



ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET D'ANALYSE DES  
SYSTÈMES - RABAT

---

*Rapport de projet d'analyse de données :*  
*Régression multiple, Classification, ACP et ACM*

---

*Réalisé par :*

Otmane LABIOUI  
Aymane ABABOU

*Encadré par :*

Pr. IBRAHIM JOUTEI IDRISSE  
AMRANI

*Année Scolaire 2021/2022*



# Table des matières

Introduction	0
1 Régression multiple	2
2 Classification	17
3 ACP	33
4 ACM	65

# Introduction

Les données concernant les maisons trouvées dans un district donné de California , et certains statistiques récapitulatifs à leur sujet basée sur les données du recensement de 1990.

Les colonnes sont les suivantes :

- 1) Médian House Value : valeur médiane de la maison pour les ménages dans un block (measured in US Dollars)
- 2) Median Income : revenue médiane des ménages dans un bloc de maison (measured in tens of thousands of US Dollars) [10k]
- 3) Median Age : Age médiane d'une maison dans un bloc ; un nombre inférieur est un bâtiment plus récente [ans]
- 4) Total Rooms : nombre total de chambre dans un bloc.
- 5) Total Bedrooms : nombre total de chambre à coucher dans un bloc.
- 6) Population : nombre total de personnes résidant dans un bloc.
- 7) Households : nombre total de ménages, un groupe de personne résidant dans une unité d'habitation pour un bloc.
- 10) Distance to coast : Distance jusqu'au point côtier le plus proche [m]
- 11) Distance to Los Angeles : Distance au centre de los angeles [m]
- 12) Distance to San Diego : Distance au centre de San Diego [m]
- 13) Distance to San Jose : Distance au centre de San Jose [m]
- 14) Distance to San Francisco : Distance au centre de San Francisco [m]

-Choix de la variable dépendante :

On a choisi la variable "Médian House Value" comme une variable dépendante expliquée par les autres variables, car le prix d'une maison dépend de la distance du centre de la ville ,nombre de chambre ....

# Chapitre 1

## Régression multiple

## Régression multiple

```
library("readxl")
data_calf<-read_excel("California_Houses.xlsx")

data_calf=transform(data_calf, Median_Income=as.numeric( Median_Income),
                    Distance_to_coast=as.numeric(Distance_to_coast),
                    Distance_to_LA=as.numeric(Distance_to_LA),
                    Distance_to_SanJose=as.numeric(Distance_to_SanJose),
                    Distance_to_SanFrancisco=as.numeric(Distance_to_SanFrancisco),
                    Distance_to_SanDiego=as.numeric(Distance_to_SanDiego))
summary(data_calf)
```

```
## Median_House_Value Median_Income      Median_Age      Tot_Rooms
## Min.   : 14999      Min.   : 0.4999  Min.   : 3.0    Min.   : 55
## 1st Qu.: 70000      1st Qu.: 2.0227  1st Qu.:18.0    1st Qu.: 1125
## Median :103100      Median : 2.8466  Median :25.0    Median : 1872
## Mean   :152960      Mean   : 3.2853  Mean   :27.2    Mean   : 2385
## 3rd Qu.:200000      3rd Qu.: 3.8438  3rd Qu.:35.0    3rd Qu.: 2602
## Max.   :500001      Max.   :11.1978  Max.   :52.0    Max.   :14316
## Tot_Bedrooms      Population      Households      Distance_to_coast
## Min.   : 11.0      Min.   : 26      Min.   : 5.0    Min.   : 2348
## 1st Qu.: 238.0      1st Qu.: 521      1st Qu.: 196.0  1st Qu.: 13126
## Median : 395.0      Median : 949      Median : 326.0  Median : 56047
## Mean   : 485.4      Mean   :1239      Mean   : 431.3  Mean   : 84719
## 3rd Qu.: 564.0      3rd Qu.:1387      3rd Qu.: 461.0  3rd Qu.:143795
## Max.   :2861.0      Max.   :7205      Max.   :2753.0  Max.   :311912
## Distance_to_LA      Distance_to_SanDiego Distance_to_SanJose
## Min.   : 8778      Min.   : 120489    Min.   : 17489
## 1st Qu.:166999      1st Qu.: 246754      1st Qu.:212558
## Median :517687      Median : 696446      Median :306467
## Mean   :439504      Mean   : 605150      Mean   :317672
## 3rd Qu.:709136      3rd Qu.: 887424      3rd Qu.:467729
## Max.   :960640      Max.   :1139711      Max.   :733835
## Distance_to_SanFrancisco
## Min.   : 16032
## 1st Qu.:205473
## Median :303884
## Mean   :327671
## 3rd Qu.:496637
## Max.   :801545
```

###1) Calcule du modèle de régression linéaire multiple incluant toute les variables explicatives

```
regression_multiple<-lm(Median_House_Value ~ Median_Income + Median_Age + Tot_Rooms + Tot_Bedrooms +
                        Population+ Households + Distance_to_coast + Distance_to_LA +
```

```
Distance_to_SanFrancisco,data=data_calf)
summary(regression_multiple)
```

```
##
## Call:
## lm(formula = Median_House_Value ~ Median_Income + Median_Age +
##      Tot_Rooms + Tot_Bedrooms + Population + Households + Distance_to_coast +
##      Distance_to_LA + Distance_to_SanDiego + Distance_to_SanJose +
##      Distance_to_SanFrancisco, data = data_calf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -104649  -35557   -4905   18872  184550
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    48786.4038  54687.6014   0.892  0.37455
## Median_Income    45943.9619   4730.5802   9.712 5.61e-16 ***
## Median_Age         52.3884    566.7750   0.092  0.92655
## Tot_Rooms        -28.6621    11.4707  -2.499  0.01415 *
## Tot_Bedrooms     252.0989     88.0077   2.865  0.00512 **
## Population       -67.8824    22.6231  -3.001  0.00343 **
## Households        98.0291    112.4860   0.871  0.38565
## Distance_to_coast  -0.2845     0.0873  -3.259  0.00154 **
## Distance_to_LA    -0.4119     0.1523  -2.705  0.00807 **
## Distance_to_SanDiego  0.2475     0.1433   1.728  0.08719 .
## Distance_to_SanJose  0.5486     0.3144   1.745  0.08418 .
## Distance_to_SanFrancisco -0.5472     0.3156  -1.734  0.08612 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57060 on 97 degrees of freedom
## Multiple R-squared:  0.8073, Adjusted R-squared:  0.7854
## F-statistic: 36.93 on 11 and 97 DF,  p-value: < 2.2e-16
```

###1-1) Existe-t-il des variables explicatives non significatives ? D'après le résultat de la commande `summary(regression_multiple)`, les variables explicatives non significatives (elles ont  $p\text{-value} > 0.05 = \alpha$ ) sont de 5 variables, elles sont `Median_Age`, `Households`, `Distance_to_SanDiego`, `Distance_to_SanJose`, `Distance_to_SanFrancisco`.

###1-2) Donner la valeur de  $R$  et  $R_{\text{ajuste}}$ : D'après le résultat de la régression on a :  $R^2 = 0.8073$   
 $R^2_{\text{ajuste}} = 0.7854$

###1-3) le test de Fisher ? Le test de Fisher est significatif car la statistique  $F=36.93$  est grande et la probabilité critique associée au test  $p\text{-value} < 0.05$ , ce test signifie qu'il existe au moins une variable significativement non nulle.

###2) Amélioration du modèle initial par la procédure `step`

```
reg_ameliore=step(regression_multiple)
```

```
## Start:  AIC=2398.77
## Median_House_Value ~ Median_Income + Median_Age + Tot_Rooms +
##      Tot_Bedrooms + Population + Households + Distance_to_coast +
```

```
## Distance_to_LA + Distance_to_SanDiego + Distance_to_SanJose +
## Distance_to_SanFrancisco
##
##          Df Sum of Sq      RSS      AIC
## - Median_Age          1 2.7813e+07 3.1580e+11 2396.8
## - Households          1 2.4723e+09 3.1824e+11 2397.6
## <none>                                3.1577e+11 2398.8
## - Distance_to_SanDiego      1 9.7190e+09 3.2549e+11 2400.1
## - Distance_to_SanFrancisco  1 9.7865e+09 3.2555e+11 2400.1
## - Distance_to_SanJose      1 9.9105e+09 3.2568e+11 2400.1
## - Tot_Rooms                1 2.0325e+10 3.3609e+11 2403.6
## - Distance_to_LA           1 2.3816e+10 3.3958e+11 2404.7
## - Tot_Bedrooms            1 2.6711e+10 3.4248e+11 2405.6
## - Population              1 2.9309e+10 3.4508e+11 2406.4
## - Distance_to_coast        1 3.4569e+10 3.5034e+11 2408.1
## - Median_Income           1 3.0706e+11 6.2283e+11 2470.8
##
## Step: AIC=2396.78
## Median_House_Value ~ Median_Income + Tot_Rooms + Tot_Bedrooms +
## Population + Households + Distance_to_coast + Distance_to_LA +
## Distance_to_SanDiego + Distance_to_SanJose + Distance_to_SanFrancisco
##
##          Df Sum of Sq      RSS      AIC
## - Households          1 2.5696e+09 3.1837e+11 2395.7
## <none>                                3.1580e+11 2396.8
## - Distance_to_SanFrancisco  1 1.0456e+10 3.2625e+11 2398.3
## - Distance_to_SanJose      1 1.0644e+10 3.2644e+11 2398.4
## - Distance_to_SanDiego      1 1.0654e+10 3.2645e+11 2398.4
## - Tot_Rooms                1 2.0307e+10 3.3610e+11 2401.6
## - Tot_Bedrooms            1 2.7775e+10 3.4357e+11 2404.0
## - Distance_to_LA           1 2.7857e+10 3.4365e+11 2404.0
## - Population              1 2.9523e+10 3.4532e+11 2404.5
## - Distance_to_coast        1 3.4705e+10 3.5050e+11 2406.2
## - Median_Income           1 3.0746e+11 6.2325e+11 2468.9
##
## Step: AIC=2395.67
## Median_House_Value ~ Median_Income + Tot_Rooms + Tot_Bedrooms +
## Population + Distance_to_coast + Distance_to_LA + Distance_to_SanDiego +
## Distance_to_SanJose + Distance_to_SanFrancisco
##
##          Df Sum of Sq      RSS      AIC
## <none>                                3.1837e+11 2395.7
## - Distance_to_SanDiego      1 1.0585e+10 3.2895e+11 2397.2
## - Distance_to_SanFrancisco  1 1.1083e+10 3.2945e+11 2397.4
## - Distance_to_SanJose      1 1.1192e+10 3.2956e+11 2397.4
## - Tot_Rooms                1 1.8738e+10 3.3710e+11 2399.9
## - Distance_to_LA           1 2.8087e+10 3.4645e+11 2402.9
## - Population              1 3.2215e+10 3.5058e+11 2404.2
## - Distance_to_coast        1 3.8430e+10 3.5680e+11 2406.1
## - Tot_Bedrooms            1 7.9254e+10 3.9762e+11 2417.9
## - Median_Income           1 3.1971e+11 6.3808e+11 2469.4

modele<-lm(Median_House_Value ~ Median_Income + Tot_Rooms +
          Tot_Bedrooms + Population + Distance_to_coast + Distance_to_LA +
```



```

        Distance_to_SanDiego + Distance_to_SanJose + Distance_to_SanFrancisco,
        data = data_calf)
summary(modele)

```

```

##
## Call:
## lm(formula = Median_House_Value ~ Median_Income + Tot_Rooms +
##     Tot_Bedrooms + Population + Distance_to_coast + Distance_to_LA +
##     Distance_to_SanDiego + Distance_to_SanJose + Distance_to_SanFrancisco,
##     data = data_calf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -93289 -36920  -3831   22080 184250
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.024e+04  5.017e+04   1.001 0.319074
## Median_Income    4.649e+04  4.662e+03   9.971 < 2e-16 ***
## Tot_Rooms       -2.722e+01  1.128e+01  -2.414 0.017622 *
## Tot_Bedrooms     3.033e+02  6.109e+01   4.964 2.87e-06 ***
## Population      -5.547e+01  1.753e+01  -3.165 0.002060 **
## Distance_to_coast -2.964e-01  8.573e-02  -3.457 0.000807 ***
## Distance_to_LA    -4.185e-01  1.416e-01  -2.955 0.003903 **
## Distance_to_SanDiego  2.500e-01  1.378e-01   1.814 0.072666 .
## Distance_to_SanJose   5.683e-01  3.046e-01   1.866 0.065062 .
## Distance_to_SanFrancisco -5.687e-01  3.064e-01  -1.856 0.066361 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56710 on 99 degrees of freedom
## Multiple R-squared:  0.8057, Adjusted R-squared:  0.788
## F-statistic: 45.6 on 9 and 99 DF, p-value: < 2.2e-16

```

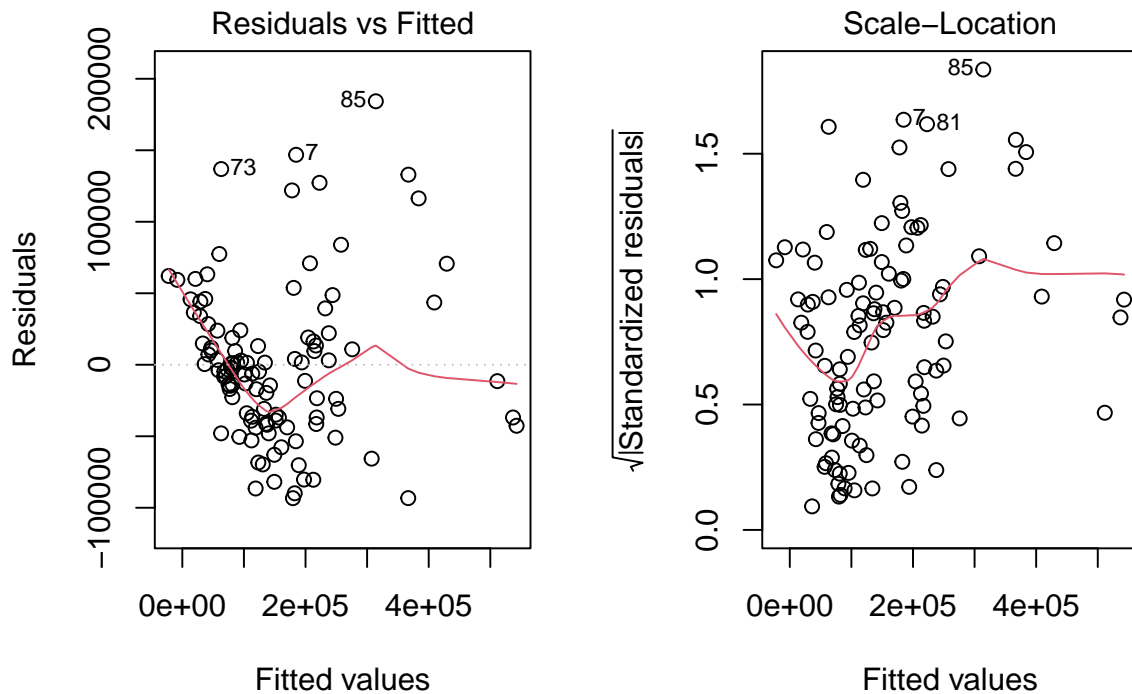
Ce modèle amélioré laisse 9 variables explicatives parmi 12, il a pris les variables qu'ont p-value moins du seuil de significativité alpha(5%), ou une p-value petite même si elle est supérieure à 5%. Après cette amélioration, nous avons remarqué que la valeur de  $R^2_{\text{ajuste}}$  à passer de 0.7854 à 0.788, c'est à dire que le modèle explique environ 79% des données, ce qui signifie qu'il y a une amélioration au niveau du modèle

###2-1) les tests de validation pour le modèle amélioré de la procédure de step : - test d'homoscédasticité

```

par(mfrow = c(1, 2))
plot(reg_ameliore,1)
plot(reg_ameliore,3)

```



On peut déduire à partir des deux graphes au-dessus que l'hypothèse d'homoscédasticité n'est pas vérifiée car la ligne en rouge n'est pas horizontale, il n'y a pas une forme distinguée.

- test de normalité (shapiro et ks)

```
#test shapiro
shapiro.test(reg_amelioire$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  reg_amelioire$residuals
## W = 0.94677, p-value = 0.0002697

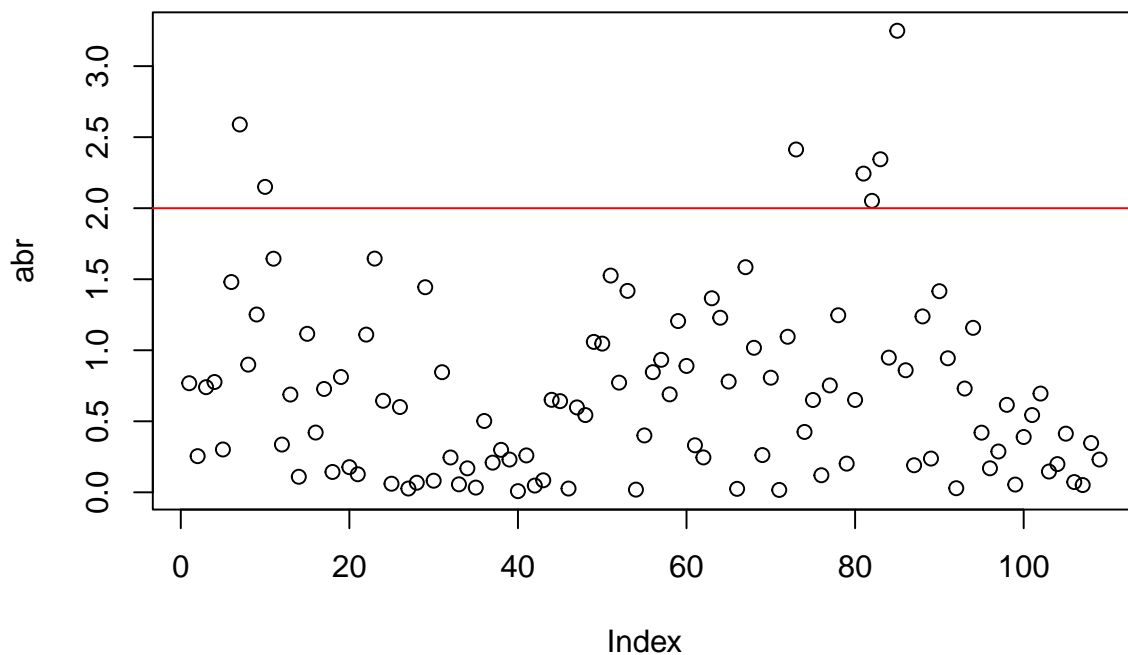
#test ks
ks.test(reg_amelioire$residuals,"pnorm")

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  reg_amelioire$residuals
## D = 0.53211, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Les deux tests montrent que l'hypothèse de normalité des résidus n'est pas vérifiée car l'hypothèse nulle (on a une distribution normale) est significativement rejetée ( $p\text{-value} < \alpha = 5\%$ ).

- recherche de valeurs aberrantes

```
rse=sqrt(deviance(reg_ameliore)/df.residual(reg_ameliore))
abr=abs(data_calf$Median_House_Value-predict(reg_ameliore))/rse
plot(abr)
abline(h=2,col='red')
```



Les valeurs qui sont au dessus de la ligne en rouge sont des valeurs aberrantes. il existe 7 valeurs aberrantes. ###3) la méthode pas à pas de sélection des variables explicatives, basée sur le test de Fisher : ####Etapel On commence la méthode pas à pas par l'intégration de la variable la plus significative (F le plus grand)

```
nva=ncol(data_calf)
Fish = rep(0,nva)
for (i in 2:ncol(data_calf)){
  mod1<-lm(data_calf[,1]~data_calf[,i])
  Fish[i]=var(predict(mod1))*(nrow(data_calf)-1)/(deviance(mod1)/df.residual(mod1))
}
Fish
```

```
## [1] 0.000000 185.733089 2.817161 21.371567 16.104678 12.029662
## [7] 18.448943 38.805635 25.730663 20.843943 1.003932 2.924802
```

```
df2=nrow(data_calf)-2
df2
```

```
## [1] 107
```

```
1-pf(max(Fish),1,df2)
```

```
## [1] 0
```

c'est la variable Mediane\_Age qui a le plus grand F #etape2: Introduction

```
nva=ncol(data_calf)-1
Fish = rep(0,nva)
SCR1<-deviance(lm(data_calf[,1]~data_calf[,2]))
for (i in 3:ncol(data_calf)) {
  mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,i])
  SCR2=deviance(mod)
  Fish[i]=(SCR1-SCR2)/(SCR2/(nrow(data_calf)-3))
}
Fish
```

```
## [1] 0.0000000 0.0000000 0.7224821 0.4271976 4.2217869 1.1999758
## [7] 3.2264290 23.0383218 11.4026051 9.0285551 1.9909412 2.9855053
```

```
df2=nrow(data_calf)-3
df2
```

```
## [1] 106
```

```
1-pf(max(Fish),1,df2)
```

```
## [1] 5.231617e-06
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = data_calf[, 1] ~ data_calf[, 2] + data_calf[, i])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -188702  -36707  -10280   27464  282272
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.437e+04  1.818e+04  -1.890   0.0615 .
## data_calf[, 2]  5.047e+04  3.718e+03  13.573   <2e-16 ***
## data_calf[, i]  6.567e-02  3.801e-02   1.728   0.0869 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74130 on 106 degrees of freedom
## Multiple R-squared:  0.6445, Adjusted R-squared:  0.6378
## F-statistic: 96.08 on 2 and 106 DF, p-value: < 2.2e-16
```

c'est la variable Distance\_to\_coast qui a le plus grand F #Etape2: Retrait

```
Fish = rep(0,2)
SCR2=deviance(lm(data_calf[,1]~data_calf[,2]+data_calf[,8]))
mod<-lm(data_calf[,1]~data_calf[,2])
SCR1<-deviance(mod)
Fish[1]=(SCR1-SCR2)/(SCR2/(nrow(data_calf)-3))

mod<-lm(data_calf[,1]~data_calf[,8])
SCR1=deviance(mod)
Fish[2]=(SCR1-SCR2)/(SCR2/(nrow(data_calf)-3))
Fish
```

```
## [1] 23.03832 153.06946
```

```
df2=nrow(data_calf)-3
df2
```

```
## [1] 106
```

```
1-pf(min(Fish),1,df2)
```

```
## [1] 5.231617e-06
```

aucun variable n'est retire , les F sont significatifs

#Etape3:Introduction

```
Fish = rep(0,nva)
SCR2<-deviance(lm(data_calf[,1]~data_calf[,2]+data_calf[,8]))
for (i in 3:7) {
  mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,i])
  SCR3=deviance(mod)
  Fish[i]=(SCR2-SCR3)/(SCR3/(nrow(data_calf)-4))
}
for (i in 9:ncol(data_calf)) {
  mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,i])
  SCR3=deviance(mod)
  Fish[i]=(SCR2-SCR3)/(SCR3/(nrow(data_calf)-4))
}
Fish
```

```
## [1] 0.0000000 0.0000000 0.2475582 0.3182474 3.1179696 0.3437354 1.8004046
## [8] 0.0000000 8.2010610 5.9533593 1.7993070 2.8854312
```

```
df2=nrow(data_calf)-4
df2
```

```
## [1] 105
```

```
1-pf(max(Fish),1,df2)
```

```
## [1] 0.005056086
```

c'est la variable Distance\_to\_LA qui a le plus grand F #Etape3: Retrait

```
SCR3<-deviance(lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,9]))
Fish<-rep(0,3)
mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,8])
SCR2<-deviance(mod)
Fish[1]=(SCR2-SCR3)/(SCR3/(nrow(data_calf)-4))

mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,9])
SCR2<-deviance(mod)
Fish[2]=(SCR2-SCR3)/(SCR3/(nrow(data_calf)-4))

mod<-lm(data_calf[,1]~data_calf[,8]+data_calf[,9])
SCR2<-deviance(mod)
Fish[3]=(SCR2-SCR3)/(SCR3/(nrow(data_calf)-4))

Fish
```

```
## [1] 8.201061 19.420365 134.135425
```

```
df2=nrow(data_calf)-4
df2
```

```
## [1] 105
```

```
1-pf(min(Fish),1,df2)
```

```
## [1] 0.005056086
```

aucun varibale n'est retire , les F sont significatifs #Etape 4: Introduction

```
Fish = rep(0,nva)
SCR3<-deviance(lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,9]))
for (i in 3:7) {
  mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,9]+data_calf[,i])
  SCR4=deviance(mod)
  Fish[i]=(SCR3-SCR4)/(SCR4/(nrow(data_calf)-4))
}
for (i in 10:ncol(data_calf)) {
  mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,9]+data_calf[,i])
  SCR4=deviance(mod)
  Fish[i]=(SCR3-SCR4)/(SCR4/(nrow(data_calf)-4))
}
Fish
```

```
## [1] 0.000000000 0.000000000 0.003104573 0.554262148 3.002698750 0.151122444
## [7] 1.646671456 0.000000000 0.000000000 5.969659075 0.076532304 0.001453928
```

```
df2=nrow(data_calf)-4
df2
```

```
## [1] 105
```

```
1-pf(max(Fish),1,df2)
```

```
## [1] 0.01622177
```

la variable Distance\_to\_SanDiego a le plus grand F #Etape4: Retrait

```
SCR4<-deviance(lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,9]+data_calf[,10]))
Fish<-rep(0,4)
mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,9])
SCR3<-deviance(mod)
Fish[1]=(SCR3-SCR4)/(SCR4/(nrow(data_calf)-4))
```

```
mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,10])
SCR3<-deviance(mod)
Fish[2]=(SCR3-SCR4)/(SCR4/(nrow(data_calf)-4))
```

```
mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,9]+data_calf[,10])
SCR3<-deviance(mod)
Fish[3]=(SCR3-SCR4)/(SCR4/(nrow(data_calf)-4))
```

```
mod<-lm(data_calf[,1]~data_calf[,8]+data_calf[,9]+data_calf[,10])
SCR3<-deviance(mod)
Fish[4]=(SCR3-SCR4)/(SCR4/(nrow(data_calf)-4))
```

```
Fish
```

```
## [1] 5.969659 8.217691 21.558516 128.790121
```

```
df2=nrow(data_calf)-4
df2
```

```
## [1] 105
```

```
1-pf(min(Fish),1,df2)
```

```
## [1] 0.01622177
```

aucun variable n'est retire

#Etape 5: Introduction

```

Fish = rep(0,nva)
SCR4<-deviance(lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,9]+data_calf[,10]))
for (i in 3:7) {
  mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,9]+data_calf[,10]+data_calf[,i])
  SCR5=deviance(mod)
  Fish[i]=(SCR4-SCR5)/(SCR5/(nrow(data_calf)-4))
}
Fish[8]=Fish[9]=Fish[10]=0
for (i in 11:ncol(data_calf)) {
  mod<-lm(data_calf[,1]~data_calf[,2]+data_calf[,8]+data_calf[,9]+data_calf[,10]+data_calf[,i])
  SCR5=deviance(mod)
  Fish[i]=(SCR4-SCR5)/(SCR5/(nrow(data_calf)-4))
}
Fish

```

```

## [1] 0.000000000 0.000000000 0.154583929 0.273997739 2.171374529 0.005241776
## [7] 1.001456412 0.000000000 0.000000000 0.000000000 2.607248618 1.721263302

```

