

Devoir_4

EL_Hadrami

25/12/2020

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

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

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

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

```
## corrplot 0.84 loaded
```

```
library("ca")
```

1.Chargement des données smoke

```
datasmoke <- ca::smoke
datasmoke
```

```
##      none light medium heavy
## SM      4      2      3      2
## JM      4      3      7      4
## SE     25     10     12      4
## JE     18     24     33     13
## SC     10      6      7      2
```

AFC et SVD généralisée

```
f <- as.matrix(datasmoke) / sum(datasmoke)
# distribution marginale ligne et colonne
r <- apply(f,1,sum)
c <- apply(f,2,sum)
# matrice Z
Z <- (f-r%*%t(c))/r%*%t(c)
```

Creation de la fonction gsvd

```
gsvd <- function(Z,r,c){
  #Z matrice numerique de dimension (n,p) et de rang k
  #r poids de la metrique des lignes N=diag(r)
  # c poids de la metrique des colonnes M=diag(c)
  #-----sortie-----
  # d vecteur de taille k contenant les valeurs singulieres (racines carres des valeurs propres)
  # U matrice de dimension (n,k) des vecteurs propres de de ZMZ'N
  # V matrice de dimension (p,k) des vecteurs propres de de Z'NZM
  k <-qr(Z)$rank
  colnames<-colnames(Z)
  rownames<-rownames(Z)
```

```

Z <-as.matrix(Z)
Ztilde <-diag(sqrt(r)) %*% Z %*%diag(sqrt(c))
e <-svd(Ztilde)
U <-diag(1/sqrt(r))%*%e$u[,1:k] # Attention : ne s'ecrit comme cela que parceque N et M sont diagonale
V <-diag(1/sqrt(c))%*%e$v[,1:k]
d <- e$d[1:k]
rownames(U) <- rownames
rownames(V) <- colnames
if(length(d)>1)
  colnames(U) <-colnames(V) <-paste("dim", 1:k, sep = "")
return(list(U=U,V=V,d=d))
}

```

Calcul de X(profil ligne), Y(profil colonne) et d

```

U <- gsvd(Z,r,c)$U
V <- gsvd(Z,r,c)$V
d <- gsvd(Z,r,c)$d
# Utilisation de la commande sweep pour calculer les cordonnés X et Y
X <- sweep(U,2,d,'*')
Y <- sweep(V,2,d,'*')

```

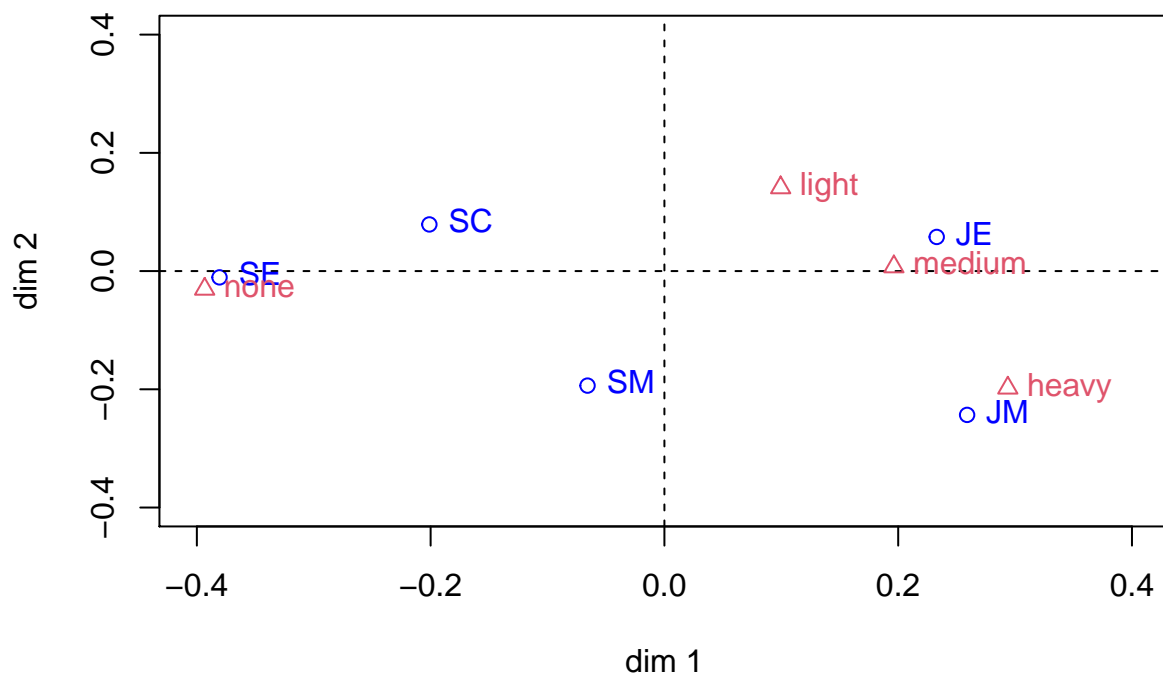
Representation de X et Y sur le premier plan de l'AFC

```

plot(X[,1:2],xlab="dim 1",ylab="dim 2",xlim=c(-0.4,0.4),ylim=c(-0.4,0.4),col="blue",main="Premier plan :")
abline(v = 0, lty = 2)
abline(h = 0, lty = 2)
text(X[,1:2],rownames(datasmoke),col="blue",pos=4)
points(Y[,1:2],pch=2,col=2)
text(Y[,1:2],colnames(datasmoke),pos=4,col=2)

