenne.en.cours
##
##  de 2 à 5 KF encours moins de 2KF encours plus de 5 KF encours
##                  308                   98                   62
```

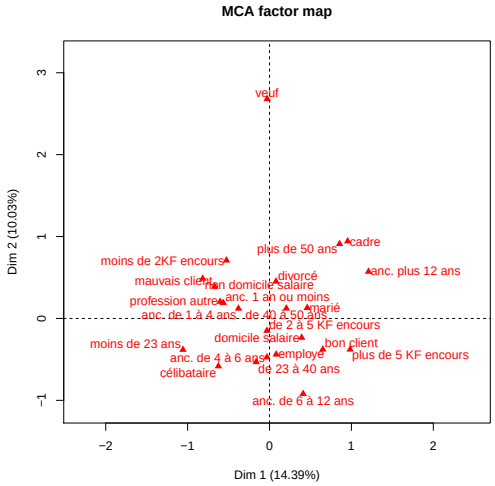
```
#ACM sur les 7 variables qualitatives
res <- MCA(credit,graph=FALSE)
dim=c(1,2)
plot(res,axes=dim,choix="var",invisible="ind")
```



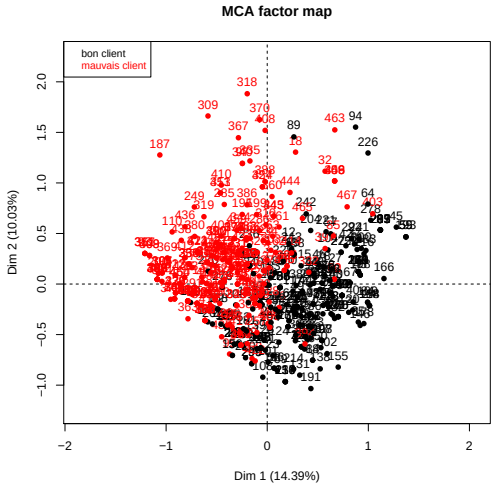

```
res$var$eta2
```

```
##                          Dim 1 Dim 2  Dim 3  Dim 4   Dim 5
## Type.de.client            0.44  0.15 0.0013 0.0739 1.8e-08
## Age.du.client             0.40  0.31 0.5174 0.5053 6.5e-03
## Situation.familiale       0.24  0.40 0.2769 0.2753 3.6e-01
## Ancienneté                0.45  0.23 0.2504 0.1022 4.1e-01
## Domiciliation.du.salaire  0.32  0.11 0.0015 0.0031 1.3e-02
## Profession                0.27  0.26 0.1471 0.0957 1.3e-01
## Moyenne.en.cours          0.19  0.14 0.0704 0.1971 1.8e-01
```

```
plot(res,axes=dim,invisible="ind")
```

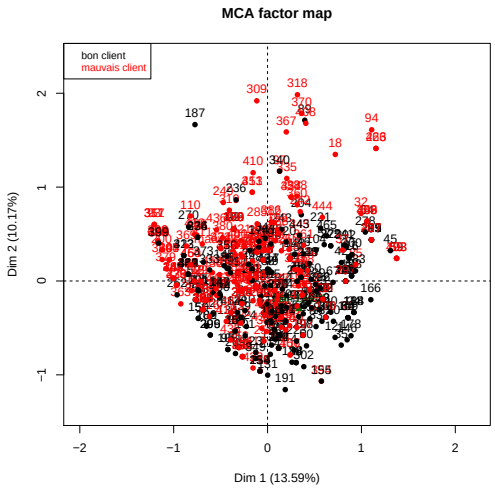



```
plot(res,axes=dim,choix="ind",habillage=1,invisible="var")
```




```
#classification hiérarchique ascendante toutes les sur coordonnées de l'ACM
#HCPC Hierarchical Clustering on Principal Components

res.hcpc <- HCPC(res,nb.clust=2,graph=FALSE)
res.hcpc$desc.var$test.chi2

##                           p.value df
## Type.de.client           4.6e-50  1
## Ancienneté               1.4e-32  4
## Domiciliation.du.salaire 5.6e-26  1
## Age.du.client            2.0e-19  3
## Situation.familiale      2.4e-14  3
## Moyenne.en.cours         2.5e-14  2
## Profession               3.1e-14  2

res.hcpc$desc.var$category$'1'[,1:3]