```

df2=nrow(data_calf)-4
df2

```

```

## [1] 105

```

```

1-pf(max(Fish),1,df2)

```

```

## [1] 0.1093773

```

LA variable Distance\_to\_SanJose a le plus grand F mais avec un p-valu>10% c'est la condition d'arret

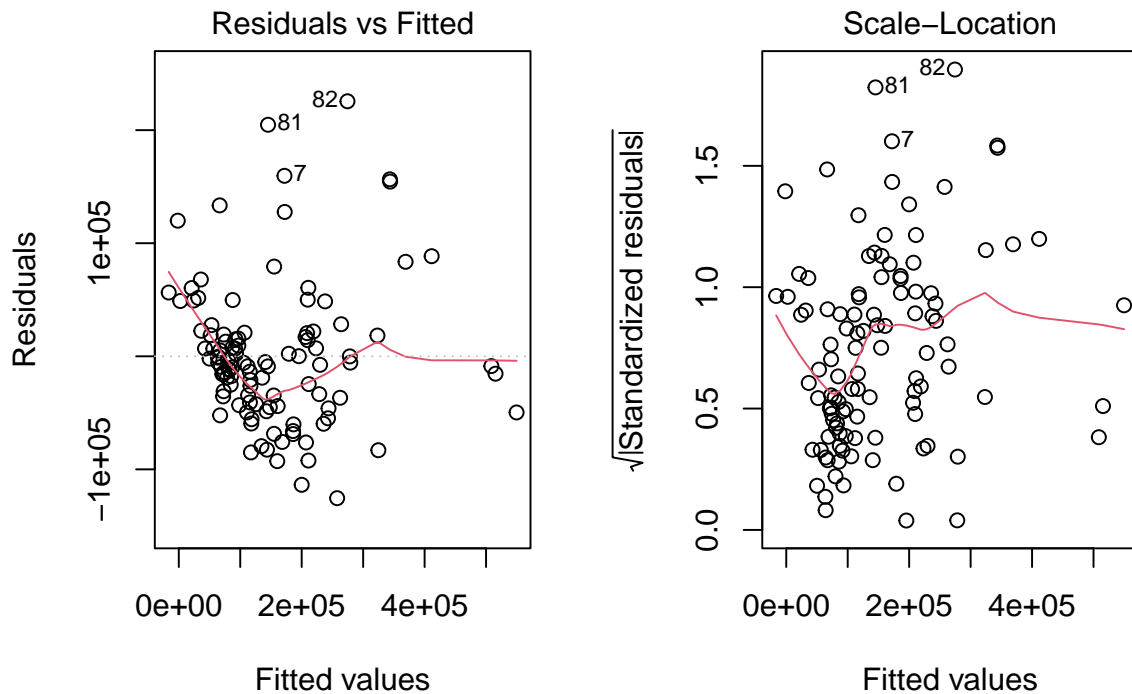
###3-1)Les tests de validation pour le modèle obtenu: - test d'homoscédasticité

```

model2<-lm(Median_House_Value ~ Median_Income + Distance_to_coast + Distance_to_LA +
           Distance_to_SanDiego + Distance_to_SanJose ,
           data = data_calf)
par(mfrow = c(1, 2))
plot(model2,1)
plot(model2,3)

```





on observe que l'hypothèse d'homoscédasticité n'est pas vérifiée car la courbe en rouge n'est pas horizontale.  
- test de normalité

```
#test shapiro
shapiro.test(model2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model2$residuals
## W = 0.92188, p-value = 7.964e-06
```

```
#test ks
ks.test(model2$residuals,"pnorm")
```

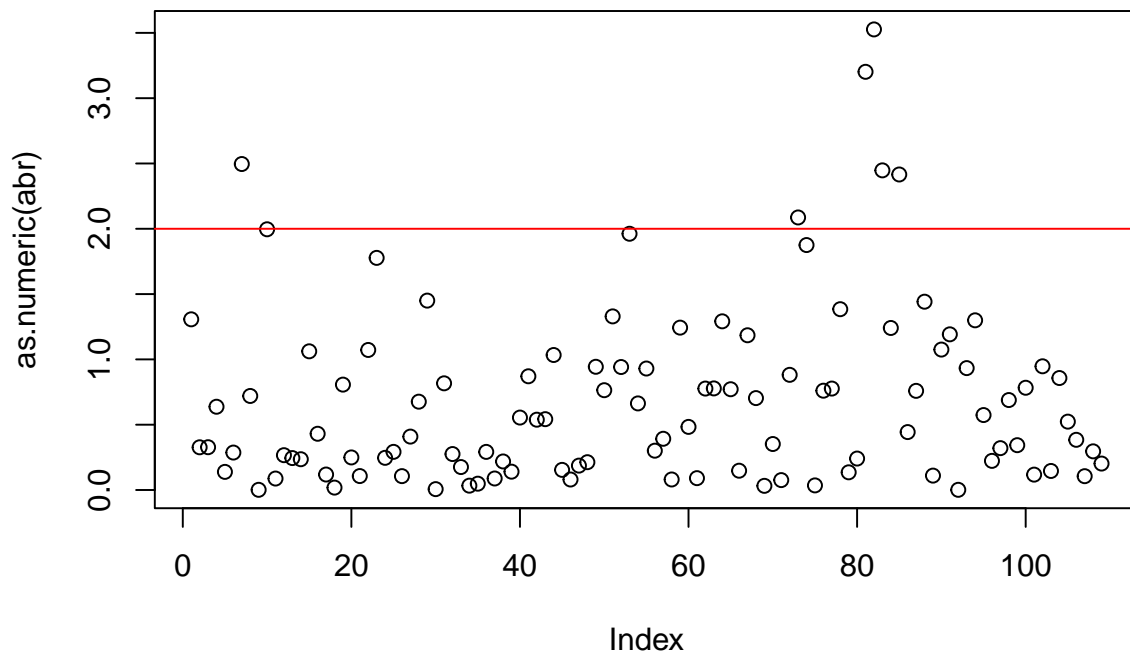
```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  model2$residuals
## D = 0.58716, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

on déduit que l'hypothèse null de normalité est rejetée, donc la distribution des résidus n'est pas une distribution normale (Gaussien) test de valeurs aberrantes

```

rse=sqrt(deviance(model2)/df.residual(model2))
abr=abs(data_calf$Median_House_Value-predict(model2))/rse
plot(as.numeric(abr))
abline(h=2,col='red')

```



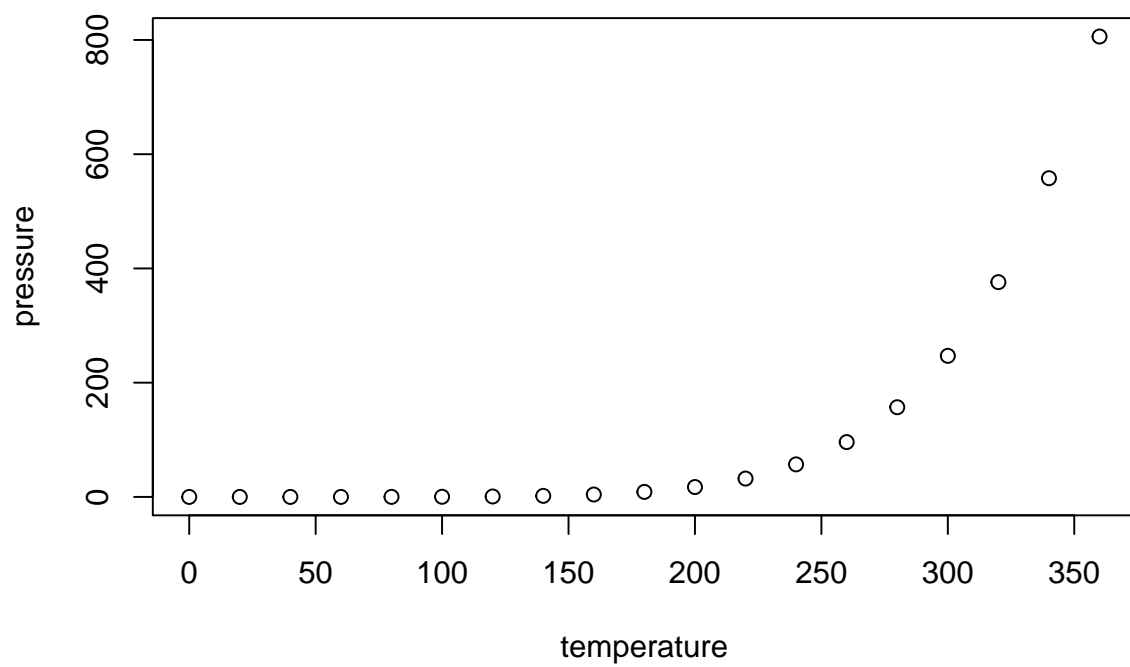
###3-2) le critère AIC du modèle obtenu par cette méthode

```
AIC(model2)
```

```
## [1] 2729.524
```

###3-3) Conclusion On constate que le premier amélioration de notre modèle de regression qu'est fait par la procédure step est mieux que la deuxième amélioration, car la valeur AIC du premier inférieure à la deuxième et pour le Adjusted R-squared de premier amélioration est un peu grand par rapport la deuxième, et ça implique que le premier modèle amélioré explique bien la valeur dépendante que le deuxième modèle amélioré.

Après la sélection des variables qui contribuent plus dans l'explication du variable dépendante , on peut conclure que le modèle obtenu est relativement valide car il est homogène, la normalité de ses résidus est acceptée par le test de Shapiro et son AIC est minimale. Le modèle obtenue après la sélection des variables est meilleur que le premier modèle obtenue avec toutes les variables d'où la sélection des variables explicatives est très utile pour l'amélioration du modèle du départ. Après la comparaison des deux modèles obtenus par la procédure step et par la sélection des variables, nous avons remarqué que le modèle obtenu par la procédure step est meilleur que le modèle obtenu par la sélection des variables car il est homogène ,il a moins de valeurs aberrante et il a aussi la minimale valeur de AIC(aic=423.24). D'après cette comparaison, on peut conclure que l'efficacité de la procédure step



## Chapitre 2

# Classification

# Classification

ababou et labioui

16/01/2022

###lecture de donnée

```
library("readxl")
data_calf<-read_excel("California_Houses.xlsx")
data_calf=transform(data_calf[, -1], Median_Income=as.numeric( Median_Income),Distance_to_coast=as.numeric(
Distance_to_LA=as.numeric(Distance_to_LA),Distance_to_SanJose=as.numeric(Distance_to
Distance_to_SanFrancisco=as.numeric(Distance_to_SanFrancisco),Distance_to_SanDiego=
summary(data_calf)
```

```
## Median_Income      Median_Age      Tot_Rooms      Tot_Bedrooms
## Min.      : 0.4999    Min.      : 3.0    Min.      : 55    Min.      : 11.0
## 1st Qu.: 2.0227    1st Qu.:18.0    1st Qu.: 1125    1st Qu.: 238.0
## Median : 2.8466    Median :25.0    Median : 1872    Median : 395.0
## Mean   : 3.2853    Mean   :27.2    Mean   : 2385    Mean   : 485.4
## 3rd Qu.: 3.8438    3rd Qu.:35.0    3rd Qu.: 2602    3rd Qu.: 564.0
## Max.   :11.1978    Max.   :52.0    Max.   :14316    Max.   :2861.0
## Population      Households      Distance_to_coast Distance_to_LA
## Min.      : 26    Min.      : 5.0    Min.      : 2348    Min.      : 8778
## 1st Qu.: 521    1st Qu.: 196.0    1st Qu.: 13126    1st Qu.:166999
## Median : 949    Median : 326.0    Median : 56047    Median :517687
## Mean   :1239    Mean   : 431.3    Mean   : 84719    Mean   :439504
## 3rd Qu.:1387    3rd Qu.: 461.0    3rd Qu.:143795    3rd Qu.:709136
## Max.   :7205    Max.   :2753.0    Max.   :311912    Max.   :960640
## Distance_to_SanDiego Distance_to_SanJose Distance_to_SanFrancisco
## Min.      : 120489    Min.      : 17489    Min.      : 16032
## 1st Qu.: 246754    1st Qu.:212558    1st Qu.:205473
## Median : 696446    Median :306467    Median :303884
## Mean   : 605150    Mean   :317672    Mean   :327671
## 3rd Qu.: 887424    3rd Qu.:467729    3rd Qu.:496637
## Max.   :1139711    Max.   :733835    Max.   :801545
```

###1) Appliquer kmeans au tableau des variables quantitatives, le nombre de classes va varier de 1 à N et les variables doivent être centrées et réduites. N étant le plus petit entier tel que le taux d'inertie expliquée de la classification à N classes est supérieur à 0.95 centrage et reduction des variables

```
data_calf.cr<-scale(data_calf,center=T,scale=T)
summary(data_calf.cr)
```

```
## Median_Income      Median_Age      Tot_Rooms      Tot_Bedrooms
## Min.      : -1.4474    Min.      : -1.9716    Min.      : -0.97464    Min.      : -1.0841
```

```
## 1st Qu.: -0.6561 1st Qu.: -0.7496 1st Qu.: -0.52714 1st Qu.: -0.5653
## Median : -0.2280 Median : -0.1794 Median : -0.21473 Median : -0.2065
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.0000
## 3rd Qu.: 0.2902 3rd Qu.: 0.6353 3rd Qu.: 0.09057 3rd Qu.: 0.1797
## Max. : 4.1118 Max. : 2.0202 Max. : 4.98962 Max. : 5.4290
## Population Households Distance_to_coast Distance_to_LA
## Min. : -1.0064 Min. : -1.00838 Min. : -1.0403 Min. : -1.3939
## 1st Qu.: -0.5956 1st Qu.: -0.55655 1st Qu.: -0.9042 1st Qu.: -0.8819
## Median : -0.2405 Median : -0.24902 Median : -0.3621 Median : 0.2530
## Mean : 0.0000 Mean : 0.00000 Mean : 0.0000 Mean : 0.0000
## 3rd Qu.: 0.1230 3rd Qu.: 0.07034 3rd Qu.: 0.7461 3rd Qu.: 0.8726
## Max. : 4.9510 Max. : 5.49230 Max. : 2.8694 Max. : 1.6865
## Distance_to_SanDiego Distance_to_SanJose Distance_to_SanFrancisco
## Min. : -1.5007 Min. : -1.86693 Min. : -1.6554
## 1st Qu.: -1.1097 1st Qu.: -0.65374 1st Qu.: -0.6491
## Median : 0.2827 Median : -0.06969 Median : -0.1264
## Mean : 0.0000 Mean : 0.00000 Mean : 0.0000
## 3rd Qu.: 0.8740 3rd Qu.: 0.93325 3rd Qu.: 0.8975
## Max. : 1.6552 Max. : 2.58824 Max. : 2.5171
```

application de k\_means au notre jeu de données:

-détermination du nombre des classes qui donne le taux d'inertie de 95%

```
inertie.expl <- rep(0,times=35)
k=2
clus <- kmeans(data_calf.cr,centers=k,nstart=5)
inertie.expl[k] <- clus$betweenss/clus$totss
while(inertie.expl[k]<0.95){
  k=k+1
  clus <- kmeans(data_calf.cr,centers=k,nstart=5)
  inertie.expl[k] <- clus$betweenss/clus$totss
}
max(inertie.expl)
```

```
## [1] 0.9516308
```

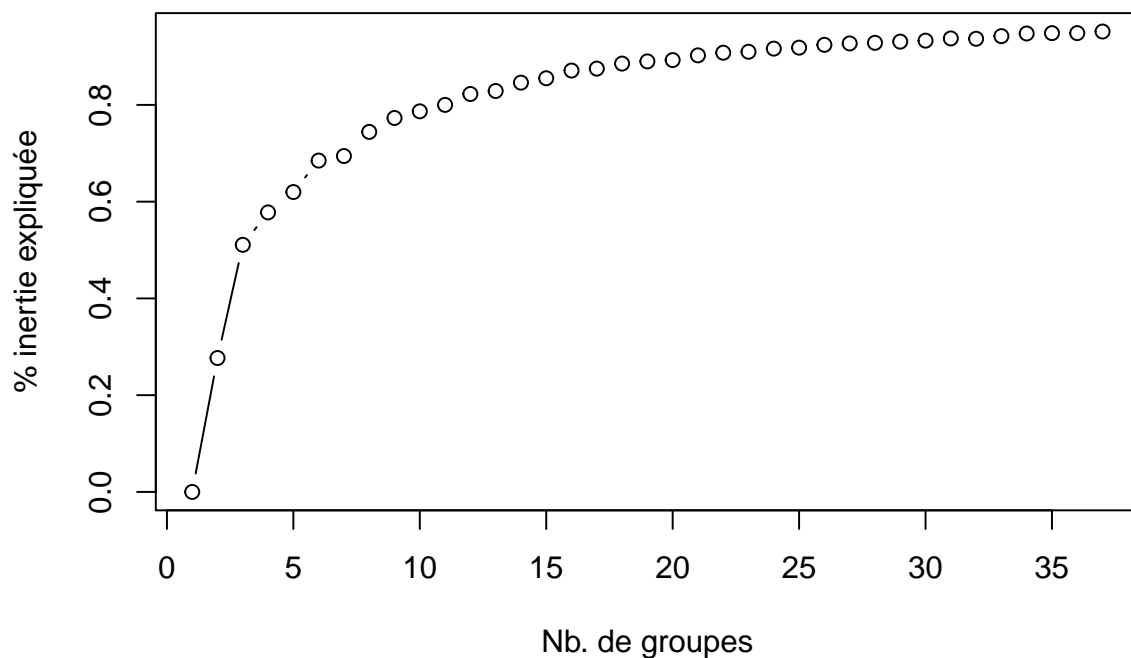
```
k-1
```

```
## [1] 36
```

N=35 est le plus petit entier tel que  $\max(\text{inertie.expl}) > 0.95$ , donc le Taux d'inertie expliquée nous a aidé à valider le choix du nombre de class adequat pour que la variace intra class soit minimal et dans l'autre cote la variance inter class soit maximal

###2)Détermination de Nc le nombre de classes à retenir en utilisant la méthode :  $\text{var}(I_2)/\text{var}(I) < 0,05$ , I étant le vecteur de taille N des taux d'inertie expliquée et I\_2 étant le vecteur des (N-Nc) dernières valeurs des taux d'inertie expliquée - graphe presente inertie explique en fonction de nombre de groupe

```
plot(1:k,inertie.expl,type="b",xlab="Nb. de groupes",ylab="% inertie expliquée")
```



- determination du Nc

```
n=1
regle=var(na.omit(inertie.expl[n+1:k]))*(k-n-1)*100/(var(inertie.expl)*(k-1))
while(regle>5){
  regle=var(na.omit(inertie.expl[n+1:k]))*(k-n-1)*100/(var(inertie.expl)*(k-1))
  n=n+1
}
n
```

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

```
regle
```

```
## [1] 4.159994
```

A partir de la valeur 3.897867 de li'inertie explique,l'adjonction d'un groupe supplémentaire n'augmente pas significativement la part d'inertie expliquée par la partition ,cela traduit par un nombre de class à retenir egal à 10 .

###3) Faire une CAH sur le tableau des variables quantitatives. Les variables doivent au préalable être centrées et réduites.

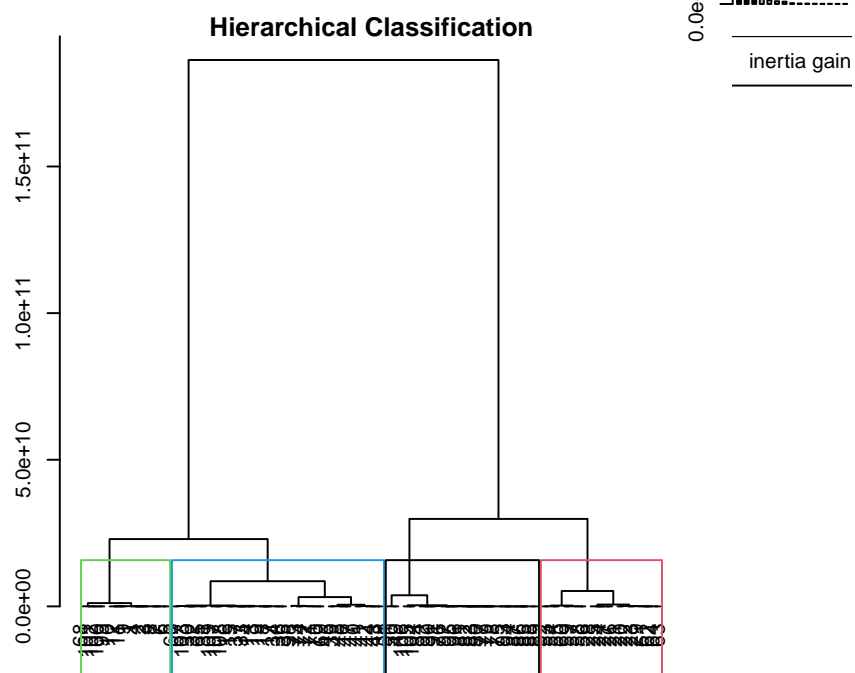
```
library(FactoMineR)
#Les données ont été préalablement centrées et réduites sur Excel
R.scale <- function(Y){
```

```

#centrage réduction d'une variable - sans utilisation de boucle for
X<-Y
for (k in 1:ncol(X)){
  i<-1
  n <- length(X[,k])
  moy <- mean(X[,k])
  et <- sqrt((n-1)/n*var(X[,k]))
  while(i<nrow(X)){
    X[i,k] <- (X[i,k]-moy)/et
    i<-i+1
  }
}
return(Y)
}
Y=R.scale(data_calf)
#CAH
Res<-HCPC(as.data.frame(data_calf),nb.clust=-1)

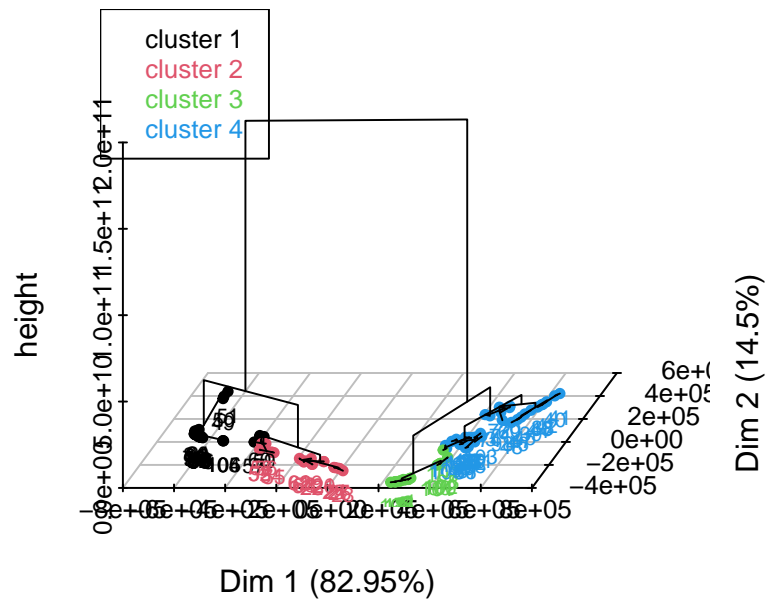
```

## Hierarchical Clustering

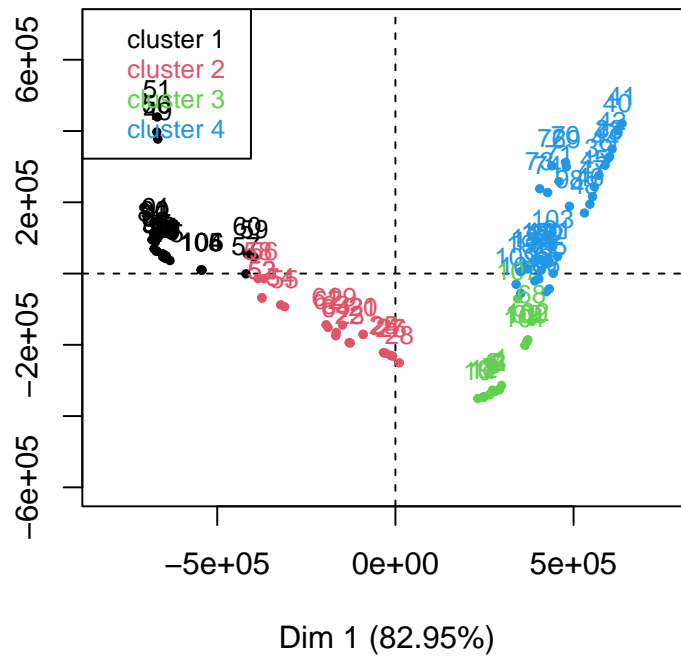




## Hierarchical clustering on the factor map



Factor map



Res\$data.clust

##	Median_Income	Median_Age	Tot_Rooms	Tot_Bedrooms	Population	Households
## 1	8.3252	41	880	129	322	126
## 2	1.9028	51	1590	414	949	392
## 3	2.1250	47	1372	395	1237	303
## 4	2.1000	51	175	43	228	55
## 5	2.0227	44	1097	239	609	215
## 6	7.2634	12	14316	2045	5781	2007
## 7	3.4861	18	4335	808	2041	734
## 8	5.2312	31	1487	280	854	301
## 9	6.1047	7	10648	1818	6075	1797
## 10	3.5714	5	1150	311	648	245
## 11	6.0824	3	14014	2861	7205	2753
## 12	4.3594	23	4332	857	2461	829
## 13	3.1142	12	9831	1921	4644	1775
## 14	3.0068	23	6010	1116	2710	1149
## 15	1.7875	18	1676	332	733	318
## 16	2.2000	20	908	206	481	211
## 17	2.8871	22	2828	610	986	391
## 18	2.3684	35	632	148	221	102
## 19	1.4896	17	1483	284	481	211
## 20	2.1010	51	956	196	662	180
## 21	1.3720	47	1314	416	1155	326
## 22	3.2639	10	2522	533	1335	493
## 23	4.3810	34	2084	339	868	347

## 24	1.7184	11	1947	488	2104	486
## 25	1.9056	25	1689	495	1745	457
## 26	1.3386	20	1287	310	954	269
## 27	2.8182	29	1563	293	883	288
## 28	2.2083	18	902	195	771	174
## 29	4.1250	46	55	11	26	5
## 30	2.0192	18	2603	576	1616	588
## 31	1.6607	16	255	73	85	38
## 32	2.3851	21	1155	210	510	175
## 33	2.5625	23	2502	481	1443	455
## 34	2.8466	34	2051	342	958	322
## 35	2.4917	43	1320	215	512	197
## 36	1.8320	39	844	161	535	165
## 37	2.1746	20	2332	518	1856	495
## 38	0.9570	52	661	316	392	244
## 39	3.0469	25	2395	431	983	375
## 40	1.0486	31	938	238	425	157
## 41	2.0156	29	1029	239	509	196
## 42	1.8993	17	1485	345	823	316
## 43	2.9911	23	2103	411	1019	387
## 44	3.8095	28	3513	634	1658	598
## 45	1.7153	25	2015	524	746	251
## 46	2.6250	17	1319	267	393	163
## 47	1.3190	16	2088	535	816	326
## 48	2.1864	17	2019	496	899	347
## 49	1.7292	17	290	94	135	57
## 50	1.2639	16	245	57	81	33
## 51	4.0208	15	208	49	51	20
## 52	3.8611	18	1894	319	846	317
## 53	5.5750	10	6435	1040	3242	1030
## 54	2.6726	29	1476	220	902	205
## 55	2.5208	21	1885	398	1539	388
## 56	1.8419	18	2327	642	799	335
## 57	1.8369	17	2354	514	775	380
## 58	1.5667	15	2966	669	1007	465
## 59	3.6250	35	2916	594	1870	432
## 60	2.0441	33	2579	564	1155	431
## 61	2.4286	23	207	45	171	50
## 62	2.4688	21	1548	308	1137	306
## 63	1.8750	43	187	38	106	40
## 64	2.9708	27	2533	518	1371	461
## 65	0.4999	16	411	114	26	19
## 66	2.2188	20	1125	231	521	196
## 67	3.8333	24	1098	193	353	145
## 68	3.3750	15	764	145	366	143
## 69	2.7188	26	1707	308	761	250
## 70	2.5000	29	779	136	364	123
## 71	3.6641	17	976	202	511	175
## 72	1.6833	27	777	185	318	115
## 73	3.6250	33	240	49	63	22
## 74	1.7214	11	2635	667	280	132
## 75	2.2264	11	2479	900	2466	855
## 76	1.4329	21	847	278	1283	277
## 77	11.1978	32	8041	1141	2768	1106