```

Premier plan factoriel



Le pourcentage d'inertie expliquée par le premier plan factoriel de l'AFC

```
IT <-sum(d^2) #Inertie totale
d[1:2]^2/IT*100 #pourcentage d'inertie des axes
```

```
## [1] 87.75587 11.75865
```

```
sum(d[1:2]^2/IT)*100#pourcentage d'inertie du plan
```

```
## [1] 99.51453
```

Les deux premières dimensions de l'AFC donnent 99.51% de la variation, donc le premier plan factoriel de L'AFC peut être

3. Retrouvons ces résultats avec le package FactoMineR et la fonction CA

```
afc <- CA(datasmoke,graph = FALSE)
afc$eig
```

```
##          eigenvalue percentage of variance cumulative percentage of variance
## dim 1 0.0747591059           87.7558731           87.75587
## dim 2 0.0100171805           11.7586535           99.51453
## dim 3 0.0004135741            0.4854734           100.00000
```

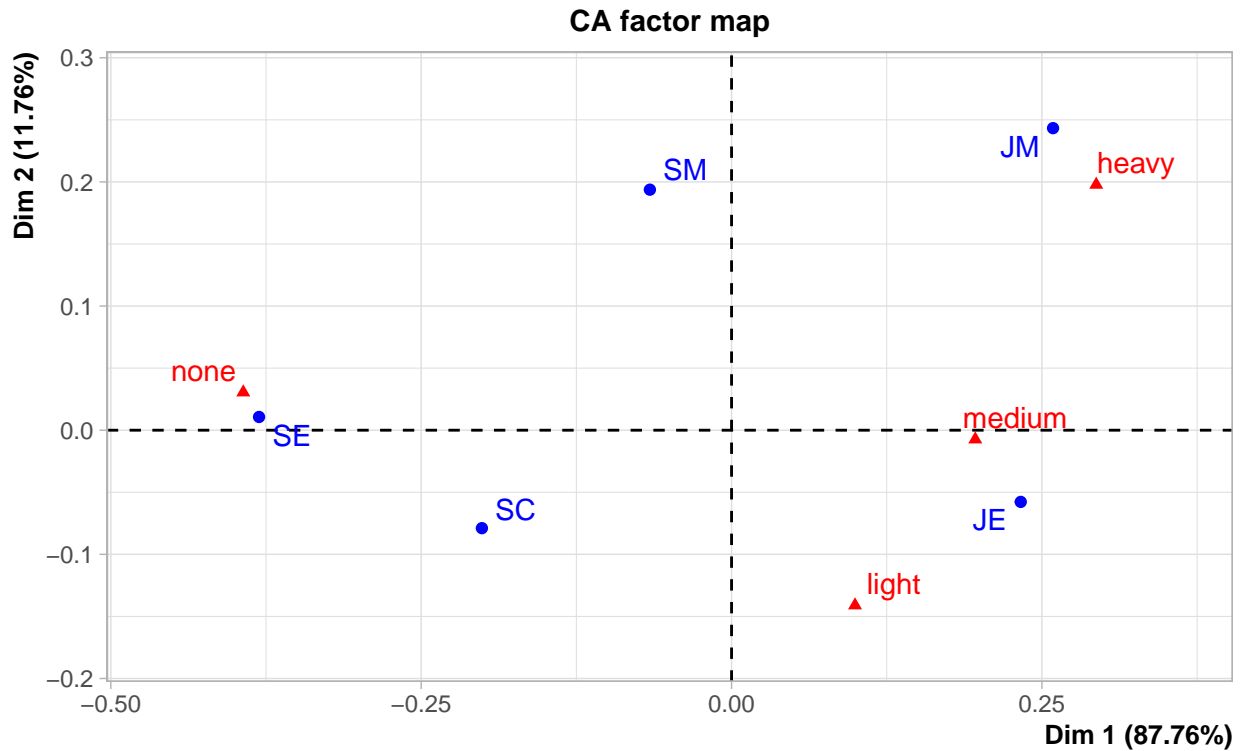
```
row <- get_ca_row(afc)
row$coord # matrice X
```

```
##          Dim 1          Dim 2          Dim 3
## SM -0.06576838  0.19373700  0.070981028
## JM  0.25895842  0.24330457 -0.033705190
## SE -0.38059489  0.01065991 -0.005155757
## JE  0.23295191 -0.05774391  0.003305371
## SC -0.20108912 -0.07891123 -0.008081076
```

```
col <- get_ca_col(afc)
col$coord # matrice Y
```

```
##          Dim 1          Dim 2          Dim 3
## none -0.39330845  0.030492071 -0.0008904827
## light  0.09945592 -0.141064289  0.0219980349
## medium 0.19632096 -0.007359109 -0.0256590867
## heavy  0.29377599  0.197765656  0.0262108499
```

```
# representation de profil ligne(X) et profil colonne sur le plan
plot(afc)
```



Exercice 2:Données textuelles

Chargement du jeux de données

```
dataw <- read.csv("data/writers.csv",header = TRUE,row.names = 1)
head(dataw,4)
```

```