##                                           Cla/Mod Mod/Cla Global
## Type.de.client=mauvais client                89.2    80.8     49
## Domiciliation.du.salaire=non domicile salaire 89.5    53.3     32
## Ancienneté=anc. 1 an ou moins                78.9    61.6     43
## Age.du.client=moins de 23 ans                90.9    31.4     19
## Situation.familiale=célibataire              75.9    50.6     36
## Profession=profession autre                  76.6    46.3     33
## Moyenne.en.cours=moins de 2KF encours        76.5    29.4     21
## Ancienneté=anc. de 1 à 4 ans                 78.7    14.5     10
## Age.du.client=de 40 à 50 ans                 46.7    22.4     26
## Ancienneté=anc. de 6 à 12 ans                37.9     9.8     14
```

```
## Profession=cadre                                  23.4     7.1     16
## Age.du.client=plus de 50 ans                      25.0    10.6     23
## Moyenne.en.cours=plus de 5 KF encours             12.9     3.1     13
## Situation.familiale=marié                         35.3    30.6     47
## Ancienneté=anc. plus 12 ans                        4.6     1.6     19
## Domiciliation.du.salaire=domicile salaire         37.7    46.7     68
## Type.de.client=bon client                         20.7    19.2     51

res.hcpc$desc.var$category$'2'[,1:3]

##                                                Cla/Mod Mod/Cla Global
## Type.de.client=bon client                         79.3    88.3     51
## Domiciliation.du.salaire=domicile salaire         62.3    92.5     68
## Ancienneté=anc. plus 12 ans                       95.4    39.0     19
## Situation.familiale=marié                         64.7    67.1     47
## Moyenne.en.cours=plus de 5 KF encours             87.1    25.4     13
## Age.du.client=plus de 50 ans                      75.0    38.0     23
## Profession=cadre                                  76.6    27.7     16
## Ancienneté=anc. de 6 à 12 ans                     62.1    19.2     14
## Age.du.client=de 40 à 50 ans                      53.3    30.5     26
## Ancienneté=anc. de 1 à 4 ans                      21.3     4.7     10
## Moyenne.en.cours=moins de 2KF encours             23.5    10.8     21
## Profession=profession autre                       23.4    16.9     33
## Situation.familiale=célibataire                   24.1    19.2     36
## Age.du.client=moins de 23 ans                      9.1     3.8     19
## Ancienneté=anc. 1 an ou moins                     21.1    19.7     43
## Domiciliation.du.salaire=non domicile salaire     10.5     7.5     32
## Type.de.client=mauvais client                     10.8    11.7     49

#Scoring sur donnees qualitatives à la main

n <- nrow(credit)
set.seed(1)
test.sample <- sample(1:n,150)
train.sample <- (1:n)[-test.sample]

mca <- MCA(credit,ncp=15,quali.sup=1,ind.sup=test.sample,graph=FALSE)
plot(mca,choix="ind",habillage=1,invisible="var")
```




```
y <- credit[,1]
train <- mca$ind$coord
test <- mca$ind.sup$coord

library(MASS)
m <- lda(train, y[-test.sample])
yhat <- predict(m, test)$class
table(y[test.sample],yhat)

##                yhat
##                 bon client mauvais client
##   bon client            54             16
```

```
##   mauvais client         30             50

sum(yhat != y[test.sample])/length(yhat)

## [1] 0.31

#Scoring sur donnees qualitatives avec un package
library(DiscriMiner)
?disqual
#Analyse Factorielle Discriminante (AFD) sur données qualitatives
disq <- disqual(credit[,-1],credit[,1],learn=train.sample,test=test.sample,validation="learntest")
disq$confusion

##                 predicted
## original         bon client mauvais client
##   bon client             58             12
##   mauvais client         39             41

disq$error_rate

## [1] 0.34

head(disq$scores)

##      bon client mauvais client
## [1,]        149            503
## [2,]        286            334
## [3,]        418            171
## [4,]        293            325
## [5,]        375            224
## [6,]        374            225

head(disq$classification)

## [1] mauvais client mauvais client bon client     mauvais client
## [5] bon client     bon client
## Levels: bon client mauvais client

disq <- disqual(credit[,-1],credit[,1],validation="crossval")
disq$error_rate

## [1] 0.31
```

2. On veut ici vérfier qu'il est équivalent de réaliser une classification ascendante hiérarchique sur la matrice des distances du χ^2 entre les données brutes recodées dans un tableau disjonctif complet, ou sur la matrice des distances Euclidiennes entre les données décrites avec toutes les composantates de l'ACM. Ce travail est à réaliser en binôme et à me rendre.
   (a) Faire une fonction pour calculer la matrice des distances du χ^2 entre les n lignes d'une matrice de données qualitatives. Appliquez cette fonction à un jeu de données de votre choix.
   (b) Faire l'ACM de ce jeu de données et conserver toutes les composantes principales. Calculer la matrice des distances Euclidiennes entre les n observations décrites par toutes les composantes principales.
   (c) Comparez la hiérarchie de Ward obtenue avec ces deux matrices de distances.