## 78	7.9096	39	2603	456	928	410
## 79	10.1882	38	4715	691	1660	637
## 80	10.3467	34	3203	483	949	439
## 81	1.3125	13	4284	1452	3806	1252
## 82	4.4674	18	4780	1192	1886	1036
## 83	6.1527	40	2933	565	1077	536
## 84	2.0469	39	2602	802	2178	737
## 85	6.2976	52	3034	406	1158	399
## 86	4.3250	52	1732	305	875	311
## 87	3.6974	10	4292	1075	2719	987
## 88	3.0179	49	192	41	83	38
## 89	3.3056	46	1840	379	866	360
## 90	3.1641	21	4031	923	2558	834
## 91	3.8438	28	5173	1069	3502	954
## 92	3.2122	29	1699	399	1052	411
## 93	3.8203	43	1810	343	988	307
## 94	5.9845	40	1983	298	853	271
## 95	4.4306	52	2203	430	1238	403
## 96	3.1204	52	1722	448	1122	425
## 97	3.1211	36	3514	818	2277	828
## 98	4.5590	23	2216	378	1006	338
## 99	4.7422	32	1437	257	752	245
## 100	4.5132	27	2565	479	1227	467
## 101	5.0323	30	2260	374	958	359
## 102	4.5147	34	1138	205	541	180
## 103	2.1759	35	1387	272	610	237
## 104	4.0074	25	1596	321	1378	308
## 105	3.6597	21	2361	464	1146	396
## 106	3.2314	31	1872	434	1511	405
## 107	2.5495	27	2080	412	1082	382
## 108	3.7125	28	2332	395	1041	344
## 109	2.3886	16	2785	616	1387	530
##	Distance_to_coast	Distance_to_LA	Distance_to_SanDiego	Distance_to_SanJose		
## 1	9263.041	556529.158	735501.8	67432.52		
## 2	5897.799	549879.078	728729.1	59581.96		
## 3	4920.814	549640.721	728459.1	59073.11		
## 4	4042.115	550230.006	729033.1	59534.20		
## 5	5542.835	540804.235	719659.4	50767.65		
## 6	14874.129	537496.672	716534.3	50508.74		
## 7	4975.515	530425.687	709202.7	39869.08		
## 8	4491.623	529602.551	708361.3	38897.98		
## 9	7070.266	520289.220	699016.9	29530.15		
## 10	12465.794	506589.506	685369.4	17489.37		
## 11	2347.977	517397.296	695988.0	25791.05		
## 12	8192.659	517686.792	696446.4	27366.30		
## 13	167886.982	710107.862	887844.7	269650.01		
## 14	168738.860	709710.023	887423.6	269669.08		
## 15	185813.489	700431.775	877530.7	270964.63		
## 16	184557.579	699809.825	876955.1	269772.21		
## 17	170914.167	678351.214	855182.5	256676.77		
## 18	177311.361	677020.122	853284.5	263561.04		
## 19	167332.446	665120.750	841424.9	253738.55		
## 20	172581.896	329859.165	507729.3	196772.33		
## 21	172581.896	329859.165	507729.3	196772.33		

## 22	90380.048	301319.704	480307.9	190753.07
## 23	91819.832	303012.766	482045.4	189184.42
## 24	124834.643	360290.996	539602.9	146979.35
## 25	125652.684	357015.361	536341.8	149233.72
## 26	119014.147	372076.121	551383.1	137220.61
## 27	120647.991	368311.120	547623.1	140165.72
## 28	98412.746	381013.998	560419.7	120092.64
## 29	171183.310	297897.186	475133.7	227601.50
## 30	136286.216	724477.856	902962.0	269805.92
## 31	89894.935	745210.343	924386.6	274118.58
## 32	120353.497	725731.338	904562.6	264118.43
## 33	149649.951	714784.390	893025.1	265454.45
## 34	148421.370	703609.852	882035.9	252173.58
## 35	143795.456	693311.766	871987.2	238031.81
## 36	147709.165	684447.619	862882.5	235159.52
## 37	152369.202	715474.279	893627.9	267563.84
## 38	3514.868	914261.432	1093636.2	432449.25
## 39	39927.733	880382.664	1059646.5	402627.86
## 40	19585.130	946300.516	1125213.3	473918.28
## 41	2854.892	960640.319	1139711.4	485366.45
## 42	3544.297	924228.966	1103532.4	444662.03
## 43	10979.972	910152.077	1089495.2	429591.51
## 44	15451.260	899592.600	1078947.8	418758.55
## 45	47871.003	863603.321	1042938.7	384061.60
## 46	4541.260	844834.098	1024179.9	357462.64
## 47	16723.032	854379.601	1033782.3	369070.91
## 48	22845.507	827955.697	1007362.7	343383.45
## 49	79296.190	218261.454	124078.9	692060.48
## 50	63505.531	228799.709	120488.5	705940.98
## 51	35909.850	252213.012	122055.2	733834.91
## 52	120389.021	166999.426	345476.2	333310.79
## 53	118246.747	167239.366	345924.9	332000.21
## 54	141059.659	196537.041	374896.5	306466.49
## 55	142781.896	202538.354	381023.5	300160.34
## 56	185414.205	187768.322	352472.6	365945.94
## 57	164169.900	158765.144	322738.8	387283.99
## 58	175174.375	176727.017	342473.0	370099.94
## 59	203171.001	192350.970	339225.6	412521.64
## 60	195241.767	183346.801	327284.1	423248.26
## 61	141968.497	268091.192	446763.4	239054.37
## 62	132049.987	270064.678	449137.3	232755.24
## 63	124693.997	282344.892	461641.6	217839.29
## 64	96829.102	280440.180	459692.4	212558.29
## 65	76935.132	726162.667	905551.2	247539.51
## 66	69917.087	700740.086	880145.6	218381.44
## 67	56046.539	709136.221	888489.2	223475.49
## 68	59008.671	653733.539	833128.1	171822.62
## 69	249861.369	820082.918	992808.2	420699.48
## 70	261505.593	821413.348	993379.3	428416.58
## 71	257737.341	795857.949	968877.2	397193.17
## 72	311911.760	792964.089	962344.6	425711.72
## 73	303538.639	754674.536	924651.4	389838.27
## 74	277649.860	766770.114	938870.8	381112.62
## 75	19879.791	24400.101	202904.3	467728.81

## 76	13125.675	13998.238	170403.7	499922.51
## 77	9773.561	21510.285	196091.5	474192.25
## 78	5056.527	29066.533	191616.0	479085.16
## 79	2539.482	24574.715	193311.6	477026.54
## 80	9139.553	17276.523	191187.4	479111.94
## 81	7813.507	19107.968	192336.9	477936.29
## 82	7097.339	18088.245	190028.0	480237.86
## 83	9249.330	16265.927	189743.9	480555.87
## 84	2365.257	35987.360	153465.4	517220.29
## 85	28284.180	14411.913	192597.4	479487.34
## 86	26160.128	13771.653	192663.9	479089.57
## 87	26893.916	14832.881	193594.8	478259.78
## 88	24918.504	15849.076	195114.5	476425.30
## 89	23105.018	17251.859	196661.9	474599.16
## 90	43670.560	39034.365	160481.2	518311.93
## 91	51280.178	47417.424	160629.0	521736.37
## 92	41417.425	29882.507	166253.0	510333.75
## 93	37792.630	18146.679	189275.7	484672.48
## 94	37541.238	19553.723	192118.4	481791.59
## 95	33794.587	9596.457	178261.7	494314.27
## 96	32900.746	8777.506	178741.5	493698.04
## 97	34075.525	10132.047	179213.1	493489.02
## 98	157240.859	800161.472	977620.9	354263.45
## 99	19197.784	633195.082	812135.1	142725.92
## 100	18490.586	632366.282	811292.9	141817.01
## 101	21360.894	626394.291	805352.5	136092.45
## 102	16152.094	636107.486	814986.5	145273.03
## 103	141338.152	740998.179	919204.4	289571.79
## 104	6092.747	86279.639	246754.0	429222.04
## 105	6092.747	86279.639	246754.0	429222.04
## 106	4267.922	88100.013	248158.5	428192.17
## 107	123269.616	642052.644	820038.7	208634.19
## 108	132007.832	651116.940	829038.8	217062.32
## 109	146866.197	648723.337	825569.2	233282.77
##	Distance_to_SanFrancisco clust			
## 1	21250.21	3		
## 2	17924.51	3		
## 3	16905.89	3		
## 4	16032.23	3		
## 5	24147.65	3		
## 6	32637.95	3		
## 7	31445.70	3		
## 8	31950.47	3		
## 9	40504.65	3		
## 10	54275.10	3		
## 11	42536.66	3		
## 12	43320.54	3		
## 13	226904.25	4		
## 14	227104.95	4		
## 15	232839.78	4		
## 16	231512.54	4		
## 17	223260.76	4		
## 18	232802.47	4		
## 19	224974.23	4		

## 20	259414.77	2
## 21	259414.77	2
## 22	258764.58	2
## 23	257181.51	2
## 24	211738.88	2
## 25	214278.34	2
## 26	201213.10	2
## 27	204451.29	2
## 28	185746.54	2
## 29	291490.60	2
## 30	220637.33	4
## 31	216934.52	4
## 32	211987.34	4
## 33	219032.36	4
## 34	205723.83	4
## 35	190772.80	4
## 36	191191.06	4
## 37	221642.68	4
## 38	368229.99	4
## 39	340128.10	4
## 40	412619.89	4
## 41	423027.14	4
## 42	381162.62	4
## 43	365841.80	4
## 44	354984.64	4
## 45	321062.69	4
## 46	291465.87	4
## 47	303883.86	4
## 48	278679.94	4
## 49	759547.48	1
## 50	773522.26	1
## 51	801545.27	1
## 52	400973.44	2
## 53	399712.57	2
## 54	373846.67	2
## 55	367524.70	2
## 56	431156.35	2
## 57	453380.60	1
## 58	435792.42	2
## 59	476454.24	1
## 60	487521.07	1
## 61	305486.18	2
## 62	299618.89	2
## 63	284898.93	2
## 64	280479.62	2
## 65	187319.89	4
## 66	156839.05	4
## 67	159671.38	4
## 68	112719.68	3
## 69	383658.23	4
## 70	392907.07	4
## 71	361519.52	4
## 72	397679.21	4
## 73	364686.03	4

## 74	350348.98	4
## 75	535745.51	1
## 76	567912.71	1
## 77	542175.87	1
## 78	546975.25	1
## 79	544966.64	1
## 80	547099.09	1
## 81	545916.10	1
## 82	548213.79	1
## 83	548542.99	1
## 84	585123.47	1
## 85	547522.45	1
## 86	547124.06	1
## 87	546294.62	1
## 88	544458.82	1
## 89	542630.21	1
## 90	586296.74	1
## 91	589674.43	1
## 92	578340.13	1
## 93	552697.42	1
## 94	549816.14	1
## 95	562349.12	1
## 96	561733.08	1
## 97	561523.54	1
## 98	304679.16	4
## 99	76482.30	3
## 100	75489.09	3
## 101	70218.98	3
## 102	78526.77	3
## 103	240724.63	4
## 104	496636.58	1
## 105	496636.58	1
## 106	495576.22	1
## 107	175914.55	3
## 108	182803.30	4
## 109	205473.38	4

Le dendrogramme « suggère » un découpage en 4 groupes.

###3-1) Quelles sont les variables quantitatives les plus corrélées avec la variable classification

```
Res$desc.var$quanti.var
```

##		Eta2	P-value
##	Distance_to_SanDiego	0.95124592	1.060594e-68
##	Distance_to_LA	0.93911998	1.222271e-63
##	Distance_to_SanFrancisco	0.84881432	6.383422e-43
##	Distance_to_SanJose	0.77728414	4.159111e-34
##	Distance_to_coast	0.37990056	6.548680e-11
##	Median_Income	0.20709518	1.984248e-05
##	Population	0.09044195	1.856219e-02
##	Households	0.08513113	2.457213e-02
##	Tot_Rooms	0.07529695	4.096247e-02



On déduit d'après l'analyse de la variance que la variable qui concerne Distance\_to\_SanDiego caractérise bien toutes les classes car elle a une p-value très petites par rapport les autres c'est à dire il y'a une grande intensité de liaison significative avec la variable de classe, et après on a la variable de Distance\_to\_LA et ainsi de suite . De plus on observe que les variables Distance\_to\_SanDiego , Distance\_to\_LA , Median\_Income , Population sont les plus corréles avec la variable classification

###3-2) Faire la description des classes retenues par variables

```
Res$desc.var$quanti
```

```
## $'1'
##               v.test Mean in category Overall mean sd in category
## Distance_to_SanFrancisco  8.422143      563248.51489 3.276710e+05 76025.840916
## Distance_to_SanJose      7.449826      495648.51928 3.176723e+05 75742.866264
## Median_Income            3.159899           4.18875 3.285273e+00  2.537943
## Median_Age               2.305255           31.40625 2.720183e+01 12.992448
## Distance_to_coast        -3.786386      40175.69719 8.471865e+04 50996.510834
## Distance_to_LA           -8.229724      61666.57396 4.395036e+05 73410.120350
## Distance_to_SanDiego     -8.479765      198257.27198 6.051504e+05 51507.106774
##               Overall sd      p.value
## Distance_to_SanFrancisco 1.873927e+05 3.696604e-17
## Distance_to_SanJose     1.600506e+05 9.346325e-14
## Median_Income           1.915513e+00 1.578238e-03
## Median_Age              1.221879e+01 2.115227e-02
## Distance_to_coast        7.881259e+04 1.528539e-04
## Distance_to_LA           3.075819e+05 1.876458e-16
## Distance_to_SanDiego     3.214683e+05 2.256498e-17
##
## $'2'
##               v.test Mean in category Overall mean sd in category
## Distance_to_coast        3.035842      133285.8      84718.65      28538.65
## Distance_to_SanDiego    -2.264380      457390.9      605150.45      73859.46
## Distance_to_LA          -2.555185      279970.3      439503.57      71764.78
## Distance_to_SanJose    -2.690949      230248.3      317672.35      76840.77
##               Overall sd      p.value
## Distance_to_coast        78812.59 0.002398650
## Distance_to_SanDiego     321468.27 0.023550742
## Distance_to_LA           307581.87 0.010613129
## Distance_to_SanJose     160050.57 0.007124914
##
## $'3'
##               v.test Mean in category Overall mean sd in category
## Tot_Rooms              2.449226      3.646667e+03 2.385440e+03 4364.126017
## Median_Income          2.435206      4.294511e+00 3.285273e+00  1.806282
## Population             2.362813      1.852000e+03 1.238789e+03 2100.452835
## Households             2.300718      6.407222e+02 4.312661e+02  735.595285
## Distance_to_SanDiego    2.042054      7.471797e+05 6.051504e+05 50631.525700
## Tot_Bedrooms           1.966268      6.706667e+02 4.853670e+02  753.134487
## Distance_to_coast       -3.855498      1.897579e+04 8.471865e+04 28244.623072
## Distance_to_SanFrancisco -6.763309      5.346016e+04 3.276710e+05 39510.418756
## Distance_to_SanJose     -6.811880      8.178930e+04 3.176723e+05 57107.495695
##               Overall sd      p.value
## Tot_Rooms               2.380081e+03 1.431638e-02
## Median_Income           1.915513e+00 1.488331e-02
```

```
## Population          1.199521e+03 1.813681e-02
## Households          4.207820e+02 2.140757e-02
## Distance_to_SanDiego 3.214683e+05 4.114618e-02
## Tot_Bedrooms        4.355714e+02 4.926762e-02
## Distance_to_coast    7.881259e+04 1.154944e-04
## Distance_to_SanFrancisco 1.873927e+05 1.348751e-11
## Distance_to_SanJose  1.600506e+05 9.633177e-12
##
## $'4'
##               v.test Mean in category Overall mean sd in category
## Distance_to_LA      8.381948      7.718649e+05 4.395036e+05 88617.097633
## Distance_to_SanDiego 8.302723      9.492337e+05 6.051504e+05 89016.706142
## Distance_to_coast    4.132286      1.267033e+05 8.471865e+04 88573.600411
## Households          -2.004842      3.225128e+02 4.312661e+02 308.961857
## Distance_to_SanFrancisco -2.093553      2.770953e+05 3.276710e+05 79341.939068
## Population          -2.427155      8.634615e+02 1.238789e+03 814.619523
## Median_Income       -3.570417      2.403597e+00 3.285273e+00 0.858904
##               Overall sd      p.value
## Distance_to_LA      3.075819e+05 5.205884e-17
## Distance_to_SanDiego 3.214683e+05 1.017519e-16
## Distance_to_coast    7.881259e+04 3.591726e-05
## Households          4.207820e+02 4.497993e-02
## Distance_to_SanFrancisco 1.873927e+05 3.629981e-02
## Population          1.199521e+03 1.521776e-02
## Median_Income       1.915513e+00 3.564132e-04
```

les variables qui ont p-value très petit inférieure à 5% ou elles ont les moyenne par catégorie plus grand que ses moyennes total elles caractérisent bien les classes. ###3.3) Calculer les taux d'inertie : Inertie Inter/Inertie total, avant et après la consolidation de la CAH. ###taux d'inertie avant la consolidation

```
Res$call$bw.before.consol
```

```
## [1] 239103455927
```

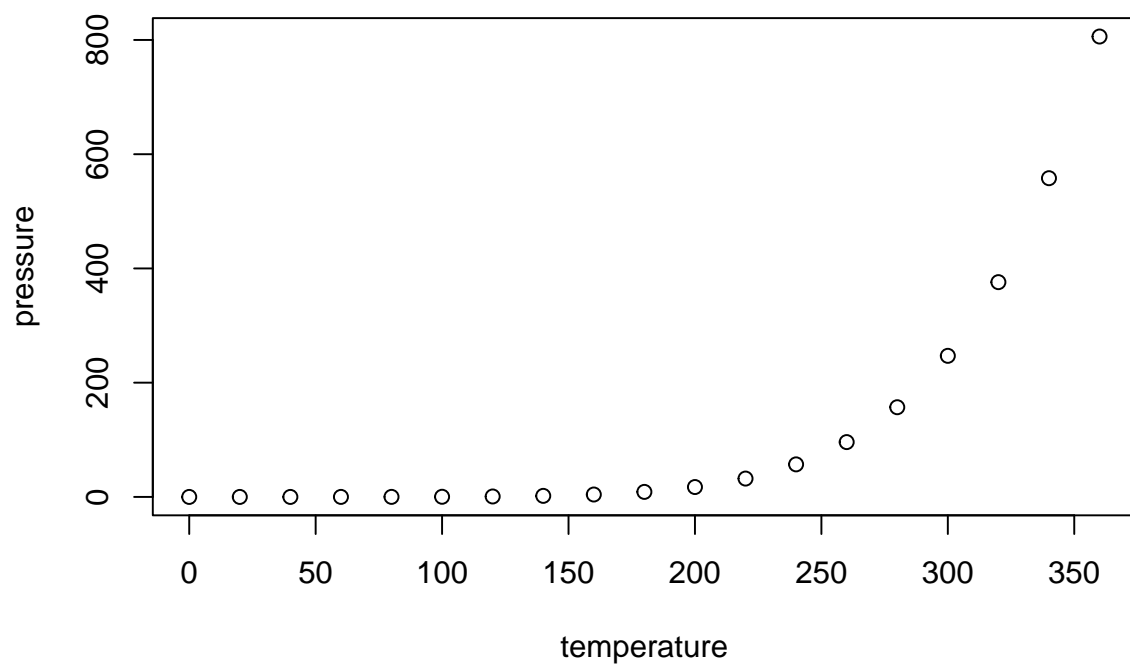
```
Res$call$bw.after.consol
```

**taux d'inertie après la consolidation:**

```
## [1] 239228821640
```

###4) Comparer les classifications faites par kmeans et CAH

La classification faites par K-Means a un taux d'inertie égale 78.3% avec un nombre de classes égale à 10, et celle faites par CAH a un taux d'inertie égale 57,77% avec un nombre de classes égale à 4. D'après cette comparaison, on peut conclure que La classification faites par K-Means est plus homogène et précise que celle faites par CAH.



## Chapitre 3

### ACP

# ACP

## Contents

Importation de data

```
library(readxl)
data_house <- read_excel("California_Houses.xlsx")
data_house$Median_Income <- as.numeric(data_house$Median_Income)
data_house$Distance_to_coast <- as.numeric(data_house$Distance_to_coast)
data_house$Distance_to_LA <- as.numeric(data_house$Distance_to_LA)
data_house$Distance_to_SanDiego <- as.numeric(data_house$Distance_to_SanDiego)
data_house$Distance_to_SanJose <- as.numeric(data_house$Distance_to_SanJose)
data_house$Distance_to_SanFrancisco <- as.numeric(data_house$Distance_to_SanFrancisco)
```

Les informations sur data

```
summary(data_house)
```

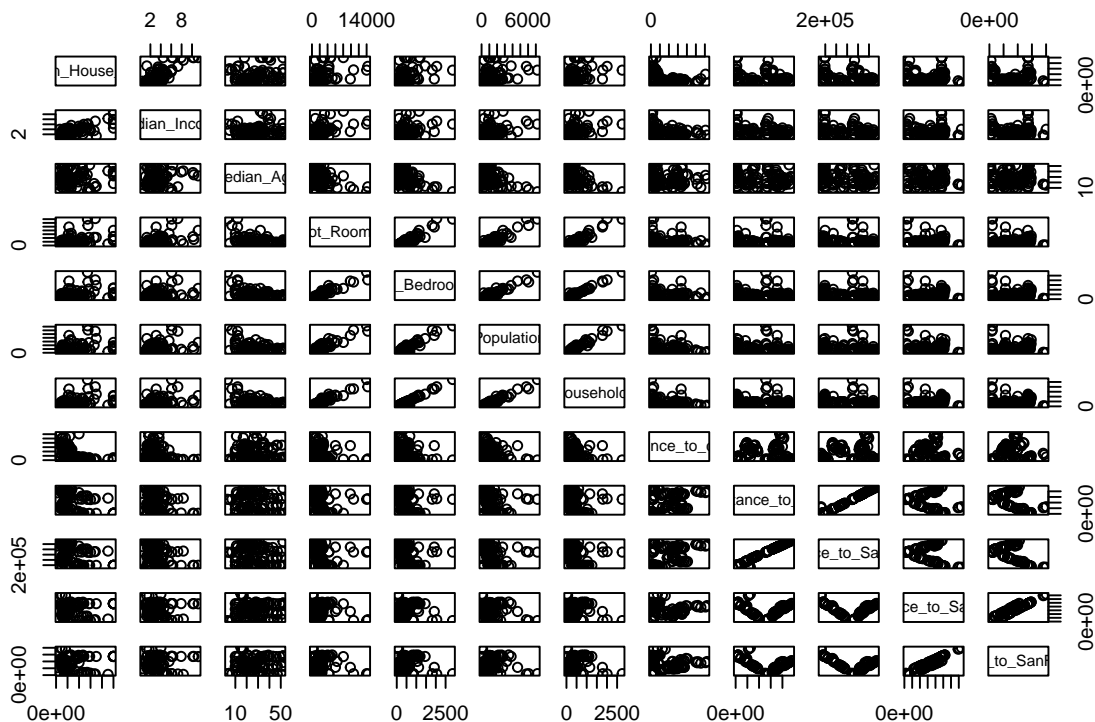
```
## Median_House_Value Median_Income      Median_Age      Tot_Rooms
## Min.   : 14999      Min.   : 0.4999  Min.   : 3.0    Min.   : 55
## 1st Qu.: 70000      1st Qu.: 2.0227  1st Qu.:18.0    1st Qu.: 1125
## Median :103100      Median : 2.8466  Median :25.0    Median : 1872
## Mean   :152960      Mean   : 3.2853  Mean   :27.2    Mean   : 2385
## 3rd Qu.:200000      3rd Qu.: 3.8438  3rd Qu.:35.0    3rd Qu.: 2602
## Max.   :500001      Max.   :11.1978  Max.   :52.0    Max.   :14316
## Tot_Bedrooms      Population      Households      Distance_to_coast
## Min.   : 11.0      Min.   : 26     Min.   : 5.0    Min.   : 2348
## 1st Qu.: 238.0      1st Qu.: 521    1st Qu.: 196.0  1st Qu.: 13126
## Median : 395.0      Median : 949     Median : 326.0  Median : 56047
## Mean   : 485.4      Mean   :1239     Mean   : 431.3  Mean   : 84719
## 3rd Qu.: 564.0      3rd Qu.:1387    3rd Qu.: 461.0  3rd Qu.:143795
## Max.   :2861.0      Max.   :7205     Max.   :2753.0  Max.   :311912
## Distance_to_LA      Distance_to_SanDiego Distance_to_SanJose
## Min.   : 8778      Min.   : 120489   Min.   : 17489
## 1st Qu.:166999      1st Qu.: 246754   1st Qu.:212558
## Median :517687      Median : 696446   Median :306467
## Mean   :439504      Mean   : 605150   Mean   :317672
## 3rd Qu.:709136      3rd Qu.: 887424   3rd Qu.:467729
## Max.   :960640      Max.   :1139711   Max.   :733835
## Distance_to_SanFrancisco
## Min.   : 16032
## 1st Qu.:205473
## Median :303884
## Mean   :327671
## 3rd Qu.:496637
## Max.   :801545
```

```
str(data_house)
```

```
## tibble [109 x 12] (S3: tbl_df/tbl/data.frame)
##  $ Median_House_Value      : num [1:109] 452600 127900 95500 75000 103100 ...
##  $ Median_Income           : num [1:109] 8.33 1.9 2.12 2.1 2.02 ...
##  $ Median_Age              : num [1:109] 41 51 47 51 44 12 18 31 7 5 ...
##  $ Tot_Rooms               : num [1:109] 880 1590 1372 175 1097 ...
##  $ Tot_Bedrooms            : num [1:109] 129 414 395 43 239 ...
##  $ Population              : num [1:109] 322 949 1237 228 609 ...
##  $ Households              : num [1:109] 126 392 303 55 215 ...
##  $ Distance_to_coast       : num [1:109] 9263 5898 4921 4042 5543 ...
##  $ Distance_to_LA          : num [1:109] 556529 549879 549641 550230 540804 ...
##  $ Distance_to_SanDiego    : num [1:109] 735502 728729 728459 729033 719659 ...
##  $ Distance_to_SanJose     : num [1:109] 67433 59582 59073 59534 50768 ...
##  $ Distance_to_SanFrancisco: num [1:109] 21250 17925 16906 16032 24148 ...
```