##      B  C  D  F  G  H  I  L  M  N  P  R  S  U  W  Y
## CD1 34 37 44 27 19 39 74 44 27 61 12 65 69 22 14 21
## CD2 18 33 47 24 14 38 66 41 36 72 15 62 63 31 12 18
## CD3 32 43 36 12 21 51 75 33 23 60 24 68 85 18 13 14
## RD1 13 31 55 29 15 62 74 43 28 73  8 59 54 32 19 20
```

```
summary(dataw)
```

```
##      B      C      D      F
## Min.   : 8.00  Min.   :14.00  Min.   :28.00  Min.   :12.00
## 1st Qu.:13.00  1st Qu.:20.00  1st Qu.:40.00  1st Qu.:17.00
## Median :17.00  Median :28.00  Median :43.00  Median :24.00
## Mean   :17.76  Mean   :26.94  Mean   :47.65  Mean   :21.82
## 3rd Qu.:19.00  3rd Qu.:33.00  3rd Qu.:55.00  3rd Qu.:26.00
## Max.   :34.00  Max.   :43.00  Max.   :80.00  Max.   :31.00
##      G      H      I      L
## Min.   :11.00  Min.   :38.00  Min.   : 61.00  Min.   :15.00
## 1st Qu.:16.00  1st Qu.:53.00  1st Qu.: 66.00  1st Qu.:33.00
## Median :19.00  Median :62.00  Median : 73.00  Median :39.00
## Mean   :21.18  Mean   :62.29  Mean   : 74.47  Mean   :36.47
## 3rd Qu.:27.00  3rd Qu.:68.00  3rd Qu.: 75.00  3rd Qu.:43.00
## Max.   :40.00  Max.   :96.00  Max.   :116.00  Max.   :54.00
##      M      N      P      R
## Min.   :20.00  Min.   : 57.00  Min.   :  8.00  Min.   :40.00
## 1st Qu.:25.00  1st Qu.: 68.00  1st Qu.:13.00  1st Qu.:56.00
```

```
## Median :29.00 Median : 71.00 Median :15.00 Median :63.00
## Mean :29.59 Mean : 75.18 Mean :16.12 Mean :60.06
## 3rd Qu.:35.00 3rd Qu.: 78.00 3rd Qu.:17.00 3rd Qu.:68.00
## Max. :40.00 Max. :129.00 Max. :30.00 Max. :78.00
## S U W Y
## Min. : 54.00 Min. :18.00 Min. :11.00 Min. : 9.00
## 1st Qu.: 63.00 1st Qu.:20.00 1st Qu.:14.00 1st Qu.:14.00
## Median : 67.00 Median :22.00 Median :20.00 Median :18.00
## Mean : 69.94 Mean :26.06 Mean :24.18 Mean :18.59
## 3rd Qu.: 72.00 3rd Qu.:31.00 3rd Qu.:25.00 3rd Qu.:23.00
## Max. :104.00 Max. :50.00 Max. :58.00 Max. :30.00
```