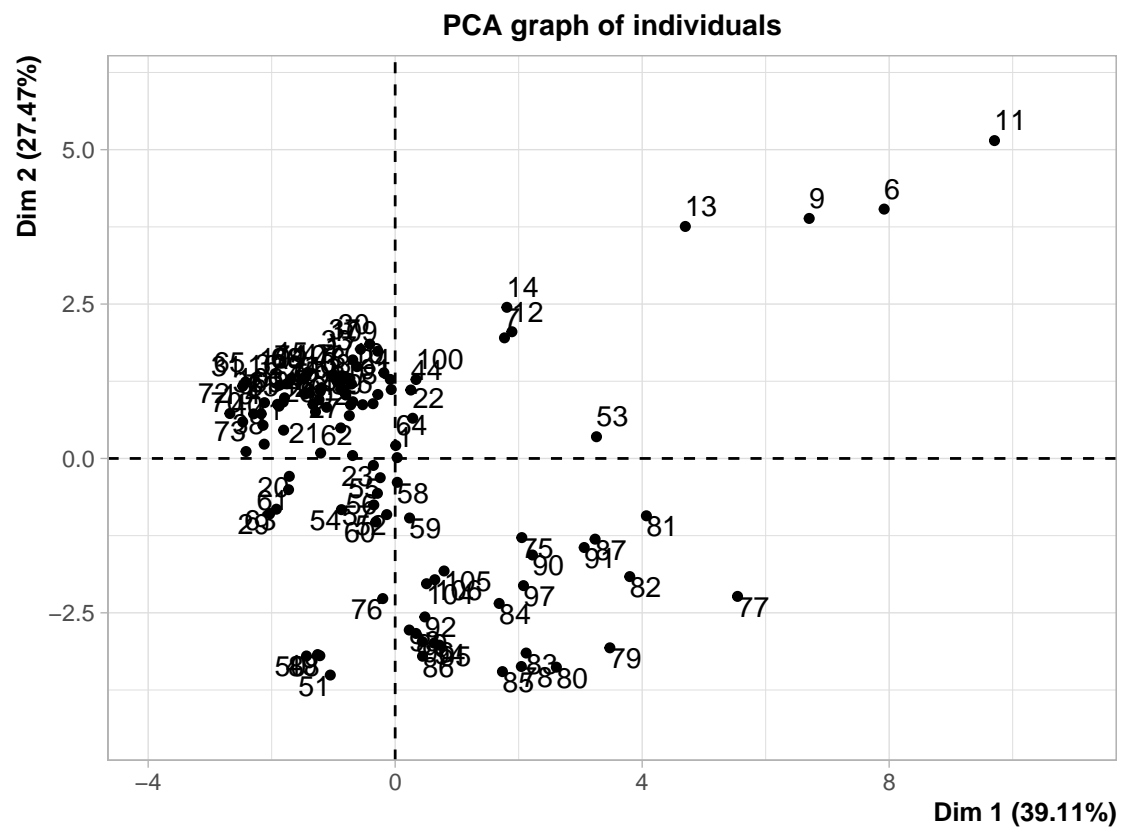
Nuages de points

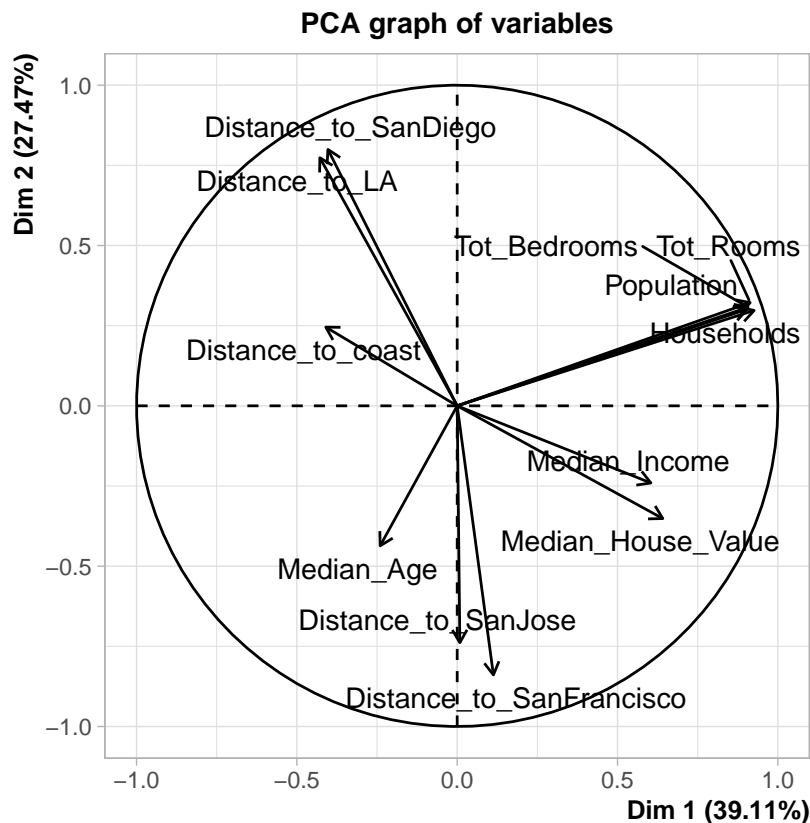
```
pairs(data_house)
```



1-Application de l'ACP normé

```
library(FactoMineR)
res<-PCA(data_house, axes=c(1,2))
```





```
attributes(res)
```

```
## $names
## [1] "eig" "var" "ind" "svd" "call"
##
## $class
## [1] "PCA" "list"
```

2-Les raisons pour centrer et réduire les variables dans l'analyse en composantes principales, ceci est particulièrement recommandé lorsque les variables sont mesurées dans différentes unités (par exemple: kilogrammes, kilomètres, centimètres, ...); sinon, le résultat de l'ACP obtenue sera fortement affecté.

3-Calcul l'indice KMO et les indices MSAI

```
library(psych)
KMO(cor(data_house))
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor(data_house))
## Overall MSA = 0.7
## MSA for each item =
```

	Median_House_Value	Median_Income	Median_Age
	0.65	0.58	0.67
	Tot_Rooms	Tot_Bedrooms	Population
	0.82	0.76	0.86



```
##           Households      Distance_to_coast      Distance_to_LA
##           0.80           0.64           0.61
## Distance_to_SanDiego      Distance_to_SanJose Distance_to_SanFrancisco
##           0.67           0.51           0.59
```

On a l'indice MSA globale = 0.7 alors les corrélations entre les variables de moyenne qualité et on a tous les MSA pour chaque variable est supérieur à 0.5 donc l'adéquation de l'échantillonnage est acceptable.

4-Calcul de valeurs propres

```
res$eig[,1]
```

```
##      comp 1      comp 2      comp 3      comp 4      comp 5      comp 6
## 4.693305624 3.296597455 1.630283298 0.814223619 0.705736132 0.588374520
##      comp 7      comp 8      comp 9      comp 10      comp 11      comp 12
## 0.187424951 0.042553715 0.020956875 0.009089954 0.006910457 0.004543399
```

Le pourcentage d'inertie de chaque valeur propre

```
res$eig[,2]
```

```
##      comp 1      comp 2      comp 3      comp 4      comp 5      comp 6
## 39.11088020 27.47164546 13.58569415 6.78519682 5.88113443 4.90312100
##      comp 7      comp 8      comp 9      comp 10      comp 11      comp 12
## 1.56187460 0.35461429 0.17464062 0.07574962 0.05758714 0.03786166
```

Le cumul des pourcentages d'inertie

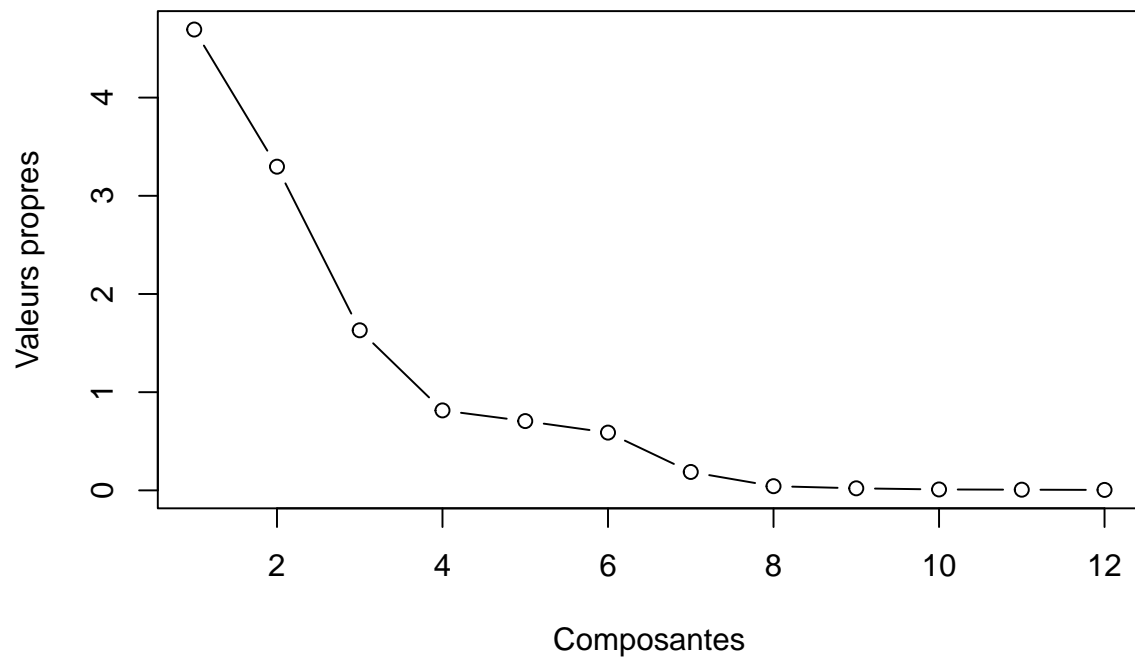
```
res$eig[,3]
```

```
##      comp 1      comp 2      comp 3      comp 4      comp 5      comp 6      comp 7      comp 8
## 39.11088 66.58253 80.16822 86.95342 92.83455 97.73767 99.29955 99.65416
##      comp 9      comp 10      comp 11      comp 12
## 99.82880 99.90455 99.96214 100.00000
```

5-Le graphique des valeurs propres.

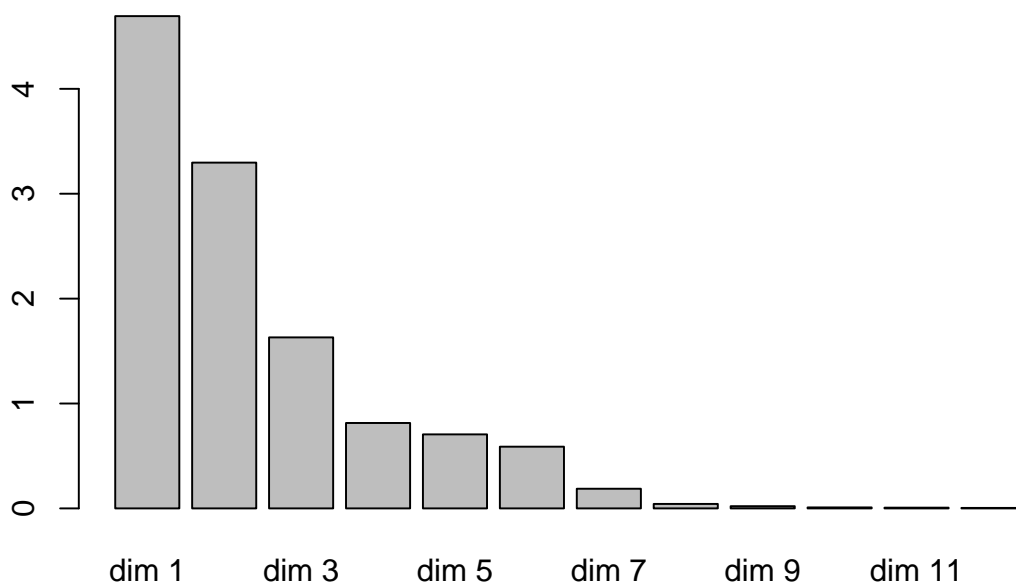
```
plot(1:12,res$eig[,1],type="b",ylab="Valeurs propres",xlab="Composantes",main="graphique des valeurs pr
```

### graphique des valeurs propres



```
barplot(res$eig[,1],main="Eigenvalues", names.arg=paste("dim",1:nrow(res$eig)))
```

## Eigenvalues



6-Détermination du dimension du sous espace en utilisant la règle rapport des variances

```
nrow(res$eig)
```

```
## [1] 12
```

```
var(res$eig[1:12,1])*11/(var(res$eig[,1])*11)
```

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

```
#supérieur à 0,05
```

```
var(res$eig[2:12,1])*10/(var(res$eig[,1])*11)
```

```
## [1] 0.407084
```

```
#supérieur à 0,05
```

```
var(res$eig[3:12,1])*9/(var(res$eig[,1])*11)
```

```
## [1] 0.1033769
```

```
#supérieur à 0,05
```

```
var(res$eig[4:12,1])*8/(var(res$eig[,1])*11)
```

```
## [1] 0.03647645
```

```
#inférieur à 0,05
```

alors la dimension de sous espace  $E$  est **3** d'après le règle rapport de variances

### Nuage des variables

7-Calculer le cos2 des variables sur le sous espace

```
print(res$var$cos2,digit=2)
```

##	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
## Median_House_Value	4.1e-01	0.124	0.2700	0.05174	1.8e-02
## Median_Income	3.6e-01	0.058	0.2768	0.05295	1.5e-01
## Median_Age	5.8e-02	0.191	0.2968	0.04490	2.1e-02
## Tot_Rooms	8.3e-01	0.104	0.0090	0.00095	1.3e-02
## Tot_Bedrooms	8.2e-01	0.095	0.0465	0.00011	7.1e-05
## Population	8.2e-01	0.091	0.0330	0.01230	2.3e-03
## Households	8.6e-01	0.088	0.0252	0.00127	3.1e-04
## Distance_to_coast	1.7e-01	0.060	0.2147	0.05519	4.9e-01
## Distance_to_LA	1.8e-01	0.598	0.0021	0.19088	1.5e-03
## Distance_to_SanDiego	1.6e-01	0.639	0.0097	0.15716	3.4e-03
## Distance_to_SanJose	6.9e-05	0.545	0.2273	0.19487	1.5e-03
## Distance_to_SanFrancisco	1.3e-02	0.704	0.2191	0.05189	1.5e-03

8-Distinguer les variables bien représentées, moyennement représentées et faiblement représentées sur le sous espace  $E$

```
print(t(apply(res$var$cos2,1,cumsum)),digit=2)
```

##	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
## Median_House_Value	4.1e-01	0.53	0.80	0.86	0.87
## Median_Income	3.6e-01	0.42	0.70	0.75	0.90
## Median_Age	5.8e-02	0.25	0.55	0.59	0.61
## Tot_Rooms	8.3e-01	0.94	0.95	0.95	0.96
## Tot_Bedrooms	8.2e-01	0.92	0.96	0.96	0.96
## Population	8.2e-01	0.91	0.94	0.95	0.96
## Households	8.6e-01	0.95	0.97	0.97	0.97
## Distance_to_coast	1.7e-01	0.23	0.44	0.50	0.99
## Distance_to_LA	1.8e-01	0.78	0.78	0.97	0.98
## Distance_to_SanDiego	1.6e-01	0.80	0.81	0.97	0.97
## Distance_to_SanJose	6.9e-05	0.55	0.77	0.97	0.97
## Distance_to_SanFrancisco	1.3e-02	0.72	0.94	0.99	0.99

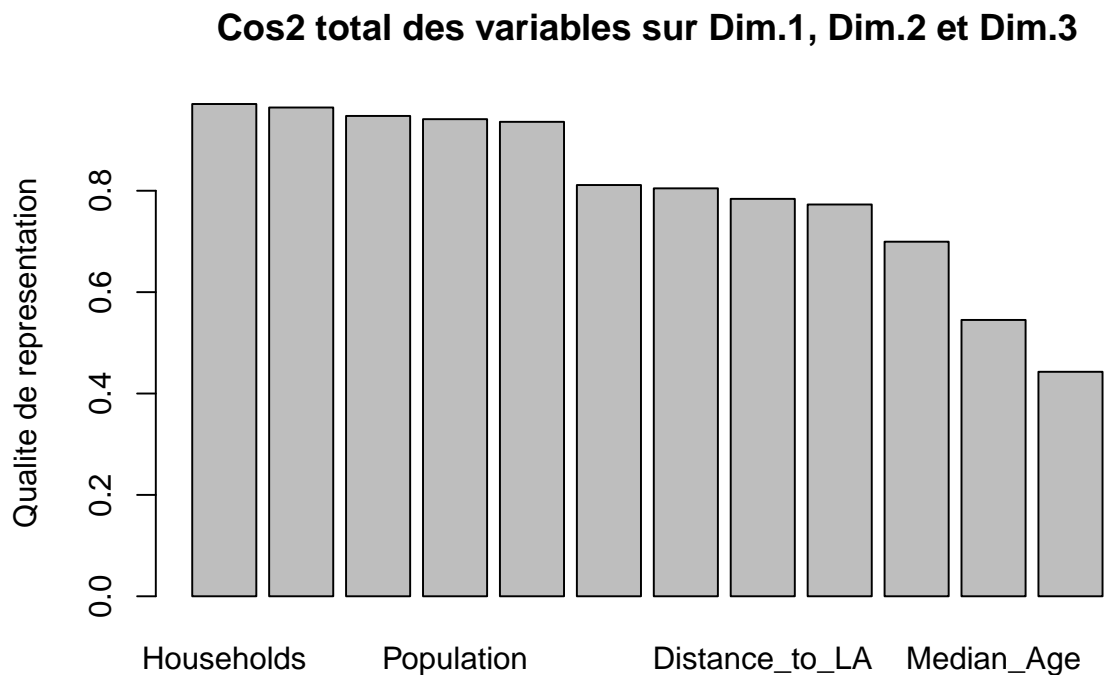
Cos2 total des variables sur Dim.1, Dim.2 et Dim.3 (**dim E=3**).

- Un cos2 élevé indique une bonne représentation de la variable sur les axes principaux.
- Un faible cos2 indique que la variable n'est pas parfaitement représentée par les axes principaux

```
cos_totale<-sort(t(apply(res$var$cos2,1,cumsum))[,3],decreasing=TRUE)
print(t(apply(res$var$cos2,1,cumsum))[,1:3],digit=2)
```

```
##              Dim.1 Dim.2 Dim.3
## Median_House_Value 4.1e-01 0.53 0.80
## Median_Income      3.6e-01 0.42 0.70
## Median_Age         5.8e-02 0.25 0.55
## Tot_Rooms          8.3e-01 0.94 0.95
## Tot_Bedrooms       8.2e-01 0.92 0.96
## Population         8.2e-01 0.91 0.94
## Households         8.6e-01 0.95 0.97
## Distance_to_coast  1.7e-01 0.23 0.44
## Distance_to_LA     1.8e-01 0.78 0.78
## Distance_to_SanDiego 1.6e-01 0.80 0.81
## Distance_to_SanJose 6.9e-05 0.55 0.77
## Distance_to_SanFrancisco 1.3e-02 0.72 0.94
```

```
barplot(cos_totale,ylab="Qualite de representation",main="Cos2 total des variables sur Dim.1, Dim.2 et 3")
```



Les variables bien représentées sont :

```
for (i in 1:12){
if (0.7 < cos_totale[i] ) print(cos_totale[i])
}
```

```
## Households
## 0.9711211
## Tot_Bedrooms
## 0.9642024
## Tot_Rooms
## 0.9475229
## Population
## 0.9411868
## Distance_to_SanFrancisco
## 0.9359422
## Distance_to_SanDiego
## 0.8112706
## Median_House_Value
## 0.804717
## Distance_to_LA
## 0.7839934
## Distance_to_SanJose
## 0.772837
```

Les variables moyennement représentées

```
for (i in 1:12){
if (0.5<cos_totale[i]&& cos_totale[i]<0.7) print(cos_totale[i])
}
```

```
## Median_Income
## 0.6994462
## Median_Age
## 0.5450949
```

Les variables faiblement représentées

```
for (i in 1:12){
if ( cos_totale[i]< 0.5) print(cos_totale[i])
}
```

```
## Distance_to_coast
## 0.442852
```

9- Calcule la contribution des variables dans chaque axe du sous espace

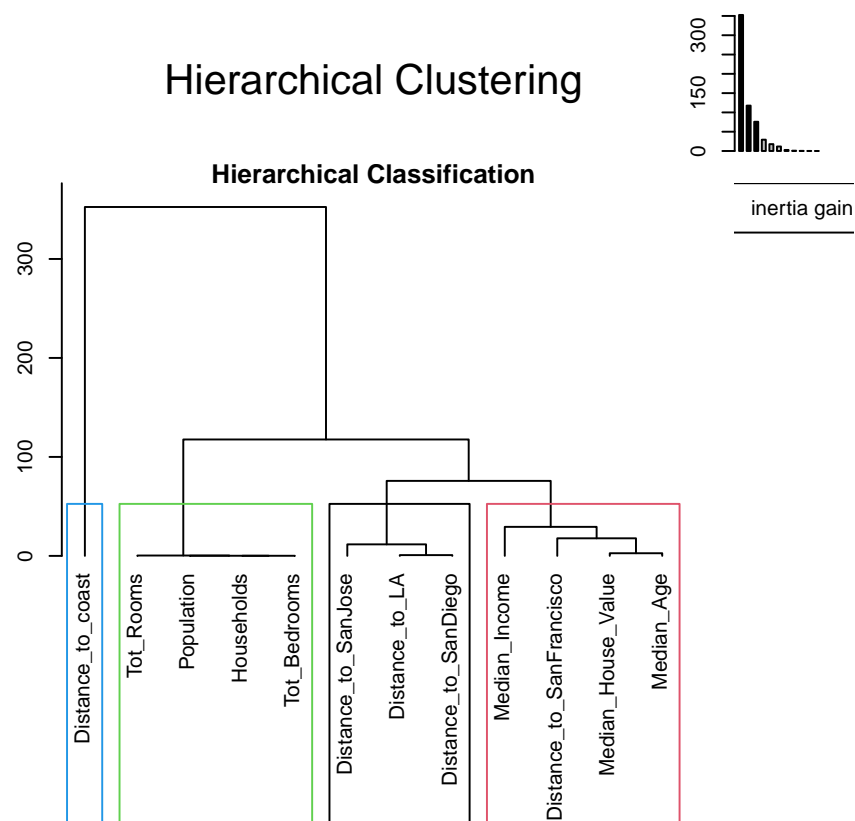
```
contrib<-data.frame(res$var$contrib)
print(res$var$contrib,digit=2)
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
## Median_House_Value	8.7596	3.7	16.56	6.355	2.557
## Median_Income	7.7733	1.8	16.98	6.503	21.295
## Median_Age	1.2302	5.8	18.21	5.515	2.907
## Tot_Rooms	17.7889	3.1	0.55	0.117	1.866

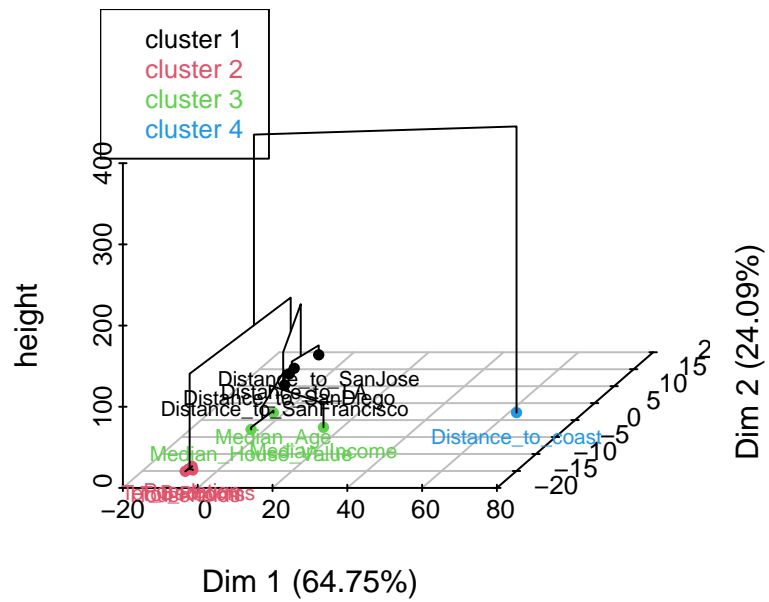
```
## Tot_Bedrooms      17.5381    2.9  2.85  0.013  0.010
## Population        17.4198    2.7  2.02  1.510  0.327
## Households        18.2700    2.7  1.55  0.156  0.044
## Distance_to_coast  3.5778    1.8 13.17  6.779 69.872
## Distance_to_LA     3.9076   18.2  0.13 23.443  0.215
## Distance_to_SanDiego 3.4615   19.4  0.60 19.302  0.484
## Distance_to_SanJose 0.0015   16.5 13.94 23.933  0.207
## Distance_to_SanFrancisco 0.2717  21.4 13.44  6.373  0.215
```

10-Application du CAH au tableau des contributions des variables aux axes du sous espace

```
cah<-HCPC(contrib,nb.clust=-1)
```

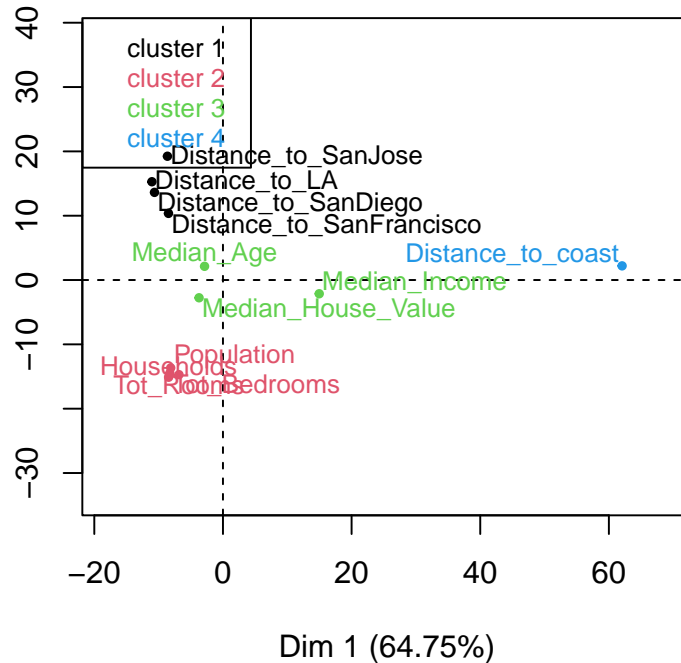


## Hierarchical clustering on the factor map





## Factor map



```
cah$desc.var
```

```
##
## Link between the cluster variable and the quantitative variables
## =====
##          Eta2      P-value
## Dim.2  0.9700891  1.946072e-06
## Dim.5  0.9486462  1.676018e-05
## Dim.1  0.9231259  8.325840e-05
## Dim.4  0.7643383  6.835550e-03
## Dim.3  0.7084517  1.556213e-02
##
## Description of each cluster by quantitative variables
## =====
## $'1'
##          v.test Mean in category Overall mean sd in category Overall sd
## Dim.2   3.258943      18.861285      8.333333      1.759295      7.576147
## Dim.4   2.745032      18.262960      8.333333      7.096447      8.483339
## Dim.1  -2.118335       1.910561      8.333333      1.783547      7.110648
##          p.value
## Dim.2  0.001118280
## Dim.4  0.006050497
## Dim.1  0.034146695
##
## $'2'
##          v.test Mean in category Overall mean sd in category Overall sd
```

```
## Dim.1  3.107164      17.7542258      8.333333      0.3262525      7.110648
## Dim.3 -2.142320      1.7435451      8.333333      0.8325211      7.213872
## Dim.4 -2.179560      0.4491928      8.333333      0.6147377      8.483339
##          p.value
## Dim.1 0.001888915
## Dim.3 0.032167736
## Dim.4 0.029290069
##
## $'3'
##          v.test Mean in category Overall mean sd in category Overall sd
## Dim.3 2.366826      17.24993      8.333333      0.6984303      7.213872
##          p.value
## Dim.3 0.01794135
##
## $'4'
##          v.test Mean in category Overall mean sd in category Overall sd
## Dim.5 3.170383      69.8719      8.333333      0      19.41045
##          p.value
## Dim.5 0.001522383
```

#les variables de '\$1' contribuent à les axes 2 et 4, ne contribuent pas à l'axe 1.

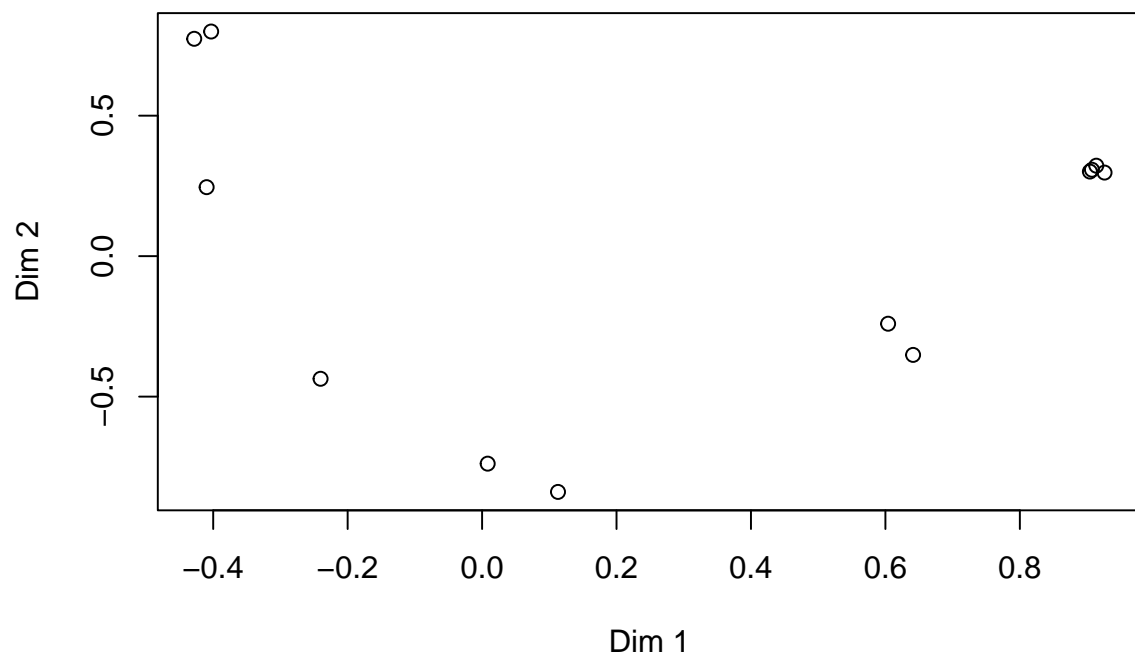
#les variables de '\$2' contribuent à l'axe 1, ne contribuent pas à les axes 3 et 4.

#les variables de '\$3' contribuent à l'axe 3.

#les variables de '\$3' contribuent à l'axe 5.

11-Tracer le nuage des variables projeté sur les 2 premiers axes.