Test de khi-deux

```
dataextr <- dataw[1:15,1:15]
chisq.test(dataextr)
```

```
##
## Pearson's Chi-squared test
##
## data: dataextr
## X-squared = 433.89, df = 196, p-value < 2.2e-16
```

Decision

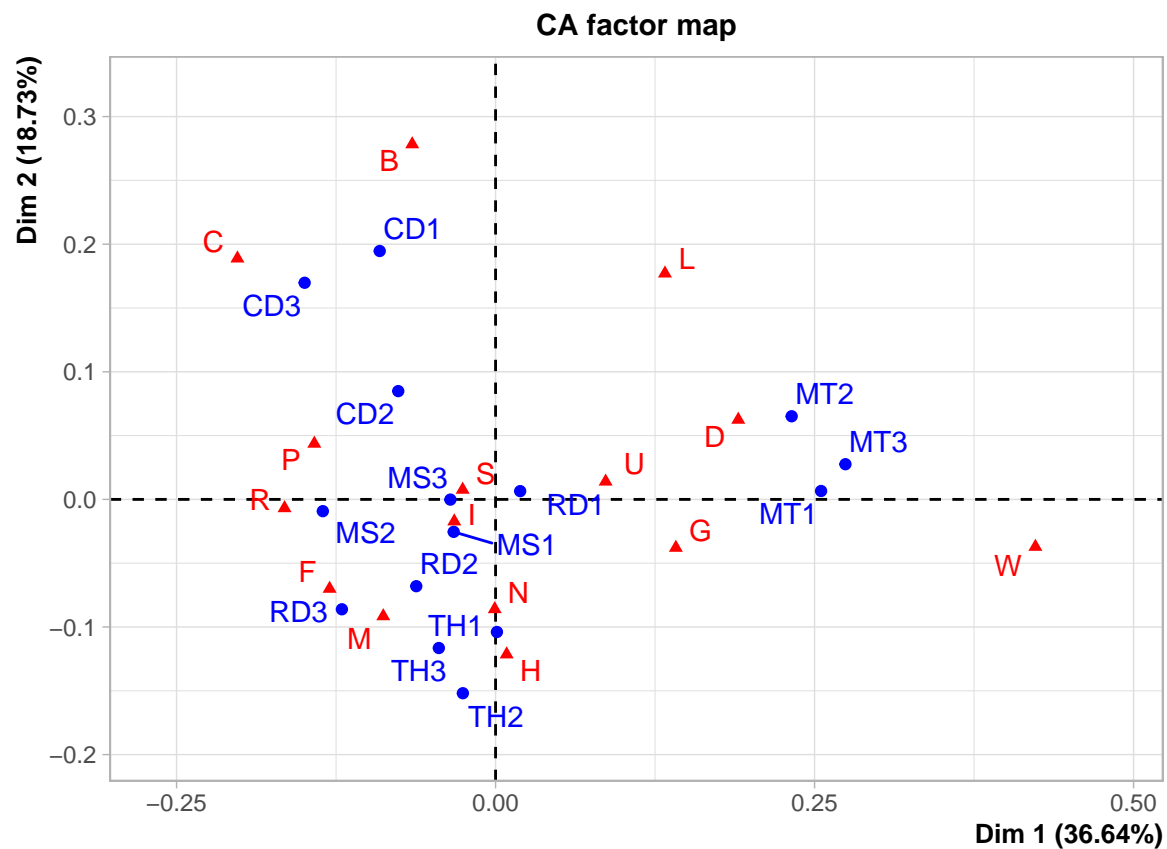
Le p-value est inferieur au seuil $\alpha = 0.05$ donc il y a une difference significative sur les distributions des lettres qui differe d'un echantillon a l'autre.