```
plot(res$var$cor[,1],res$var$cor[,2],xlab="Dim 1",ylab="Dim 2")
```



12-les variables qui sont relativement bien corrélées (positivement et négativement) avec les axes du 1<sup>er</sup> plan factoriel sont :

```
dim1=res$var$cor[,1]
for (i in 1:12){
if (abs(dim1[i])>0.9) print(dim1[i])
}
```

```
## Tot_Rooms
## 0.9137222
## Tot_Bedrooms
## 0.907259
## Population
## 0.9041932
## Households
## 0.9259956
```

### Nuage des individus

13-Calcule du cos2 des individus sur le sous espace

```
print(res$ind$cos2,digit=2)
```

```
##      Dim.1  Dim.2  Dim.3  Dim.4  Dim.5
```

```

## 1  4.6e-05 7.9e-06 9.0e-01 1.4e-02 3.4e-02
## 2  6.8e-02 1.1e-01 4.2e-01 1.4e-01 9.0e-02
## 3  8.5e-02 1.5e-01 3.8e-01 1.5e-01 1.1e-01
## 4  3.2e-01 3.6e-02 4.0e-01 1.1e-01 7.8e-02
## 5  1.9e-01 1.1e-01 4.1e-01 1.3e-01 1.2e-01
## 6  7.8e-01 2.0e-01 3.5e-03 3.1e-05 3.0e-03
## 7  2.7e-01 3.3e-01 2.1e-01 4.2e-03 4.9e-02
## 8  9.2e-03 1.3e-01 7.4e-01 1.7e-02 2.0e-02
## 9  7.4e-01 2.5e-01 5.8e-04 1.2e-03 9.4e-04
## 10 2.7e-03 1.6e-01 2.1e-01 4.9e-10 7.7e-02
## 11 7.7e-01 2.2e-01 6.2e-03 1.3e-03 6.9e-04
## 12 3.3e-01 3.9e-01 2.0e-01 2.8e-02 3.5e-02
## 13 5.1e-01 3.2e-01 1.2e-01 2.9e-05 1.5e-02
## 14 2.7e-01 4.9e-01 9.6e-02 2.8e-05 6.0e-02
## 15 3.8e-01 3.7e-01 9.5e-02 2.4e-03 6.3e-02
## 16 6.0e-01 2.4e-01 3.9e-02 1.9e-03 6.8e-02
## 17 1.1e-01 6.3e-01 5.6e-02 1.2e-03 1.6e-01
## 18 8.0e-01 7.9e-02 3.5e-03 7.7e-03 9.1e-02
## 19 5.1e-01 3.0e-01 7.8e-02 1.0e-04 1.1e-02
## 20 3.5e-01 1.0e-02 3.4e-02 3.9e-01 6.4e-02
## 21 2.1e-01 1.1e-03 2.4e-05 5.3e-01 3.3e-02
## 22 2.3e-02 1.2e-01 7.9e-02 1.5e-01 4.1e-02
## 23 5.3e-02 5.4e-03 2.4e-01 4.0e-01 2.8e-02
## 24 1.1e-03 3.0e-01 1.5e-01 2.7e-01 4.1e-02
## 25 3.9e-02 2.4e-01 3.4e-02 6.2e-01 1.5e-02
## 26 2.7e-01 1.5e-01 1.9e-02 3.7e-01 4.8e-02
## 27 2.8e-01 8.7e-02 5.9e-02 5.1e-01 2.8e-03
## 28 3.3e-01 1.1e-01 9.7e-04 2.4e-01 5.4e-02
## 29 4.5e-01 8.7e-02 9.1e-02 1.8e-01 1.4e-01
## 30 3.8e-02 7.8e-01 1.6e-01 6.4e-03 5.5e-04
## 31 7.0e-01 1.6e-01 1.1e-02 2.5e-02 4.6e-02
## 32 6.4e-01 3.0e-01 1.8e-03 2.3e-02 2.3e-05
## 33 1.4e-01 7.3e-01 6.5e-02 2.8e-03 4.7e-02
## 34 4.0e-01 3.7e-01 2.2e-02 7.1e-04 1.2e-01
## 35 5.7e-01 1.2e-01 7.9e-02 1.8e-02 5.6e-02
## 36 7.0e-01 1.3e-01 2.8e-02 2.7e-02 2.2e-02
## 37 7.2e-02 7.2e-01 1.4e-01 2.8e-04 1.5e-02
## 38 3.2e-01 3.9e-03 2.0e-02 1.0e-01 7.5e-02
## 39 9.8e-02 1.8e-01 1.6e-04 6.4e-01 2.9e-02
## 40 4.1e-01 2.6e-02 1.9e-02 2.9e-01 1.1e-01
## 41 3.1e-01 2.0e-02 4.9e-03 4.5e-01 1.0e-01
## 42 1.7e-01 1.0e-01 4.0e-02 5.0e-01 1.7e-01
## 43 8.2e-02 1.2e-01 7.3e-04 6.6e-01 8.7e-02
## 44 1.0e-02 1.9e-01 2.1e-04 5.6e-01 2.5e-02
## 45 2.6e-01 2.1e-01 2.5e-02 3.1e-01 9.0e-02
## 46 2.6e-01 1.1e-01 8.3e-03 4.4e-01 1.5e-01
## 47 1.4e-01 2.6e-01 5.6e-02 2.9e-01 2.4e-01
## 48 1.2e-01 3.2e-01 9.8e-03 3.5e-01 1.9e-01
## 49 8.5e-02 5.5e-01 2.9e-01 2.7e-02 2.2e-02
## 50 1.0e-01 5.0e-01 3.0e-01 2.6e-02 4.4e-02
## 51 5.0e-02 5.6e-01 1.8e-01 8.2e-02 1.6e-02
## 52 6.4e-03 2.9e-01 1.1e-01 9.0e-02 2.7e-02
## 53 7.3e-01 8.5e-03 1.3e-01 9.8e-03 3.0e-02
## 54 2.5e-01 2.3e-01 7.2e-02 3.7e-01 2.7e-02

```

```

## 55 2.1e-02 3.4e-02 3.9e-01 4.1e-01 3.8e-03
## 56 1.6e-02 6.4e-02 5.6e-01 1.5e-01 4.6e-02
## 57 2.3e-02 1.1e-01 6.1e-01 1.4e-01 7.5e-03
## 58 2.1e-04 2.6e-02 6.9e-01 1.4e-01 1.1e-02
## 59 9.0e-03 1.5e-01 2.6e-01 1.7e-01 2.7e-01
## 60 1.6e-02 1.7e-01 4.3e-01 2.0e-01 1.0e-01
## 61 5.6e-01 4.8e-02 2.9e-03 1.7e-01 3.7e-03
## 62 1.8e-01 8.5e-04 1.3e-01 4.6e-01 2.2e-04
## 63 5.2e-01 9.6e-02 8.4e-02 2.5e-01 3.1e-03
## 64 5.3e-05 2.7e-02 2.3e-02 7.8e-01 1.7e-02
## 65 6.3e-01 1.7e-01 4.4e-03 1.0e-02 1.0e-01
## 66 4.7e-01 3.5e-01 3.1e-02 1.7e-02 6.7e-02
## 67 4.3e-01 2.2e-01 1.7e-01 5.2e-02 1.5e-02
## 68 3.4e-01 2.6e-01 8.9e-02 9.2e-03 4.3e-02
## 69 3.5e-01 9.0e-02 1.8e-01 6.3e-02 2.9e-01
## 70 5.3e-01 3.0e-02 1.0e-01 4.8e-02 2.6e-01
## 71 3.5e-01 7.4e-02 1.2e-01 8.9e-02 3.2e-01
## 72 4.7e-01 3.5e-02 1.8e-01 9.9e-03 2.7e-01
## 73 4.1e-01 8.8e-04 3.0e-03 3.0e-02 5.0e-01
## 74 1.7e-01 1.5e-01 2.8e-01 4.2e-02 2.1e-01
## 75 3.7e-01 1.5e-01 2.7e-01 3.4e-03 1.4e-01
## 76 4.1e-03 5.3e-01 1.9e-01 1.7e-02 2.3e-01
## 77 7.0e-01 1.1e-01 6.0e-02 3.9e-02 7.2e-02
## 78 1.9e-01 5.2e-01 1.9e-01 4.4e-02 3.4e-02
## 79 4.0e-01 3.1e-01 1.6e-01 5.0e-02 6.9e-02
## 80 2.4e-01 3.9e-01 1.8e-01 6.5e-02 7.3e-02
## 81 6.4e-01 3.3e-02 1.3e-01 4.8e-04 7.2e-02
## 82 6.6e-01 1.7e-01 6.8e-04 2.6e-02 2.0e-03
## 83 2.4e-01 5.4e-01 1.3e-01 2.5e-02 1.4e-02
## 84 2.4e-01 4.7e-01 3.7e-02 1.2e-02 8.4e-02
## 85 1.4e-01 5.6e-01 1.9e-01 7.6e-03 4.0e-02
## 86 1.6e-02 8.1e-01 8.5e-02 5.3e-03 7.0e-04
## 87 6.8e-01 1.1e-01 1.2e-01 3.0e-03 2.6e-02
## 88 1.1e-01 7.6e-01 1.3e-02 3.1e-02 2.1e-02
## 89 1.2e-02 8.6e-01 1.6e-02 1.8e-02 1.3e-02
## 90 4.4e-01 2.2e-01 2.9e-01 6.5e-03 3.5e-02
## 91 6.0e-01 1.3e-01 1.7e-01 9.2e-03 4.2e-03
## 92 3.1e-02 8.8e-01 5.2e-02 1.9e-03 2.9e-02
## 93 6.1e-03 9.1e-01 1.7e-03 2.0e-02 3.8e-03
## 94 3.8e-02 8.6e-01 5.4e-02 6.3e-04 1.5e-02
## 95 4.4e-02 7.8e-01 3.1e-02 1.4e-02 1.2e-03
## 96 1.7e-02 7.7e-01 9.7e-03 3.0e-02 4.7e-03
## 97 4.2e-01 4.1e-01 4.3e-02 1.6e-02 1.7e-02
## 98 1.4e-01 2.4e-01 4.0e-03 2.9e-01 2.8e-01
## 99 4.2e-02 1.2e-01 7.9e-01 1.1e-02 7.2e-03
## 100 2.0e-02 2.9e-01 6.2e-01 1.7e-02 1.6e-02
## 101 6.9e-04 2.1e-01 7.4e-01 9.2e-03 3.9e-03
## 102 7.3e-02 6.3e-02 8.1e-01 1.4e-02 7.4e-03
## 103 6.5e-01 2.0e-01 6.2e-04 2.0e-03 3.1e-02
## 104 4.7e-02 7.5e-01 5.2e-04 5.1e-05 1.0e-01
## 105 1.2e-01 6.3e-01 2.0e-02 5.0e-04 1.3e-01
## 106 8.0e-02 7.5e-01 1.9e-03 1.2e-02 1.5e-01
## 107 2.8e-01 6.8e-01 1.1e-02 2.2e-02 9.6e-03
## 108 2.0e-01 5.7e-01 8.4e-02 2.0e-04 1.4e-01

```

```
## 109 2.1e-02 8.0e-01 1.2e-01 7.8e-04 1.5e-02
```

14-Distinguer les individus bien représentées, moyennement représentées et faiblement représentées sur le sous espace

```
head(t(apply(res$ind$cos2,1,cumsum)),digit=2)
```

```
##          Dim.1          Dim.2          Dim.3          Dim.4          Dim.5
## 1 4.630675e-05 5.424525e-05 0.9004595 0.9145879 0.9484667
## 2 6.832609e-02 1.829146e-01 0.6015932 0.7464905 0.8368280
## 3 8.450087e-02 2.326882e-01 0.6162729 0.7712367 0.8859783
## 4 3.207658e-01 3.564085e-01 0.7537842 0.8611578 0.9388086
## 5 1.884896e-01 2.978226e-01 0.7038911 0.8364049 0.9567663
## 6 7.760080e-01 9.777528e-01 0.9812295 0.9812600 0.9842919
```

Cos2 total des individus sur Dim.1, Dim.2 et Dim.3 (**dim E=3**).

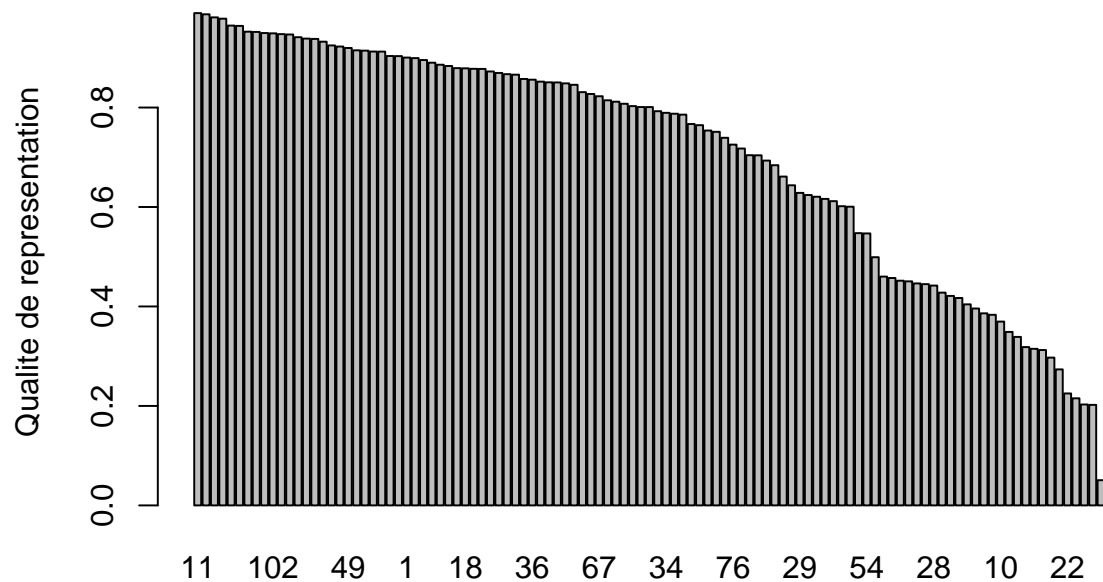
- Un cos2 élevé indique une bonne représentation de l'individu sur les axes principaux.
- Un faible cos2 indique que l'individu n'est pas parfaitement représentée par les axes principaux

```
cos_totale_ind<-sort(t(apply(res$ind$cos2,1,cumsum))[,3],decreasing=TRUE)
head(t(apply(res$ind$cos2,1,cumsum))[,1:3],digit=2)
```

```
##          Dim.1          Dim.2          Dim.3
## 1 4.630675e-05 5.424525e-05 0.9004595
## 2 6.832609e-02 1.829146e-01 0.6015932
## 3 8.450087e-02 2.326882e-01 0.6162729
## 4 3.207658e-01 3.564085e-01 0.7537842
## 5 1.884896e-01 2.978226e-01 0.7038911
## 6 7.760080e-01 9.777528e-01 0.9812295
```

```
barplot(cos_totale_ind,ylab="Qualite de representation",main="Cos2 total d'individus sur Dim.1, Dim.2 e
```

## Cos2 total d'individus sur Dim.1, Dim.2 et Dim.3



L'individus bien représentées sont :

```
for (i in 1:109){
  if (0.7 < cos_totale_ind[i] ) print(cos_totale_ind[i])
}
```

```
##      11
## 0.9898243
##      9
## 0.9874544
##      6
## 0.9812295
##     30
## 0.9786278
##     92
## 0.9647248
##    107
## 0.9641271
##     94
## 0.9525266
##     99
## 0.9520127
##    101
## 0.9497744
##    102
## 0.9491345
```

## 13  
## 0.9475825  
## 32  
## 0.9468562  
## 109  
## 0.9413639  
## 37  
## 0.938597  
## 90  
## 0.9379807  
## 33  
## 0.9323897  
## 100  
## 0.9247924  
## 12  
## 0.9225986  
## 49  
## 0.919686  
## 87  
## 0.9149978  
## 93  
## 0.9143706  
## 86  
## 0.9125517  
## 83  
## 0.9124321  
## 78  
## 0.9035758  
## 91  
## 0.9033524  
## 1  
## 0.9004595  
## 50  
## 0.8993582  
## 85  
## 0.8954692  
## 19  
## 0.8901612  
## 88  
## 0.8860173  
## 89  
## 0.8835401  
## 97  
## 0.8791687  
## 18  
## 0.8787676  
## 8  
## 0.877851  
## 16  
## 0.8776312  
## 77  
## 0.8724763  
## 79  
## 0.8695496



## 31  
## 0.8670702  
## 53  
## 0.866032  
## 14  
## 0.857411  
## 36  
## 0.8559845  
## 103  
## 0.8520419  
## 95  
## 0.8506257  
## 15  
## 0.8504203  
## 66  
## 0.8485618  
## 108  
## 0.8456871  
## 106  
## 0.8309307  
## 82  
## 0.8271827  
## 67  
## 0.8226732  
## 80  
## 0.8145195  
## 7  
## 0.8116749  
## 81  
## 0.8075735  
## 65  
## 0.8028336  
## 104  
## 0.8009644  
## 17  
## 0.8008827  
## 96  
## 0.792575  
## 34  
## 0.789288  
## 75  
## 0.7875437  
## 51  
## 0.7856887  
## 105  
## 0.7667868  
## 35  
## 0.7645206  
## 4  
## 0.7537842  
## 84  
## 0.751143  
## 57  
## 0.7391625

```
##          76
## 0.7254404
##          58
## 0.7175237
##          63
## 0.7041217
##          5
## 0.7038911
```

L'individus moyennement représentées

```
for (i in 1:109){
if (0.5<cos_totale_ind[i]&& cos_totale_ind[i]<0.7) print(cos_totale_ind[i])
}
```

```
##          72
## 0.693288
##          68
## 0.6840067
##          70
## 0.661166
##          56
## 0.6436413
##          29
## 0.628386
##          60
## 0.6240582
##          69
## 0.6205844
##          3
## 0.6162729
##          61
## 0.6117401
##          2
## 0.6015932
##          74
## 0.6006211
##          71
## 0.5474108
##          54
## 0.546866
```

L'individus faiblement représentées

```
for (i in 1:109){
if ( cos_totale_ind[i]< 0.5) print(cos_totale_ind[i])
}
```

```
##          45
## 0.4990167
##          47
```

## 0.460052  
## 40  
## 0.4572024  
## 48  
## 0.451795  
## 24  
## 0.4504915  
## 26  
## 0.4463198  
## 55  
## 0.4448893  
## 28  
## 0.4420027  
## 59  
## 0.4278237  
## 27  
## 0.4211836  
## 73  
## 0.4170074  
## 52  
## 0.4042233  
## 20  
## 0.3958552  
## 98  
## 0.3861728  
## 46  
## 0.3832654  
## 10  
## 0.3695482  
## 38  
## 0.3487752  
## 41  
## 0.3387638  
## 42  
## 0.3183005  
## 25  
## 0.3148111  
## 62  
## 0.3124077  
## 23  
## 0.2971944  
## 39  
## 0.2734616  
## 22  
## 0.2250692  
## 21  
## 0.215315  
## 44  
## 0.2029188  
## 43  
## 0.2021246  
## 64  
## 0.05069212

## 15-Calcul de la contribution des individus dans chaque axe du sous espace

```
contrib_ind<-data.frame(res$ind$contrib)
print(res$ind$contrib,digit=2)
```

##	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
## 1	2.1e-04	0.00005	1.1e+01	3.6e-01	0.99955
## 2	1.5e-01	0.35624	2.6e+00	1.8e+00	1.31187
## 3	1.7e-01	0.42026	2.2e+00	1.8e+00	1.52001
## 4	9.2e-01	0.14559	3.3e+00	1.8e+00	1.48157
## 5	3.8e-01	0.31315	2.4e+00	1.5e+00	1.61030
## 6	1.2e+01	4.53712	1.6e-01	2.8e-03	0.31850
## 7	6.1e-01	1.06120	1.4e+00	5.4e-02	0.74644
## 8	1.5e-02	0.29863	3.6e+00	1.6e-01	0.21789
## 9	8.8e+00	4.20345	2.0e-02	8.5e-02	0.07402
## 10	6.5e-03	0.53524	1.4e+00	6.8e-09	1.22403
## 11	1.8e+01	7.37243	4.3e-01	1.8e-01	0.10942
## 12	7.0e-01	1.17137	1.2e+00	3.4e-01	0.49326
## 13	4.3e+00	3.92741	2.9e+00	1.4e-03	0.84105
## 14	6.4e-01	1.66654	6.6e-01	3.8e-04	0.95409
## 15	3.9e-01	0.54257	2.8e-01	1.4e-02	0.42688
## 16	6.9e-01	0.39048	1.3e-01	1.3e-02	0.51973
## 17	7.5e-02	0.61486	1.1e-01	4.9e-03	0.73760
## 18	1.0e+00	0.14482	1.3e-02	5.7e-02	0.78031
## 19	5.6e-01	0.46491	2.4e-01	6.6e-04	0.08253
## 20	5.7e-01	0.02338	1.6e-01	3.7e+00	0.69053
## 21	2.8e-01	0.00214	9.3e-05	4.1e+00	0.29061
## 22	1.6e-02	0.11789	1.5e-01	5.7e-01	0.18173
## 23	2.4e-02	0.00354	3.1e-01	1.1e+00	0.08381
## 24	1.2e-03	0.45583	4.7e-01	1.7e+00	0.29724
## 25	2.5e-02	0.21888	6.2e-02	2.3e+00	0.06243
## 26	2.4e-01	0.19084	4.9e-02	1.9e+00	0.28087
## 27	1.5e-01	0.06786	9.3e-02	1.6e+00	0.01037
## 28	3.3e-01	0.15879	2.8e-03	1.4e+00	0.35628
## 29	8.1e-01	0.22161	4.7e-01	1.9e+00	1.69645
## 30	3.3e-02	0.95340	3.9e-01	3.1e-02	0.00313
## 31	1.2e+00	0.38249	5.5e-02	2.5e-01	0.52486
## 32	6.0e-01	0.40685	4.9e-03	1.3e-01	0.00014
## 33	9.2e-02	0.70867	1.3e-01	1.1e-02	0.21395
## 34	2.8e-01	0.36627	4.4e-02	2.9e-03	0.53835
## 35	6.9e-01	0.19924	2.7e-01	1.3e-01	0.44919
## 36	8.8e-01	0.22711	1.0e-01	1.9e-01	0.18416
## 37	6.1e-02	0.87039	3.5e-01	1.4e-03	0.08364
## 38	8.8e-01	0.01489	1.6e-01	1.6e+00	1.35965
## 39	9.2e-02	0.23485	4.4e-04	3.4e+00	0.18414
## 40	9.0e-01	0.08003	1.2e-01	3.7e+00	1.56917
## 41	6.4e-01	0.05831	2.9e-02	5.2e+00	1.37013
## 42	3.0e-01	0.25139	1.9e-01	4.9e+00	1.95688
## 43	1.0e-01	0.20978	2.6e-03	4.7e+00	0.71114
## 44	1.3e-02	0.34170	7.6e-04	4.0e+00	0.20787
## 45	2.9e-01	0.33202	7.9e-02	2.0e+00	0.65968
## 46	3.5e-01	0.21263	3.2e-02	3.3e+00	1.35954
## 47	2.0e-01	0.51258	2.2e-01	2.3e+00	2.15371
## 48	1.2e-01	0.46507	2.9e-02	2.1e+00	1.27387