Realisation d'une ACP

```
caw1 <- CA(dataextr,graph = FALSE)
caw1$eig
```

```
## eigenvalue percentage of variance cumulative percentage of variance
## dim 1 1.819711e-02 36.64273828 36.64274
## dim 2 9.300360e-03 18.72773574 55.37047
## dim 3 7.320330e-03 14.74063391 70.11111
## dim 4 5.535310e-03 11.14621554 81.25732
## dim 5 3.666189e-03 7.38244803 88.63977
## dim 6 1.964005e-03 3.95483274 92.59460
## dim 7 1.561611e-03 3.14454947 95.73915
## dim 8 9.116786e-04 1.83580804 97.57496
## dim 9 6.636511e-04 1.33636565 98.91133
## dim 10 3.210046e-04 0.64639309 99.55772
## dim 11 1.415890e-04 0.28511167 99.84283
## dim 12 3.807211e-05 0.07666418 99.91950
## dim 13 2.206443e-05 0.04443019 99.96393
## dim 14 1.791440e-05 0.03607346 100.00000
```

```
plot(caw1)
```

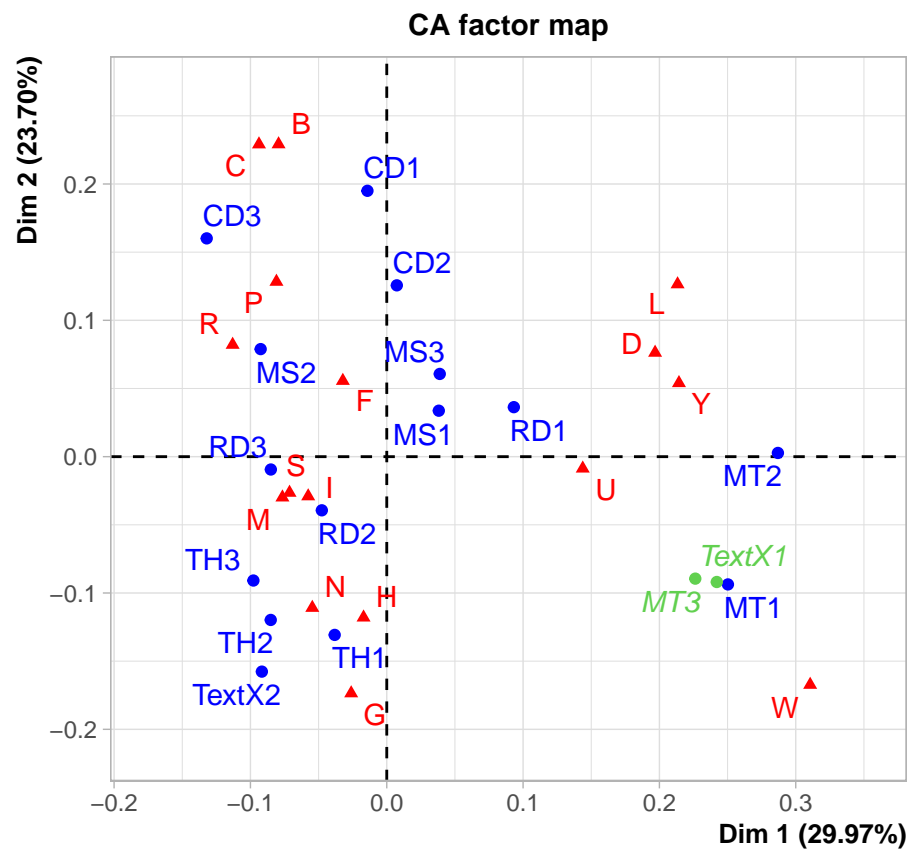


AFC en ajoutant les deux textes inconnus en lignes supplémentaires

```
caw2 <- CA(dataaw,row.sup=c(15,16),graph = FALSE)
caw2$eig
```

##	eigenvalue	percentage of variance	cumulative percentage of variance
## dim 1	1.450860e-02	29.970421699	29.97042
## dim 2	1.147339e-02	23.700583507	53.67101
## dim 3	6.721180e-03	13.883942471	67.55495
## dim 4	5.494277e-03	11.349528710	78.90448
## dim 5	4.558295e-03	9.416071315	88.32055
## dim 6	2.126041e-03	4.391763249	92.71231
## dim 7	1.441874e-03	2.978478544	95.69079
## dim 8	7.780333e-04	1.607183532	97.29797
## dim 9	5.629991e-04	1.162987380	98.46096
## dim 10	3.741861e-04	0.772956324	99.23392
## dim 11	2.224399e-04	0.459494134	99.69341
## dim 12	9.447666e-05	0.195160454	99.88857
## dim 13	5.306802e-05	0.109622626	99.99819
## dim 14	8.743066e-07	0.001806055	100.00000

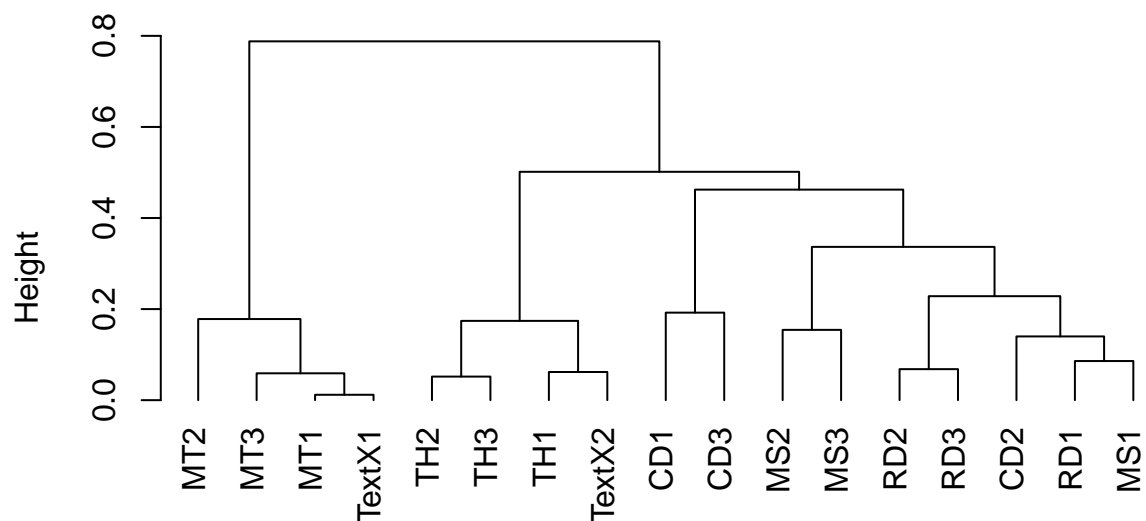
```
plot(caw2,col.row.sup=3)
```



Classification ascendante hiérarchique de Ward

```
#matrice des coordonnees factorielles sur 4 dimensionsX
mcf <- rbind(caw2$row$coord[,1:4],caw2$row.sup$coord[,1:4])
#matrice de distance euclidiennes entre les 17 echantillons
d <-dist(mcf)
#CAH
tree <-hclust(d,method="ward.D2")
plot(tree,hang=-1)
```

Cluster Dendrogram



d
hclust (*, "ward.D2")

```
#partition en 4 classes
cutree(tree,k=4)
```

##	CD1	CD2	CD3	RD1	RD2	RD3	TH1	TH2	TH3	MS1	MS2
##	1	2	1	2	2	2	3	3	3	2	2
##	MS3	MT1	MT2	TextX2	MT3	TextX1					
##	2	4	4	3	4	4					