```

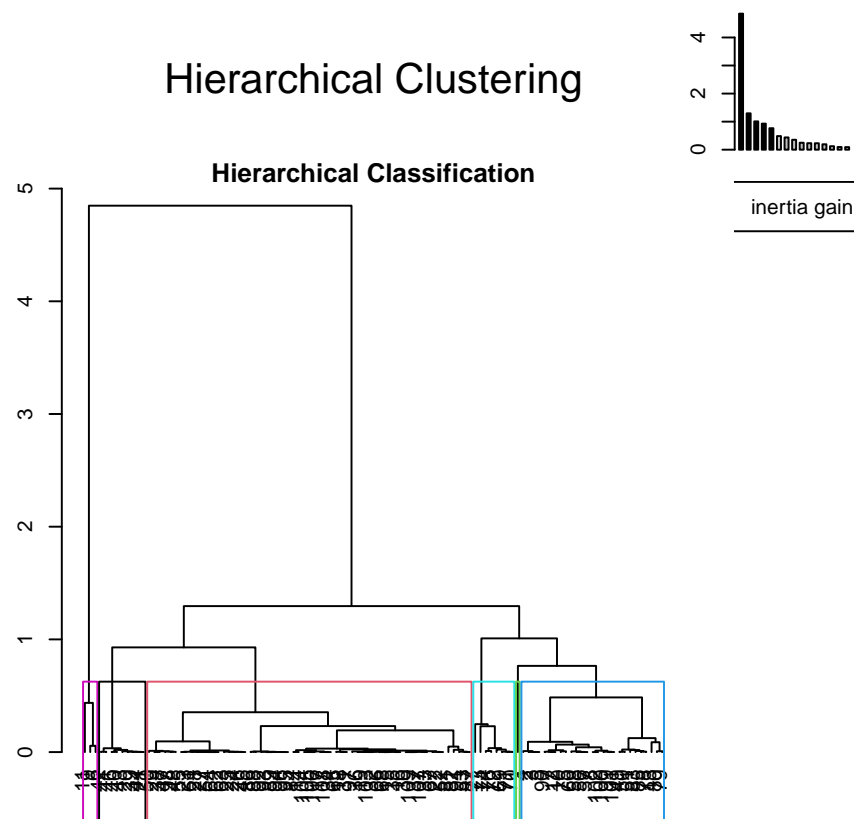
## 49 3.1e-01 2.80782 3.0e+00 5.7e-01 0.51966
## 50 4.0e-01 2.84769 3.5e+00 6.1e-01 1.18473
## 51 2.2e-01 3.42468 2.2e+00 2.0e+00 0.47230
## 52 3.6e-03 0.23015 1.8e-01 2.9e-01 0.10144
## 53 2.1e+00 0.03437 1.0e+00 1.6e-01 0.56467
## 54 1.5e-01 0.19137 1.2e-01 1.3e+00 0.10722
## 55 1.1e-02 0.02708 6.3e-01 1.3e+00 0.01397
## 56 1.6e-02 0.08885 1.6e+00 8.7e-01 0.29730
## 57 2.3e-02 0.15780 1.8e+00 8.1e-01 0.05147
## 58 2.3e-04 0.04164 2.2e+00 9.2e-01 0.08372
## 59 1.1e-02 0.25878 9.0e-01 1.2e+00 2.14970
## 60 1.9e-02 0.29423 1.5e+00 1.4e+00 0.81873
## 61 5.8e-01 0.07115 8.6e-03 1.0e+00 0.02554
## 62 9.3e-02 0.00061 1.9e-01 1.4e+00 0.00075
## 63 7.2e-01 0.18803 3.3e-01 2.0e+00 0.02863
## 64 1.7e-05 0.01220 2.1e-02 1.4e+00 0.03641
## 65 1.1e+00 0.42682 2.3e-02 1.1e-01 1.26454
## 66 4.4e-01 0.46477 8.5e-02 9.3e-02 0.41843
## 67 4.2e-01 0.30278 4.9e-01 2.9e-01 0.09650
## 68 4.3e-01 0.46476 3.2e-01 6.7e-02 0.36319
## 69 6.4e-01 0.23240 9.3e-01 6.6e-01 3.50911
## 70 1.2e+00 0.09762 6.8e-01 6.3e-01 3.98518
## 71 7.1e-01 0.21354 7.0e-01 1.0e+00 4.33672
## 72 1.4e+00 0.14729 1.6e+00 1.7e-01 5.33883
## 73 1.1e+00 0.00347 2.4e-02 4.8e-01 9.16172
## 74 4.1e-01 0.50422 1.9e+00 5.8e-01 3.38759
## 75 8.2e-01 0.45717 1.7e+00 4.4e-02 2.10085
## 76 7.9e-03 1.43339 1.0e+00 1.8e-01 2.90087
## 77 6.0e+00 1.38890 1.5e+00 2.0e+00 4.11805
## 78 8.2e-01 3.15890 2.3e+00 1.1e+00 0.97092
## 79 2.4e+00 2.61740 2.7e+00 1.7e+00 2.69817
## 80 1.3e+00 3.17969 3.0e+00 2.1e+00 2.74987
## 81 3.2e+00 0.24013 1.9e+00 1.4e-02 2.41131
## 82 2.8e+00 1.01875 8.4e-03 6.5e-01 0.05555
## 83 8.8e-01 2.76380 1.3e+00 5.1e-01 0.32615
## 84 5.5e-01 1.53445 2.5e-01 1.6e-01 1.28415
## 85 5.9e-01 3.31657 2.2e+00 1.8e-01 1.08895
## 86 3.8e-02 2.85483 6.0e-01 7.5e-02 0.01155
## 87 2.0e+00 0.47563 1.1e+00 5.1e-02 0.52511
## 88 2.9e-01 2.84025 9.8e-02 4.6e-01 0.37048
## 89 2.2e-02 2.23496 8.6e-02 1.9e-01 0.15662
## 90 9.7e-01 0.68018 1.8e+00 8.3e-02 0.51898
## 91 1.8e+00 0.57909 1.5e+00 1.6e-01 0.08642
## 92 4.5e-02 1.83331 2.2e-01 1.6e-02 0.27956
## 93 1.0e-02 2.14870 8.1e-03 1.9e-01 0.04227
## 94 7.8e-02 2.48882 3.2e-01 7.4e-03 0.20876
## 95 1.0e-01 2.54428 2.0e-01 1.9e-01 0.01876
## 96 3.8e-02 2.44725 6.3e-02 3.9e-01 0.06953
## 97 8.4e-01 1.18046 2.5e-01 1.9e-01 0.22458
## 98 1.3e-01 0.29336 1.0e-02 1.4e+00 1.62633
## 99 5.4e-02 0.21076 2.9e+00 8.4e-02 0.06112
## 100 2.2e-02 0.45463 2.0e+00 1.1e-01 0.11653
## 101 7.9e-04 0.34643 2.5e+00 6.1e-02 0.03041
## 102 1.1e-01 0.13353 3.5e+00 1.2e-01 0.07279

```

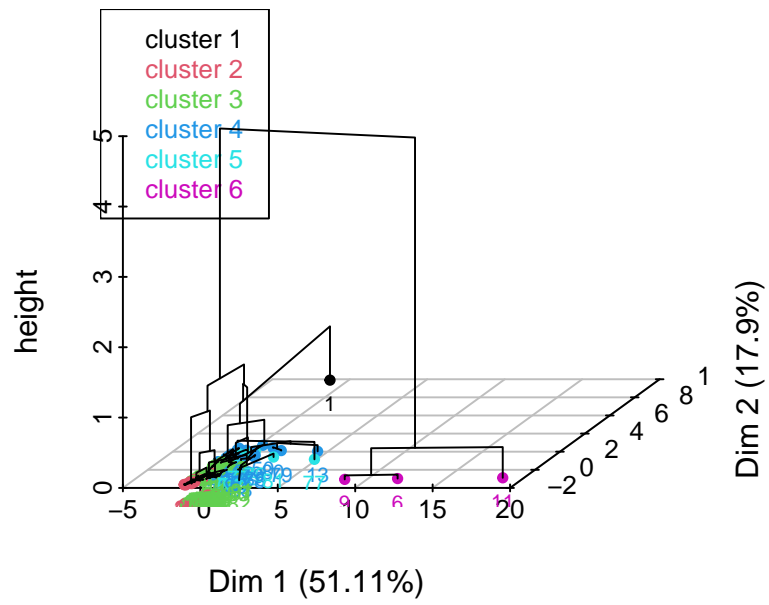
```
## 103 6.3e-01 0.26824 1.7e-03 1.1e-02 0.19915
## 104 5.0e-02 1.14535 1.6e-03 3.1e-04 0.72015
## 105 1.2e-01 0.92425 5.8e-02 3.0e-03 0.88979
## 106 8.1e-02 1.07345 5.4e-03 7.2e-02 0.97575
## 107 1.4e-01 0.50080 1.7e-02 6.5e-02 0.03332
## 108 9.8e-02 0.40430 1.2e-01 5.7e-04 0.46359
## 109 1.5e-02 0.83911 2.6e-01 3.3e-03 0.07295
```

16-Application du CAH au tableau des contributions des individus aux axes du sous espace

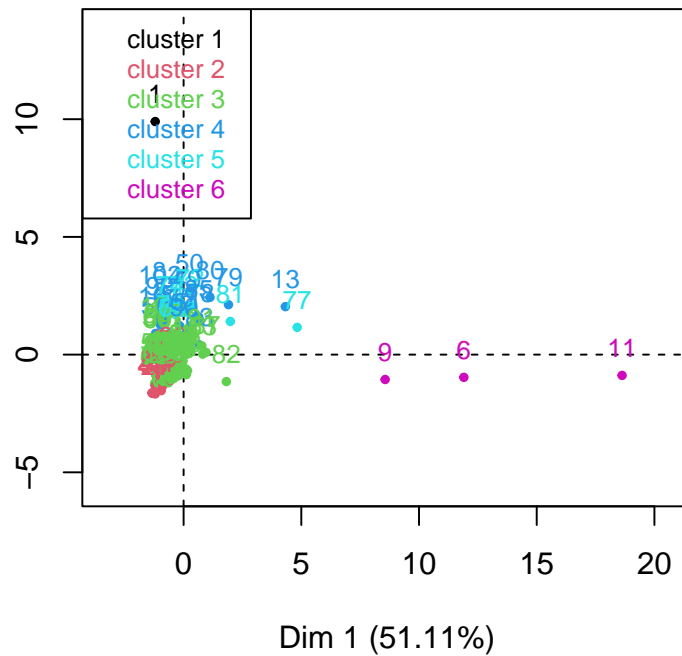
```
cah_ind<-HCPC(contrib_ind,nb.clust=-1)
```



## Hierarchical clustering on the factor map



## Factor map



```
cah_ind$desc.var
```

```
##
## Link between the cluster variable and the quantitative variables
## =====
##           Eta2      P-value
## Dim.3  0.8785889  1.645839e-45
## Dim.1  0.8024837  1.103943e-34
## Dim.5  0.6584047  1.478106e-22
## Dim.4  0.6564232  1.982281e-22
## Dim.2  0.4826755  1.807980e-13
##
## Description of each cluster by quantitative variables
## =====
## $'1'
##           v.test Mean in category Overall mean sd in category Overall sd
## Dim.3  7.394707           11.49985    0.9174312           0    1.43108
##           p.value
## Dim.3  1.417208e-13
##
## $'2'
##           v.test Mean in category Overall mean sd in category Overall sd
## Dim.4   8.291622           3.38838777    0.9174312    1.0924303    1.188885
## Dim.2  -2.241626           0.22602637    0.9174312    0.1477774    1.230505
## Dim.3  -2.279335           0.09979912    0.9174312    0.1254402    1.431080
```



```

##           p.value
## Dim.4 1.117149e-16
## Dim.2 2.498554e-02
## Dim.3 2.264716e-02
##
## $'3'
##           v.test Mean in category Overall mean sd in category Overall sd
## Dim.1 -2.525375      0.4243182    0.9174312    0.5731845    2.330662
## Dim.3 -4.361269      0.3945320    0.9174312    0.5122941    1.431080
## Dim.4 -4.505618      0.4686493    0.9174312    0.5913426    1.188885
## Dim.5 -4.623904      0.4080912    0.9174312    0.4513851    1.314794
##           p.value
## Dim.1 1.155748e-02
## Dim.3 1.293101e-05
## Dim.4 6.618009e-06
## Dim.5 3.765850e-06
##
## $'4'
##           v.test Mean in category Overall mean sd in category Overall sd
## Dim.3 5.752321      2.588426    0.9174312    0.6040732    1.431080
## Dim.2 2.622128      1.572377    0.9174312    1.4255622    1.230505
##           p.value
## Dim.3 8.802662e-09
## Dim.2 8.738257e-03
##
## $'5'
##           v.test Mean in category Overall mean sd in category Overall sd
## Dim.5 8.139282      4.349933    0.9174312    1.879991    1.314794
##           p.value
## Dim.5 3.976298e-16
##
## $'6'
##           v.test Mean in category Overall mean sd in category Overall sd
## Dim.1 9.178673      13.15344    0.9174312    3.980450    2.330662
## Dim.2 6.327679      5.37100    0.9174312    1.421766    1.230505
##           p.value
## Dim.1 4.364347e-20
## Dim.2 2.488756e-10

```

#les ind de \$'1' contribuent à l'axe 3

#les ind de \$'2' contribuent à l'axe 4 et ne contribuent ni à l'axe 2, ni à l'axe 3.

#les ind de \$'3' ne contribuent ni à l'axe 1, ni à l'axe 3, ni à l'axe 4, ni à l'axe 5.

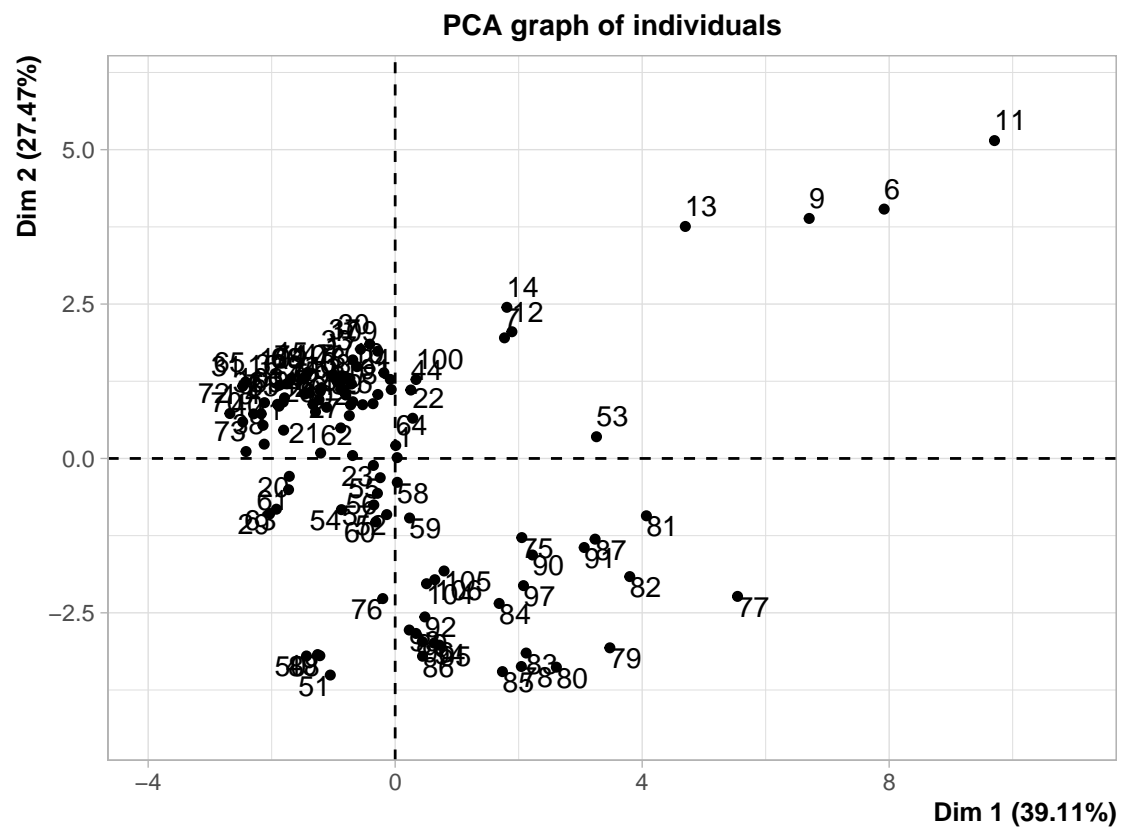
#les ind de \$'4' contribuent à l'axe 2 et à l'axe 3

#les ind de \$'5' contribuent à l'axe 5.

#les ind de \$'6' contribuent à les axes 1 et 2.

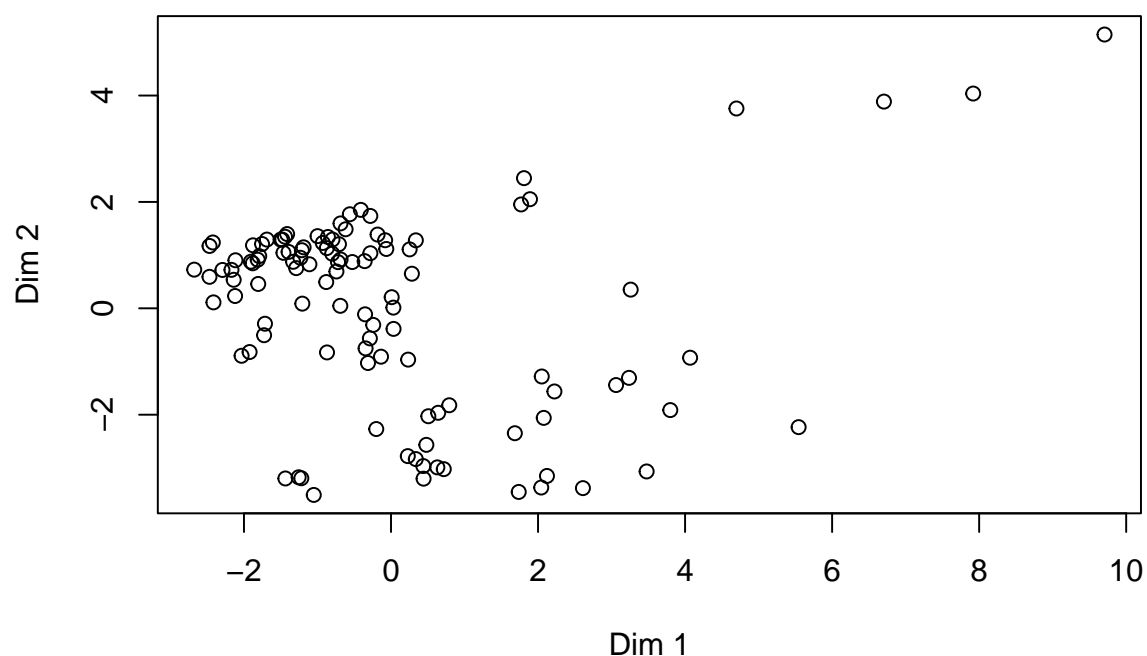
17-Tracer le nuage des individus projeté sur les 2 premiers axes

```
plot(res,invisible="var")
```



```
#plot(res$ind$coord[,1],res$ind$coord[,2],xlab="Dim 1",ylab="Dim 2")
```

```
plot(res$ind$coord[,1],res$ind$coord[,2],xlab="Dim 1",ylab="Dim 2")
```



### 18-Conclusion

Après l'application de l'ACP, nous avons réduit les 12 variables en 5 composantes principales. Cette réduction de dimensionnalité nous a permis d'utiliser les informations importantes contenues dans une table de données sur un plan factoriel et nous a permis de connaître les individus et les variables qui contribuent le plus dans le sous espace obtenu. On peut conclure que L'ACP peut être une première analyse pour l'étude d'une population dont les résultats seront enrichis par une autre analyse factorielle ou encore une classification automatique des données avec des variables quantitatives.

## Chapitre 4

### ACM

# ACM

Importation de data

```
library(readxl)

data_house <- read_excel("California_Houses.xlsx")
data_house$Median_Income <- as.numeric(data_house$Median_Income)
data_house$Distance_to_coast <- as.numeric(data_house$Distance_to_coast)
data_house$Distance_to_LA <- as.numeric(data_house$Distance_to_LA)
data_house$Distance_to_SanDiego <- as.numeric(data_house$Distance_to_SanDiego)
data_house$Distance_to_SanJose <- as.numeric(data_house$Distance_to_SanJose)
data_house$Distance_to_SanFrancisco <- as.numeric(data_house$Distance_to_SanFrancisco)
```

1-Regroupez dans un fichier Excel, les variables quantitatives et les variables qualitatives ainsi que le taux de discrétisation de chaque transformation

```
#centrage réduction des données
data_house<-as.data.frame(scale(data_house,center=T,scale=T))
data_kmeans=matrix(nrow = 109, ncol = 12)
taux_inertie=matrix(nrow = 1, ncol = 12)
for(i in 1:ncol(data_house)){
  km <- kmeans(data_house[,i],centers=3,nstart=5)
  data_kmeans[,i] <- km$cluster
  taux_inertie[,i]=km$betweenss/km$totss
}
#transformer matrice to data frame
taux_inertie=as.data.frame(taux_inertie)
data_kmeans=as.data.frame(data_kmeans)
#copier les noms des colonnes
names(data_kmeans)=names(data_house)
names(taux_inertie)=names(data_house)

data_kmeans[,1]=factor(data_kmeans[,1],labels = c('MH-', 'MH+', 'MH++'))
data_kmeans[,2]=factor(data_kmeans[,2],labels = c('MI-', 'MI+', 'MI++'))
data_kmeans[,3]=factor(data_kmeans[,3],labels = c('MA-', 'MA+', 'MA++'))
data_kmeans[,4]=factor(data_kmeans[,4],labels = c('TR-', 'TR+', 'TR++'))
data_kmeans[,5]=factor(data_kmeans[,5],labels = c('TB-', 'TB+', 'TB++'))
data_kmeans[,6]=factor(data_kmeans[,6],labels = c('P-', 'P+', 'P++'))
data_kmeans[,7]=factor(data_kmeans[,7],labels = c('H-', 'H+', 'H++'))
data_kmeans[,8]=factor(data_kmeans[,8],labels = c('DC-', 'DC+', 'DC++'))
data_kmeans[,9]=factor(data_kmeans[,9],labels = c('DL-', 'DL+', 'DL++'))
data_kmeans[,10]=factor(data_kmeans[,10],labels = c('DSD-', 'DSD+', 'DSD++'))
data_kmeans[,11]=factor(data_kmeans[,11],labels = c('DSJ-', 'DSJ+', 'DSJ++'))
data_kmeans[,12]=factor(data_kmeans[,12],labels = c('DSF-', 'DSF+', 'DSF++'))

#ecrire dans un fichier excel
```

```
library(xlsx)
# Ecrire la première table
write.xlsx(data_house, file = "data_kmeans.xlsx", sheetName="les variables quantitatives", append=FALSE)
# Ajouter une deuxième table
write.xlsx(data_kmeans, file = "data_kmeans.xlsx", sheetName="les variables qualitatives ", append=TRUE)
# Ajouter une troisième table
write.xlsx(taux_inertie, file = "data_kmeans.xlsx", sheetName="le taux de discrétisation de chaque trajectoire", append=TRUE)
# on a le taux de discrétisation est supérieur à 0.5.
```

2-Construction de tableau disjonctif complet

```
library(FactoMineR)
#Construction du tableau disjonctif complet
tabl_dis <- tab.disjonctif(data_kmeans)
```

3-Calcul de fréquence de chaque modalité et rechercher la présence d'une modalité rare (fréquence < 0.01).  
Si elle existe, il faut soit l'éliminer soit la combiner avec une autre modalité

```
table_freq=matrix(nrow = 12, ncol = 3)
table_freq[1,]=table(data_kmeans$Median_House_Value)
table_freq[2,]=table(data_kmeans$Median_Income)
table_freq[3,]=table(data_kmeans$Median_Age)
table_freq[4,]=table(data_kmeans$Tot_Rooms)
table_freq[5,]=table(data_kmeans$Tot_Bedrooms)
table_freq[6,]=table(data_kmeans$Population)
table_freq[7,]=table(data_kmeans$Households)
table_freq[8,]=table(data_kmeans$Distance_to_coast)
table_freq[9,]=table(data_kmeans$Distance_to_LA)
table_freq[10,]=table(data_kmeans$Distance_to_SanDiego)
table_freq[11,]=table(data_kmeans$Distance_to_SanJose)
table_freq[12,]=table(data_kmeans$Distance_to_SanFrancisco)
table_freq=as.data.frame(table_freq)
colnames(table_freq) <- c("-", "+", "++")
table_freq=table_freq/109
print(colSums(table_freq < 0.01))
```

```
##  -  +  ++
##  0  0  0
```

```
#n'existe aucune modalité rare
```

4-) Application de l'ACM au tableau disjonctif complet et précision de nombre d'individus, et de nombre de variables et celui des modalités

```
res<-MCA(data_kmeans, ncp=6, graph=FALSE)
#le nombre d'individus
nrow(res$ind$coord)
```

```
## [1] 109
```

```
#le nombre de variables
nrow(res$var$eta2)
```

```
## [1] 12
```

```
#le nombre de modalités
nrow(res$var$coord)
```

```
## [1] 36
```

5-les valeurs propres, le pourcentage d'inertie de chaque valeur propre ainsi que le cumul des pourcentages d'inertie.

```
#les valeurs propres
res$eig[,1]
```

```
##      dim 1      dim 2      dim 3      dim 4      dim 5      dim 6
## 4.282950e-01 4.156682e-01 2.336491e-01 2.025077e-01 1.615616e-01 1.199037e-01
##      dim 7      dim 8      dim 9      dim 10     dim 11     dim 12
## 8.960815e-02 8.322177e-02 5.909268e-02 5.097009e-02 3.866742e-02 2.887454e-02
##      dim 13     dim 14     dim 15     dim 16     dim 17     dim 18
## 2.393939e-02 2.011728e-02 1.350036e-02 1.057047e-02 8.712510e-03 5.027720e-03
##      dim 19     dim 20     dim 21     dim 22     dim 23     dim 24
## 4.056329e-03 2.055968e-03 2.830646e-31 1.294672e-31 6.466607e-32 3.640204e-32
```

```
# le pourcentage d'inertie de chaque valeur propre
res$eig[,2]
```

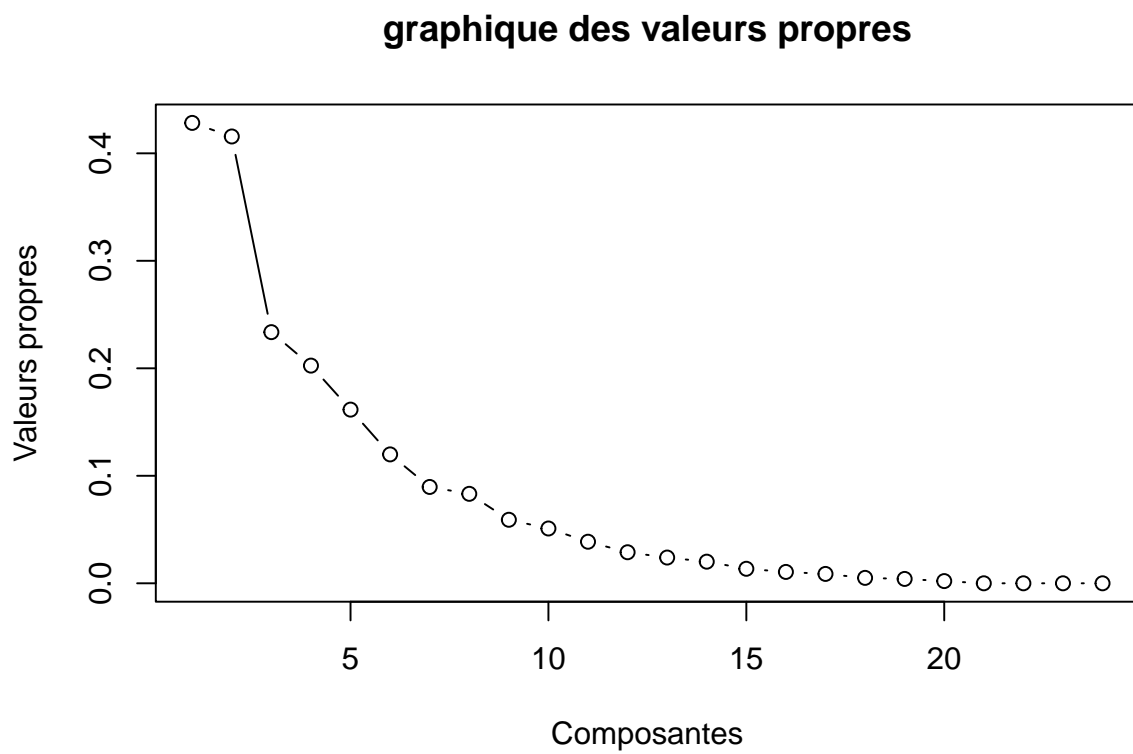
```
##      dim 1      dim 2      dim 3      dim 4      dim 5      dim 6
## 2.141475e+01 2.078341e+01 1.168245e+01 1.012538e+01 8.078082e+00 5.995185e+00
##      dim 7      dim 8      dim 9      dim 10     dim 11     dim 12
## 4.480407e+00 4.161088e+00 2.954634e+00 2.548505e+00 1.933371e+00 1.443727e+00
##      dim 13     dim 14     dim 15     dim 16     dim 17     dim 18
## 1.196969e+00 1.005864e+00 6.750179e-01 5.285235e-01 4.356255e-01 2.513860e-01
##      dim 19     dim 20     dim 21     dim 22     dim 23     dim 24
## 2.028165e-01 1.027984e-01 1.415323e-29 6.473360e-30 3.233303e-30 1.820102e-30
```

```
#le cumul des pourcentages d'inertie
res$eig[,3]
```

```
##      dim 1      dim 2      dim 3      dim 4      dim 5      dim 6      dim 7      dim 8
## 21.41475  42.19816  53.88062  64.00600  72.08408  78.07927  82.55967  86.72076
##      dim 9      dim 10     dim 11     dim 12     dim 13     dim 14     dim 15     dim 16
## 89.67540  92.22390  94.15727  95.60100  96.79797  97.80383  98.47885  99.00737
##      dim 17     dim 18     dim 19     dim 20     dim 21     dim 22     dim 23     dim 24
## 99.44300  99.69439  99.89720  100.00000  100.00000  100.00000  100.00000  100.00000
```

6-le graphique des valeurs propres

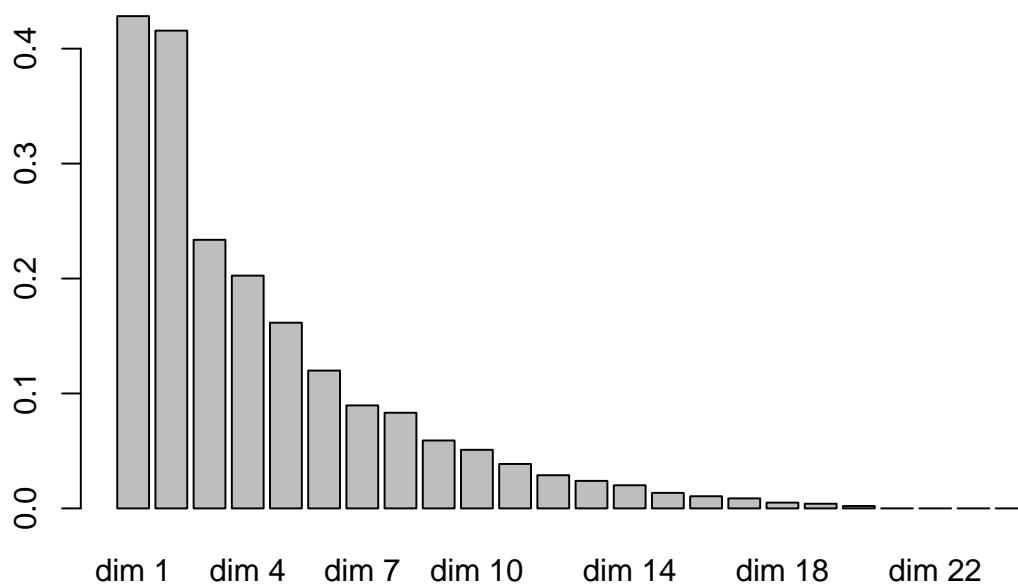
```
plot(res$eig[,1],type="b",ylab="Valeurs propres",xlab="Composantes",main="graphique des valeurs propres")
```



```
barplot(res$eig[,1],main="Eigenvalues", names.arg=paste("dim",1:nrow(res$eig)))
```



## Eigenvalues



7-Détermination du dimension du sous espace en utilisant la règle rapport des variances

```
nrow(res$eig)
```

```
## [1] 24
```

```
var(res$eig[1:24,1])*23/(var(res$eig[,1])*23)
```

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

```
#supérieur à 0,05
```

```
var(res$eig[2:24,1])*22/(var(res$eig[,1])*23)
```

```
## [1] 0.645625
```

```
#supérieur à 0,05
```

```
var(res$eig[3:24,1])*21/(var(res$eig[,1])*23)
```

```
## [1] 0.2856802
```

```
#supérieur à 0,05
```

```
var(res$eig[4:24,1])*20/(var(res$eig[,1])*23)
```

```
## [1] 0.1876212
```

```
#supérieur à 0,05  
var(res$eig[5:24,1])*19/(var(res$eig[,1])*23)
```

```
## [1] 0.1122598
```

```
#supérieur à 0,05  
var(res$eig[6:24,1])*18/(var(res$eig[,1])*23)
```

```
## [1] 0.06489347
```

```
#supérieur à 0,05  
var(res$eig[7:24,1])*17/(var(res$eig[,1])*23)
```

```
## [1] 0.04021069
```

```
#inférieur à 0,05  
#Le sous espace de projection est constitué des 6 premiers axes
```

alors la dimension de sous espace  $E$  est **6** d'après le règle rapport de variances

## Nuage de Modalités

8- Calcule le cos2 des modalités sur le sous espace

```
print(res$var$cos2,digit=2)
```

```
##          Dim 1   Dim 2   Dim 3   Dim 4   Dim 5   Dim 6  
## MH-   1.9e-01 0.01013 0.0646 0.00559 8.0e-02 0.50650  
## MH+   5.0e-01 0.02327 0.0898 0.00165 9.0e-02 0.00561  
## MH++  2.5e-01 0.05098 0.0291 0.00795 2.4e-01 0.12177  
## MI-   8.0e-02 0.00860 0.0041 0.02155 2.6e-01 0.12114  
## MI+   2.1e-01 0.14715 0.0809 0.04377 5.9e-02 0.34156  
## MI++  3.1e-01 0.01985 0.0120 0.00028 1.2e-01 0.00018  
## MA-   1.4e-02 0.02262 0.0101 0.00336 2.6e-01 0.02255  
## MA+   8.9e-05 0.03092 0.1319 0.03140 7.5e-02 0.09445  
## MA++  2.3e-02 0.00116 0.3010 0.02050 7.3e-02 0.03629  
## TR-   4.4e-01 0.07785 0.3636 0.00315 1.0e-07 0.01184  
## TR+   3.7e-02 0.80426 0.0227 0.12185 1.4e-03 0.00312  
## TR++  4.0e-01 0.04322 0.3453 0.07038 4.8e-04 0.02378  
## TB-   3.3e-01 0.06667 0.3548 0.02258 3.5e-03 0.00373  
## TB+   3.9e-01 0.02747 0.3903 0.00032 5.3e-03 0.00101  
## TB++  3.7e-02 0.80426 0.0227 0.12185 1.4e-03 0.00312  
## P-    3.2e-01 0.04702 0.3644 0.08530 5.6e-03 0.00447  
## P+    3.8e-01 0.04177 0.3997 0.01310 7.5e-03 0.00138  
## P++   3.7e-02 0.80426 0.0227 0.12185 1.4e-03 0.00312  
## H-    4.7e-01 0.04886 0.4024 0.00297 7.8e-04 0.00500  
## H+    4.2e-01 0.05508 0.3767 0.06003 1.1e-04 0.01145  
## H++   3.7e-02 0.80426 0.0227 0.12185 1.4e-03 0.00312  
## DC-   2.8e-01 0.00075 0.1001 0.00348 3.2e-01 0.03589  
## DC+   3.7e-01 0.00468 0.1171 0.00250 1.5e-01 0.00585
```

```
## DC++ 3.5e-02 0.00816 0.0047 0.05549 1.4e-01 0.05721
## DL- 5.0e-01 0.16037 0.0580 0.07395 5.3e-02 0.07183
## DL+ 3.6e-01 0.00809 0.1294 0.11619 2.0e-01 0.09118
## DL++ 8.7e-03 0.29363 0.0214 0.47206 6.8e-02 0.00254
## DSD- 5.0e-01 0.16154 0.0645 0.07373 5.3e-02 0.08377
## DSD+ 1.0e-02 0.30616 0.0173 0.48636 6.8e-02 0.00064
## DSD++ 3.6e-01 0.00809 0.1294 0.11619 2.0e-01 0.09118
## DSJ- 2.8e-01 0.20326 0.0517 0.18801 2.6e-02 0.00034
## DSJ+ 2.9e-03 0.34509 0.0247 0.37976 4.6e-02 0.00366
## DSJ++ 3.6e-01 0.00024 0.1334 0.00326 1.2e-01 0.00093
## DSF- 8.1e-03 0.31627 0.0265 0.20811 1.7e-01 0.01271
## DSF+ 5.6e-01 0.00469 0.1743 0.00768 3.6e-02 0.01416
## DSF++ 5.2e-01 0.14718 0.0991 0.07665 1.8e-02 0.04798
```

9-Distinguer les modalités bien représentées, moyennement représentées et faiblement représentées sur le sous espace E

```
print(t(apply(res$var$cos2,1,cumsum)),digit=2)
```

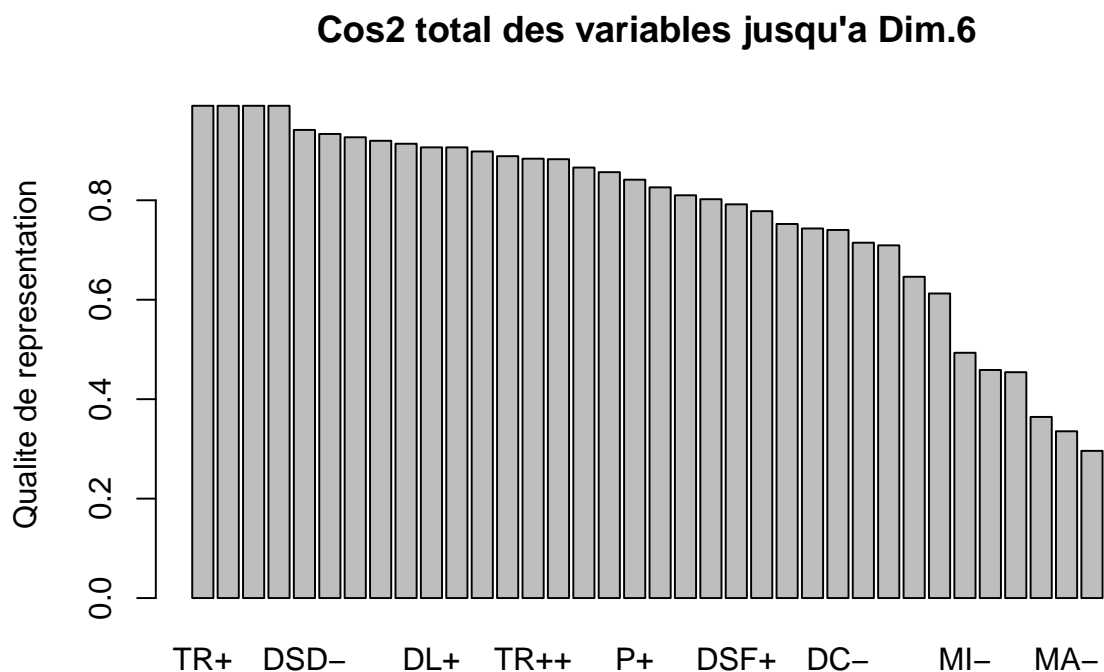
```
##      Dim 1 Dim 2 Dim 3 Dim 4 Dim 5 Dim 6
## MH- 1.9e-01 0.200 0.265 0.270 0.35 0.86
## MH+ 5.0e-01 0.528 0.618 0.619 0.71 0.71
## MH++ 2.5e-01 0.306 0.335 0.343 0.59 0.71
## MI- 8.0e-02 0.089 0.093 0.115 0.37 0.49
## MI+ 2.1e-01 0.358 0.439 0.482 0.54 0.88
## MI++ 3.1e-01 0.325 0.337 0.337 0.46 0.46
## MA- 1.4e-02 0.037 0.047 0.051 0.31 0.34
## MA+ 8.9e-05 0.031 0.163 0.194 0.27 0.36
## MA++ 2.3e-02 0.024 0.325 0.345 0.42 0.45
## TR- 4.4e-01 0.520 0.883 0.886 0.89 0.90
## TR+ 3.7e-02 0.841 0.864 0.986 0.99 0.99
## TR++ 4.0e-01 0.444 0.789 0.859 0.86 0.88
## TB- 3.3e-01 0.393 0.748 0.771 0.77 0.78
## TB+ 3.9e-01 0.413 0.803 0.804 0.81 0.81
## TB++ 3.7e-02 0.841 0.864 0.986 0.99 0.99
## P- 3.2e-01 0.366 0.731 0.816 0.82 0.83
## P+ 3.8e-01 0.420 0.819 0.833 0.84 0.84
## P++ 3.7e-02 0.841 0.864 0.986 0.99 0.99
## H- 4.7e-01 0.522 0.925 0.928 0.93 0.93
## H+ 4.2e-01 0.478 0.855 0.915 0.92 0.93
## H++ 3.7e-02 0.841 0.864 0.986 0.99 0.99
## DC- 2.8e-01 0.280 0.380 0.384 0.70 0.74
## DC+ 3.7e-01 0.371 0.488 0.490 0.64 0.65
## DC++ 3.5e-02 0.043 0.048 0.103 0.24 0.30
## DL- 5.0e-01 0.663 0.721 0.795 0.85 0.92
## DL+ 3.6e-01 0.368 0.497 0.614 0.82 0.91
## DL++ 8.7e-03 0.302 0.324 0.796 0.86 0.87
## DSD- 5.0e-01 0.666 0.731 0.804 0.86 0.94
## DSD+ 1.0e-02 0.316 0.334 0.820 0.89 0.89
## DSD++ 3.6e-01 0.368 0.497 0.614 0.82 0.91
## DSJ- 2.8e-01 0.486 0.538 0.726 0.75 0.75
## DSJ+ 2.9e-03 0.348 0.373 0.752 0.80 0.80
## DSJ++ 3.6e-01 0.356 0.489 0.492 0.61 0.61
```

```
## DSF- 8.1e-03 0.324 0.351 0.559 0.73 0.74
## DSF+ 5.6e-01 0.560 0.734 0.742 0.78 0.79
## DSF++ 5.2e-01 0.672 0.771 0.848 0.87 0.91
```

Cos2 total des modalités sur Dim.1, Dim.2, Dim.3, Dim.4, Dim.5, Dim.6. (**dim E=6**).

- Un cos2 élevé indique une bonne représentation de la modalité sur les axes principaux.
- Un faible cos2 indique que la modalité n'est pas parfaitement représentée par les axes principaux.

```
cos_totale<-sort(t(apply(res$var$cos2,1,cumsum))[,6],decreasing=TRUE)
barplot(cos_totale,ylab="Qualite de representation",main="Cos2 total des variables jusqu'a Dim.6")
```



Les modalités bien représentées sont :

```
for (i in 1:36){
if (0.7 < cos_totale[i] ) print(cos_totale[i])
}
```

```
##      TR+
## 0.9900793
##      TB++
## 0.9900793
##      P++
## 0.9900793
```

```

##      H++
## 0.9900793
##      DSD-
## 0.9415
##      H-
## 0.9334067
##      H+
## 0.926747
##      DL-
## 0.9196826
##      DSF++
## 0.9137051
##      DL+
## 0.9064108
##      DSD++
## 0.9064108
##      TR-
## 0.8981924
##      DSD+
## 0.8887223
##      TR++
## 0.8836263
##      MI+
## 0.882608
##      DL++
## 0.8658508
##      MH-
## 0.8565668
##      P+
## 0.8414154
##      P-
## 0.8259881
##      TB+
## 0.8100195
##      DSJ+
## 0.8021221
##      DSF+
## 0.7918618
##      TB-
## 0.778122
##      DSJ-
## 0.7523257
##      DSF-
## 0.7433153
##      DC-
## 0.7403408
##      MH+
## 0.71467
##      MH++
## 0.7093598

```

Les modalités moyennement représentées

```
for (i in 1:36){
if (0.5<cos_totale[i]&& cos_totale[i]<0.7) print(cos_totale[i])
}
```

```
##          DC+
## 0.6462008
##          DSJ++
## 0.6125426
```

Les modalités faiblement représentées

```
for (i in 1:36){
if ( cos_totale[i]< 0.5) print(cos_totale[i])
}
```

```
##          MI-
## 0.4933009
##          MI++
## 0.4587594
##          MA++
## 0.4542392
##          MA+
## 0.3642866
##          MA-
## 0.3354072
##          DC++
## 0.2960437
```

10-Calcul de la contribution des modalités dans chaque axe du sous espace

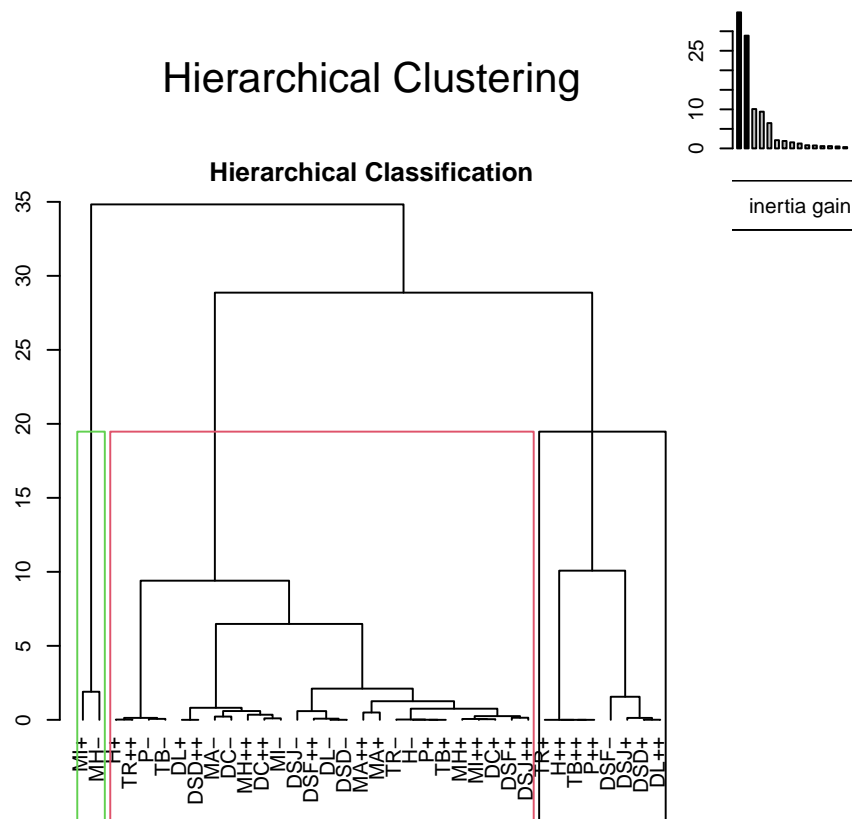
```
contrib<-data.frame(res$var$contrib)
print(res$var$contrib,digit=2)
```

```
##          Dim 1   Dim 2   Dim 3   Dim 4   Dim 5   Dim 6
## MH-    3.427   0.1883   2.136   0.2133  3.8e+00  32.618
## MH+    3.152   0.1498   1.029   0.0219  1.5e+00   0.125
## MH++   3.731   0.7689   0.781   0.2462  9.5e+00   6.367
## MI-    0.962   0.1060   0.089   0.5451  8.2e+00   5.175
## MI+    3.681   2.6523   2.595   1.6196  2.7e+00  21.343
## MI++   2.889   0.1935   0.207   0.0057  3.0e+00   0.006
## MA-    0.178   0.2871   0.229   0.0874  8.6e+00   0.992
## MA+    0.001   0.3639   2.763   0.7588  2.3e+00   3.854
## MA++   0.346   0.0181   8.372   0.6579  2.9e+00   1.967
## TR-    1.341   0.2434   2.022   0.0202  8.3e-07   0.128
## TR+    0.689  15.5321   0.779   4.8300  6.9e-02   0.209
## TR++   6.862   0.7632  10.848   2.5507  2.2e-02   1.456
## TB-    5.191   1.0913  10.333   0.7586  1.5e-01   0.212
## TB+    1.652   0.1212   3.065   0.0029  6.0e-02   0.015
## TB++   0.689  15.5321   0.779   4.8300  6.9e-02   0.209
## P-     5.072   0.7696  10.611   2.8662  2.3e-01   0.254
## P+     1.619   0.1844   3.139   0.1187  8.5e-02   0.021
```

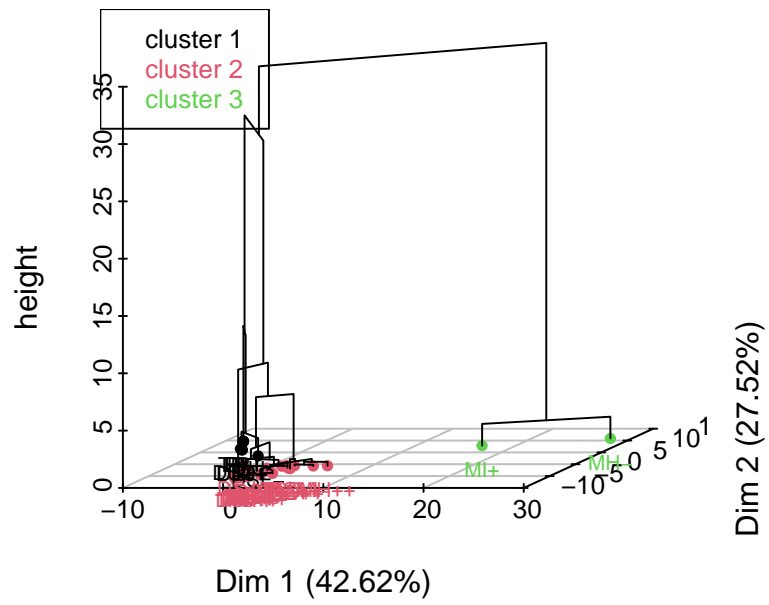
## P++	0.689	15.5321	0.779	4.8300	6.9e-02	0.209
## H-	1.690	0.1797	2.633	0.0224	7.4e-03	0.064
## H+	7.028	0.9422	11.464	2.1078	5.0e-03	0.679
## H++	0.689	15.5321	0.779	4.8300	6.9e-02	0.209
## DC-	3.292	0.0092	2.162	0.0868	1.0e+01	1.510
## DC+	3.201	0.0422	1.877	0.0463	3.5e+00	0.183
## DC++	0.641	0.1546	0.157	2.1579	6.6e+00	3.757
## DL-	6.459	2.1237	1.367	2.0101	1.8e+00	3.298
## DL+	4.110	0.0953	2.711	2.8075	6.1e+00	3.721
## DL++	0.128	4.4285	0.573	14.6138	2.6e+00	0.133
## DSD-	6.396	2.1095	1.498	1.9763	1.8e+00	3.792
## DSD+	0.152	4.6739	0.471	15.2401	2.7e+00	0.034
## DSD++	4.110	0.0953	2.711	2.8075	6.1e+00	3.721
## DSJ-	3.027	2.2431	1.014	4.2587	7.5e-01	0.013
## DSJ+	0.046	5.5854	0.711	12.6166	1.9e+00	0.205
## DSJ++	4.443	0.0031	3.055	0.0860	3.9e+00	0.042
## DSF-	0.131	5.2353	0.780	7.0712	7.3e+00	0.730
## DSF+	5.351	0.0466	3.080	0.1565	9.2e-01	0.488
## DSF++	6.934	2.0032	2.401	2.1415	6.2e-01	2.264

11-Application du CAH au tableau des contributions des modalités aux axes du sous espace

```
cah<-HCPC(contrib,nb.clust=-1)
```

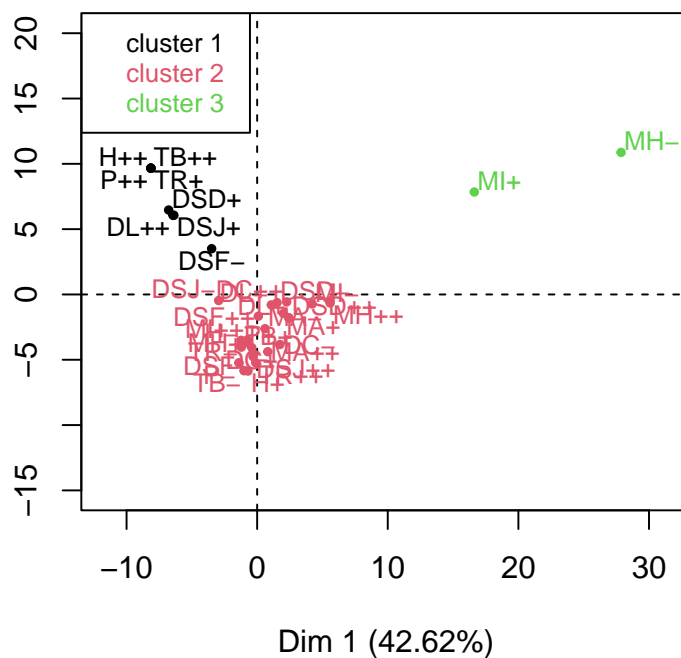


## Hierarchical clustering on the factor map





## Factor map



```
cah$desc.var
```

```
##
## Link between the cluster variable and the quantitative variables
## =====
##          Eta2      P-value
## Dim.6  0.8908590  1.339062e-16
## Dim.2  0.7060350  1.686080e-09
## Dim.4  0.6410534  4.549806e-08
## Dim.1  0.3132153  2.030205e-03
##
## Description of each cluster by quantitative variables
## =====
## $'1'
##          v.test Mean in category Overall mean sd in category Overall sd
## Dim.2  4.965397      10.2564318      2.777778      5.2854871      4.762878
## Dim.4  4.736199       8.6077339      2.777778      4.4099081      3.892560
## Dim.3 -2.047267       0.7065238      2.777778      0.1117705      3.199329
## Dim.1 -3.310388       0.4019059      2.777778      0.2889936      2.269573
##          p.value
## Dim.2  6.856069e-07
## Dim.4  2.177641e-06
## Dim.3  4.063191e-02
## Dim.1  9.316685e-04
##
## $'2'
```

```
##          v.test Mean in category Overall mean sd in category Overall sd
## Dim.1  2.821649          3.4490928    2.777778    2.2070962    2.269573
## Dim.3  1.994793          3.4467930    2.777778    3.5190811    3.199329
## Dim.4 -4.045209          1.1271254    2.777778    1.2283280    3.892560
## Dim.2 -4.399695          0.5810745    2.777778    0.7217168    4.762878
##          p.value
## Dim.1  4.777750e-03
## Dim.3  4.606550e-02
## Dim.4  5.227652e-05
## Dim.2  1.084029e-05
##
## $'3'
##          v.test Mean in category Overall mean sd in category Overall sd
## Dim.6  5.555016          26.98048    2.777778    5.637915    6.251556
##          p.value
## Dim.6  2.775856e-08
```

#les modalités de '\$1' contribuent à les axes 2, 3 et 4 mais ne contribuent pas à l'axe 5.

#les modalités de '\$2' ne contribuent ni à l'axe 2, ni à l'axe 4.

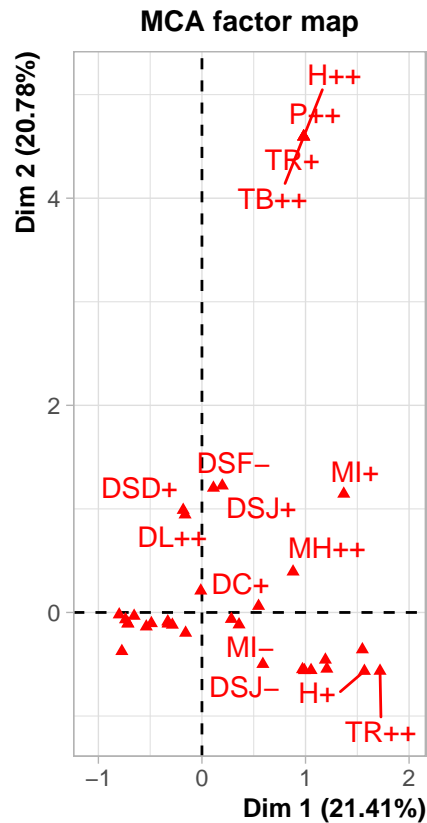
#les modalités de '\$3' contribuent à l'axe 6.

#Aucun modalité ne contribuent significativement aux axes 1 et 5.

12-Tracer le nuage des modalités projeté sur les 2 premiers axes

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

```
## Warning: ggrepel: 21 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



### Nuage des individus

13-Calcul du cos<sup>2</sup> des individus sur le sous espace

```
print(res$ind$cos2,digit=2)
```

```
##      Dim 1   Dim 2   Dim 3   Dim 4   Dim 5   Dim 6
## 1  2.8e-02 0.10643 2.6e-01 1.6e-01 2.3e-02 3.8e-01
## 2  2.0e-02 0.12486 1.9e-01 4.4e-01 7.8e-04 7.4e-03
## 3  2.0e-02 0.12486 1.9e-01 4.4e-01 7.8e-04 7.4e-03
## 4  2.0e-02 0.12486 1.9e-01 4.4e-01 7.8e-04 7.4e-03
## 5  2.0e-02 0.12486 1.9e-01 4.4e-01 7.8e-04 7.4e-03
## 6  6.5e-02 0.89931 2.1e-03 2.9e-02 1.3e-04 2.5e-05
## 7  2.4e-01 0.03497 1.5e-01 4.7e-01 3.4e-02 4.1e-04
## 8  8.6e-04 0.14311 1.2e-01 4.5e-01 2.1e-01 1.3e-02
## 9  6.5e-02 0.89931 2.1e-03 2.9e-02 1.3e-04 2.5e-05
## 10 1.9e-03 0.17539 9.6e-02 4.1e-01 9.4e-02 4.5e-02
## 11 6.5e-02 0.89931 2.1e-03 2.9e-02 1.3e-04 2.5e-05
## 12 2.3e-01 0.02529 1.3e-01 5.1e-01 8.5e-02 2.5e-03
## 13 4.7e-05 0.56352 1.1e-01 2.3e-01 3.9e-03 1.2e-02
## 14 1.9e-02 0.05065 7.2e-01 3.1e-02 5.2e-03 3.1e-02
## 15 6.2e-01 0.01143 9.1e-02 6.0e-02 5.2e-02 1.4e-05
## 16 6.2e-01 0.01143 9.1e-02 6.0e-02 5.2e-02 1.4e-05
## 17 2.5e-01 0.02060 2.5e-01 2.5e-02 5.1e-02 1.2e-03
## 18 6.4e-01 0.02552 5.9e-02 3.5e-02 1.4e-04 2.1e-02
## 19 6.2e-01 0.01143 9.1e-02 6.0e-02 5.2e-02 1.4e-05
```

```

## 20 2.4e-01 0.01427 2.8e-02 1.5e-01 3.3e-01 5.4e-04
## 21 2.4e-01 0.01427 2.8e-02 1.5e-01 3.3e-01 5.4e-04
## 22 2.0e-01 0.02532 2.1e-03 1.3e-01 1.5e-01 7.4e-02
## 23 2.1e-01 0.01231 3.6e-07 1.7e-01 3.8e-02 2.2e-02
## 24 5.4e-02 0.04771 2.1e-02 3.2e-01 1.4e-01 8.7e-03
## 25 6.0e-02 0.03143 1.1e-02 3.6e-01 4.4e-02 8.4e-05
## 26 1.8e-01 0.07966 5.1e-03 2.7e-01 1.4e-01 1.9e-02
## 27 1.9e-01 0.05549 1.3e-02 3.2e-01 4.0e-02 4.9e-04
## 28 1.8e-01 0.07966 5.1e-03 2.7e-01 1.4e-01 1.9e-02
## 29 1.5e-01 0.01424 2.8e-02 1.7e-01 1.5e-01 1.2e-02
## 30 5.9e-02 0.02478 4.4e-01 2.2e-05 4.2e-02 7.3e-03
## 31 6.2e-01 0.01143 9.1e-02 6.0e-02 5.2e-02 1.4e-05
## 32 6.2e-01 0.01143 9.1e-02 6.0e-02 5.2e-02 1.4e-05
## 33 6.4e-01 0.02552 5.9e-02 3.5e-02 1.4e-04 2.1e-02
## 34 6.4e-01 0.02552 5.9e-02 3.5e-02 1.4e-04 2.1e-02
## 35 4.7e-01 0.01735 1.1e-03 1.9e-02 6.0e-02 2.9e-02
## 36 4.7e-01 0.01735 1.1e-03 1.9e-02 6.0e-02 2.9e-02
## 37 2.5e-01 0.01720 2.5e-01 9.2e-03 5.4e-02 1.4e-03
## 38 1.5e-01 0.03556 4.6e-02 5.1e-02 9.2e-03 6.7e-02
## 39 1.2e-01 0.04590 3.2e-04 5.6e-02 3.3e-01 1.5e-02
## 40 2.4e-01 0.05014 1.9e-04 8.1e-02 1.4e-01 5.7e-02
## 41 5.0e-02 0.07729 2.9e-02 1.1e-01 1.2e-01 1.4e-02
## 42 2.2e-01 0.02849 1.9e-03 1.2e-01 2.2e-02 8.2e-03
## 43 1.2e-01 0.04590 3.2e-04 5.6e-02 3.3e-01 1.5e-02
## 44 1.2e-01 0.06758 5.1e-01 1.7e-02 8.4e-02 5.2e-02
## 45 2.4e-01 0.05014 1.9e-04 8.1e-02 1.4e-01 5.7e-02
## 46 4.2e-01 0.00944 3.3e-02 6.7e-02 7.1e-07 6.9e-03
## 47 2.2e-01 0.02849 1.9e-03 1.2e-01 2.2e-02 8.2e-03
## 48 4.2e-01 0.00944 3.3e-02 6.7e-02 7.1e-07 6.9e-03
## 49 5.6e-02 0.11118 1.4e-01 1.3e-01 5.1e-02 1.5e-01
## 50 5.6e-02 0.11118 1.4e-01 1.3e-01 5.1e-02 1.5e-01
## 51 3.6e-02 0.01584 1.2e-01 8.4e-04 2.0e-03 1.5e-01
## 52 4.2e-02 0.04410 3.1e-04 3.2e-02 1.3e-01 2.0e-01
## 53 1.6e-01 0.06777 5.1e-01 2.2e-02 6.9e-02 1.4e-02
## 54 1.1e-01 0.07162 8.5e-04 2.8e-02 1.3e-01 4.7e-02
## 55 1.0e-01 0.04762 5.0e-04 4.9e-02 3.3e-01 1.3e-01
## 56 5.5e-02 0.11624 2.4e-04 6.4e-02 1.6e-01 1.3e-01
## 57 6.4e-03 0.11280 6.3e-02 1.2e-01 1.9e-01 2.0e-01
## 58 5.5e-02 0.11624 2.4e-04 6.4e-02 1.6e-01 1.3e-01
## 59 9.3e-02 0.13062 8.1e-04 1.3e-02 3.6e-03 1.0e-01
## 60 3.7e-03 0.14654 8.9e-02 8.5e-02 5.1e-02 9.6e-02
## 61 3.3e-01 0.01233 1.2e-05 1.5e-01 1.5e-01 3.6e-03
## 62 3.2e-01 0.02570 2.6e-03 1.1e-01 3.3e-01 3.4e-02
## 63 2.4e-01 0.01427 2.8e-02 1.5e-01 3.3e-01 5.4e-04
## 64 2.1e-01 0.01231 3.6e-07 1.7e-01 3.8e-02 2.2e-02
## 65 4.2e-01 0.00944 3.3e-02 6.7e-02 7.1e-07 6.9e-03
## 66 2.0e-01 0.00510 1.0e-03 1.3e-03 4.3e-02 9.9e-03
## 67 1.2e-01 0.00059 3.7e-04 2.2e-03 3.2e-01 1.6e-02
## 68 1.1e-01 0.00520 7.5e-04 1.2e-06 1.5e-01 1.8e-05
## 69 1.7e-01 0.03095 3.7e-03 9.2e-02 1.6e-01 8.8e-02
## 70 1.7e-01 0.03095 3.7e-03 9.2e-02 1.6e-01 8.8e-02
## 71 1.0e-01 0.01898 8.4e-03 9.7e-02 1.5e-01 1.8e-02
## 72 1.7e-01 0.03095 3.7e-03 9.2e-02 1.6e-01 8.8e-02
## 73 4.3e-02 0.01523 1.1e-06 5.8e-02 4.1e-01 1.7e-02

```

```
## 74 6.3e-02 0.02833 6.9e-02 7.9e-02 4.7e-02 4.6e-02
## 75 4.6e-01 0.07838 6.6e-02 3.7e-04 3.1e-03 5.9e-02
## 76 5.6e-02 0.11118 1.4e-01 1.3e-01 5.1e-02 1.5e-01
## 77 5.8e-01 0.04998 3.5e-02 1.9e-04 2.5e-02 2.5e-01
## 78 2.1e-01 0.01911 3.4e-01 6.5e-02 9.0e-02 2.5e-01
## 79 6.0e-01 0.04486 8.1e-03 9.8e-06 7.2e-02 2.6e-01
## 80 1.9e-01 0.02330 2.3e-01 8.0e-02 2.5e-02 2.4e-01
## 81 5.7e-01 0.08031 1.7e-01 4.0e-03 1.8e-03 2.3e-02
## 82 5.9e-01 0.08885 9.6e-02 1.2e-03 1.7e-02 4.0e-02
## 83 2.1e-01 0.01911 3.4e-01 6.5e-02 9.0e-02 2.5e-01
## 84 4.6e-01 0.08541 8.0e-03 1.4e-03 6.0e-03 1.5e-02
## 85 2.1e-01 0.01911 3.4e-01 6.5e-02 9.0e-02 2.5e-01
## 86 2.2e-01 0.06260 3.6e-01 2.8e-02 1.9e-02 1.4e-01
## 87 6.6e-01 0.07804 1.7e-01 7.0e-03 5.6e-03 4.4e-02
## 88 1.2e-01 0.10879 3.3e-01 4.5e-02 2.7e-03 7.9e-02
## 89 2.2e-01 0.06260 3.6e-01 2.8e-02 1.9e-02 1.4e-01
## 90 5.4e-01 0.10782 2.3e-01 4.5e-03 3.0e-03 1.9e-02
## 91 5.2e-01 0.12832 2.0e-01 9.6e-03 4.7e-03 3.1e-03
## 92 2.0e-01 0.07835 2.1e-01 4.4e-02 1.3e-01 1.8e-01
## 93 2.2e-01 0.06260 3.6e-01 2.8e-02 1.9e-02 1.4e-01
## 94 2.3e-01 0.00904 3.8e-01 6.0e-02 6.9e-03 3.3e-03
## 95 2.2e-01 0.06260 3.6e-01 2.8e-02 1.9e-02 1.4e-01
## 96 2.2e-01 0.06260 3.6e-01 2.8e-02 1.9e-02 1.4e-01
## 97 6.4e-01 0.09511 1.4e-01 1.3e-02 3.7e-02 1.7e-02
## 98 4.4e-01 0.02334 5.4e-02 2.1e-02 3.7e-02 1.3e-03
## 99 5.7e-03 0.03754 3.7e-02 6.7e-02 6.6e-01 1.3e-03
## 100 5.7e-03 0.03754 3.7e-02 6.7e-02 6.6e-01 1.3e-03
## 101 5.7e-03 0.03754 3.7e-02 6.7e-02 6.6e-01 1.3e-03
## 102 5.7e-03 0.03754 3.7e-02 6.7e-02 6.6e-01 1.3e-03
## 103 6.4e-01 0.02552 5.9e-02 3.5e-02 1.4e-04 2.1e-02
## 104 2.0e-01 0.07835 2.1e-01 4.4e-02 1.3e-01 1.8e-01
## 105 2.2e-01 0.05553 1.8e-01 6.7e-02 3.3e-02 3.1e-01
## 106 2.0e-01 0.07835 2.1e-01 4.4e-02 1.3e-01 1.8e-01
## 107 6.4e-01 0.02552 5.9e-02 3.5e-02 1.4e-04 2.1e-02
## 108 4.4e-01 0.02334 5.4e-02 2.1e-02 3.7e-02 1.3e-03
## 109 2.5e-01 0.02060 2.5e-01 2.5e-02 5.1e-02 1.2e-03
```

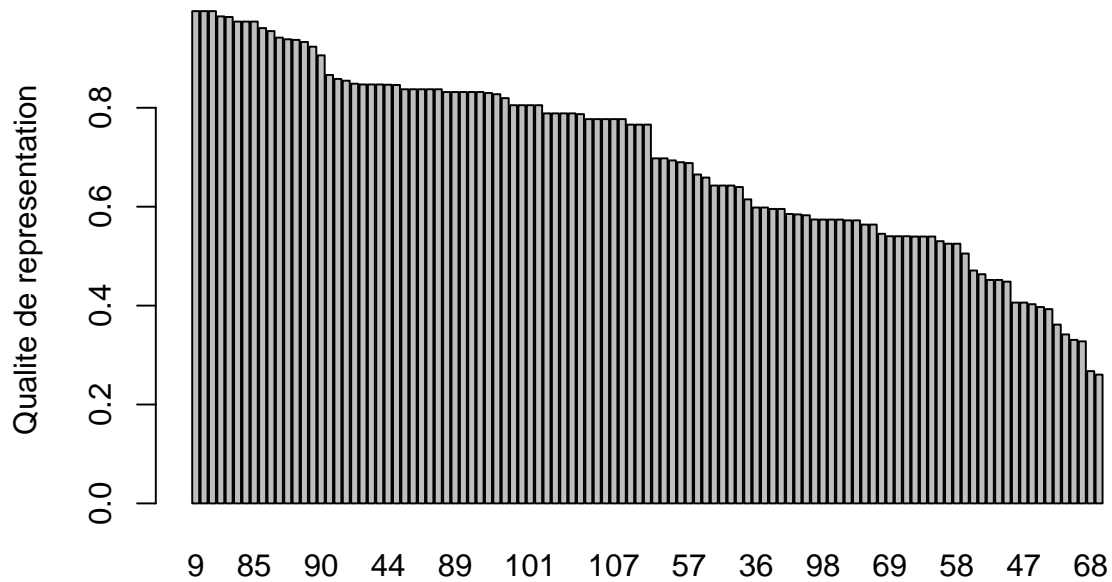
14-Distinguer les individus bien représentés, moyennement représentés et faiblement représentés sur le sous espace

```
head(t(apply(res$ind$cos2,1,cumsum)),digit=2)
```

```
##      Dim 1      Dim 2      Dim 3      Dim 4      Dim 5      Dim 6
## 1 0.02765330 0.1340798 0.3896791 0.5480094 0.5708189 0.9550835
## 2 0.01999611 0.1448603 0.3355883 0.7804967 0.7812754 0.7886643
## 3 0.01999611 0.1448603 0.3355883 0.7804967 0.7812754 0.7886643
## 4 0.01999611 0.1448603 0.3355883 0.7804967 0.7812754 0.7886643
## 5 0.01999611 0.1448603 0.3355883 0.7804967 0.7812754 0.7886643
## 6 0.06459885 0.9639085 0.9660501 0.9953184 0.9954446 0.9954693
```

```
cos_totale_ind<-sort(t(apply(res$ind$cos2,1,cumsum))[,6],decreasing=TRUE)
barplot(cos_totale_ind,ylab="Qualite de representation",main="Cos2 total d'individus jusqu'a Dim.6")
```

## Cos2 total d'individus jusqu'a Dim.6



L'individus bien représentées sont :

```
for (i in 1:109){
if (0.7 < cos_totale_ind[i] ) print(cos_totale_ind[i])
}
```

```
##          9
## 0.9954693
##          6
## 0.9954693
##         11
## 0.9954693
##         79
## 0.9847892
##         12
## 0.9835674
##         78
## 0.9742806
##         83
## 0.9742806
##         85
## 0.9742806
##         87
## 0.9611113
##          1
## 0.9550835
```

## 97  
## 0.9418381  
## 77  
## 0.9385409  
## 7  
## 0.9372704  
## 8  
## 0.9330932  
## 13  
## 0.9235466  
## 90  
## 0.9060437  
## 91  
## 0.8663628  
## 14  
## 0.8583019  
## 105  
## 0.8547326  
## 81  
## 0.848698  
## 92  
## 0.8471288  
## 104  
## 0.8471288  
## 106  
## 0.8471288  
## 44  
## 0.8466062  
## 53  
## 0.8460731  
## 32  
## 0.8374188  
## 16  
## 0.8374188  
## 15  
## 0.8374188  
## 19  
## 0.8374188  
## 31  
## 0.8374188  
## 86  
## 0.8318469  
## 89  
## 0.8318469  
## 93  
## 0.8318469  
## 95  
## 0.8318469  
## 96  
## 0.8318469  
## 82  
## 0.8299379  
## 62  
## 0.8274775

```

##          10
## 0.8195559
##          99
## 0.8052119
##          100
## 0.8052119
##          101
## 0.8052119
##          102
## 0.8052119
##          2
## 0.7886643
##          3
## 0.7886643
##          5
## 0.7886643
##          4
## 0.7886643
##          80
## 0.7871632
##          18
## 0.777078
##          33
## 0.777078
##          103
## 0.777078
##          107
## 0.777078
##          34
## 0.777078
##          21
## 0.7659442
##          63
## 0.7659442
##          20
## 0.7659442

```

L'individus moyennement représentées

```

for (i in 1:109){
if (0.5<cos_totale_ind[i]&& cos_totale_ind[i]<0.7) print(cos_totale_ind[i])
}

```

```

##          26
## 0.6977483
##          28
## 0.6977483
##          94
## 0.6935423
##          88
## 0.6898734
##          57
## 0.6881997

```



## 75  
## 0.6650894  
## 55  
## 0.6589083  
## 49  
## 0.6427504  
## 50  
## 0.6427504  
## 76  
## 0.6427504  
## 61  
## 0.6397451  
## 27  
## 0.6148098  
## 36  
## 0.598335  
## 35  
## 0.598335  
## 109  
## 0.5954136  
## 17  
## 0.5954136  
## 37  
## 0.5852683  
## 24  
## 0.5843992  
## 22  
## 0.5826585  
## 30  
## 0.5741058  
## 98  
## 0.573948  
## 108  
## 0.573948  
## 84  
## 0.5738721  
## 39  
## 0.5723264  
## 43  
## 0.5723264  
## 40  
## 0.563828  
## 45  
## 0.563828  
## 73  
## 0.5451773  
## 69  
## 0.5402383  
## 70  
## 0.5402383  
## 72  
## 0.5402383  
## 46  
## 0.5395974

```
##          48
## 0.5395974
##          65
## 0.5395974
##          29
## 0.5302028
##          56
## 0.5251165
##          58
## 0.5251165
##          25
## 0.5052812
```

L'individus faiblement représentées

```
for (i in 1:109){
if ( cos_totale_ind[i]< 0.5) print(cos_totale_ind[i])
}
```

```
##          60
## 0.4708872
##          67
## 0.4634076
##          64
## 0.4518594
##          23
## 0.4518594
##          52
## 0.4484461
##          42
## 0.4060905
##          47
## 0.4060905
##          41
## 0.4028038
##          71
## 0.3970424
##          54
## 0.3928598
##          38
## 0.3615519
##          59
## 0.3416933
##          74
## 0.3306338
##          51
## 0.3277076
##          68
## 0.267464
##          66
## 0.2604359
```

15-Calculer la contribution des individus dans chaque axe du sous espace

```
contrib_ind<-data.frame(res$ind$contrib)
print(res$ind$contrib,digit=2)
```

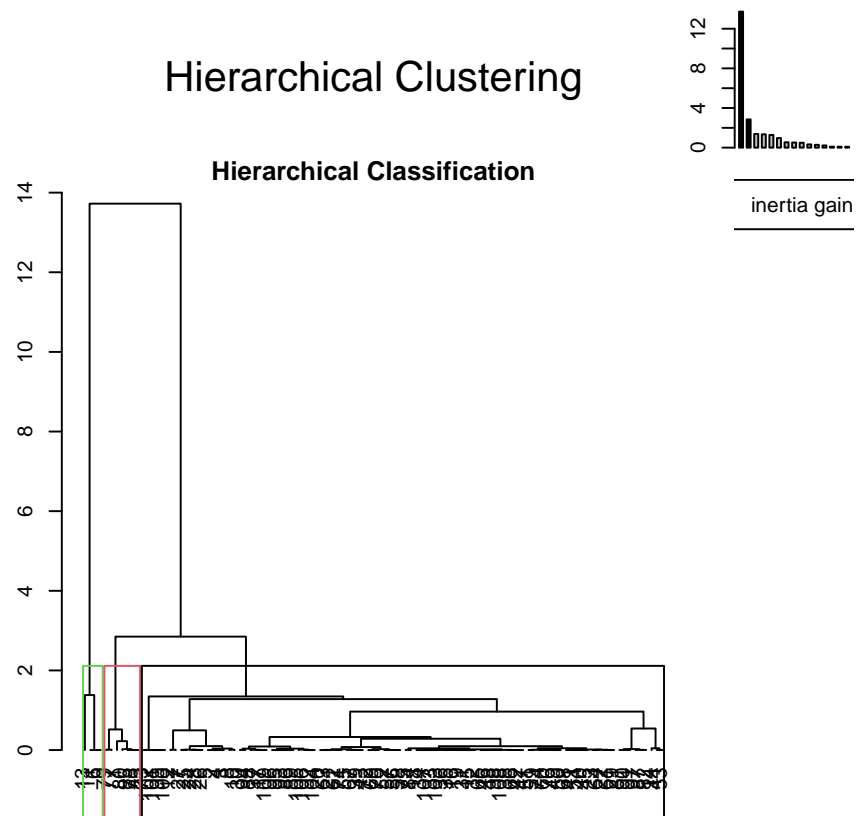
##	Dim 1	Dim 2	Dim 3	Dim 4	Dim 5	Dim 6
## 1	0.20744	0.8226	3.5e+00	2.5e+00	4.5e-01	1.0e+01
## 2	0.07820	0.5032	1.4e+00	3.7e+00	8.1e-03	1.0e-01
## 3	0.07820	0.5032	1.4e+00	3.7e+00	8.1e-03	1.0e-01
## 4	0.07820	0.5032	1.4e+00	3.7e+00	8.1e-03	1.0e-01
## 5	0.07820	0.5032	1.4e+00	3.7e+00	8.1e-03	1.0e-01
## 6	1.54912	22.2211	9.4e-02	1.5e+00	8.0e-03	2.1e-03
## 7	1.91361	0.2838	2.2e+00	7.9e+00	7.0e-01	1.2e-02
## 8	0.00359	0.6134	9.1e-01	4.0e+00	2.3e+00	1.9e-01
## 9	1.54912	22.2211	9.4e-02	1.5e+00	8.0e-03	2.1e-03
## 10	0.00761	0.7420	7.2e-01	3.5e+00	1.0e+00	6.7e-01
## 11	1.54912	22.2211	9.4e-02	1.5e+00	8.0e-03	2.1e-03
## 12	1.83884	0.2066	1.9e+00	8.5e+00	1.8e+00	7.2e-02
## 13	0.00096	11.9864	4.1e+00	1.0e+01	2.1e-01	9.2e-01
## 14	0.10946	0.3079	7.8e+00	3.9e-01	8.2e-02	6.5e-01
## 15	1.22075	0.0231	3.3e-01	2.5e-01	2.7e-01	9.5e-05
## 16	1.22075	0.0231	3.3e-01	2.5e-01	2.7e-01	9.5e-05
## 17	0.67018	0.0574	1.2e+00	1.4e-01	3.6e-01	1.1e-02
## 18	1.28181	0.0529	2.2e-01	1.5e-01	7.4e-04	1.5e-01
## 19	1.22075	0.0231	3.3e-01	2.5e-01	2.7e-01	9.5e-05
## 20	0.71799	0.0433	1.5e-01	9.5e-01	2.5e+00	5.7e-03
## 21	0.71799	0.0433	1.5e-01	9.5e-01	2.5e+00	5.7e-03
## 22	0.54089	0.0699	1.0e-02	7.5e-01	1.0e+00	7.1e-01
## 23	0.58178	0.0347	1.8e-06	9.6e-01	2.8e-01	2.2e-01
## 24	0.20229	0.1836	1.5e-01	2.5e+00	1.3e+00	1.2e-01
## 25	0.22759	0.1227	7.9e-02	2.9e+00	4.4e-01	1.1e-03
## 26	0.53698	0.2454	2.8e-02	1.7e+00	1.1e+00	2.0e-01
## 27	0.57772	0.1741	7.3e-02	2.0e+00	3.2e-01	5.4e-03
## 28	0.53698	0.2454	2.8e-02	1.7e+00	1.1e+00	2.0e-01
## 29	0.46297	0.0449	1.6e-01	1.1e+00	1.2e+00	1.3e-01
## 30	0.21971	0.0945	3.0e+00	1.7e-04	4.1e-01	9.7e-02
## 31	1.22075	0.0231	3.3e-01	2.5e-01	2.7e-01	9.5e-05
## 32	1.22075	0.0231	3.3e-01	2.5e-01	2.7e-01	9.5e-05
## 33	1.28181	0.0529	2.2e-01	1.5e-01	7.4e-04	1.5e-01
## 34	1.28181	0.0529	2.2e-01	1.5e-01	7.4e-04	1.5e-01
## 35	1.10217	0.0418	4.9e-03	9.4e-02	3.7e-01	2.4e-01
## 36	1.10217	0.0418	4.9e-03	9.4e-02	3.7e-01	2.4e-01
## 37	0.67545	0.0479	1.3e+00	5.3e-02	3.9e-01	1.4e-02
## 38	0.32194	0.0772	1.8e-01	2.3e-01	5.2e-02	5.0e-01
## 39	0.23309	0.0898	1.1e-03	2.2e-01	1.7e+00	1.0e-01
## 40	0.42214	0.0921	6.1e-04	3.1e-01	6.6e-01	3.6e-01
## 41	0.09974	0.1581	1.0e-01	4.6e-01	6.5e-01	9.6e-02
## 42	0.38742	0.0507	6.1e-03	4.4e-01	1.0e-01	5.1e-02
## 43	0.23309	0.0898	1.1e-03	2.2e-01	1.7e+00	1.0e-01
## 44	0.66144	0.3948	5.3e+00	2.0e-01	1.3e+00	1.1e+00
## 45	0.42214	0.0921	6.1e-04	3.1e-01	6.6e-01	3.6e-01
## 46	0.77605	0.0178	1.1e-01	2.6e-01	3.4e-06	4.5e-02
## 47	0.38742	0.0507	6.1e-03	4.4e-01	1.0e-01	5.1e-02
## 48	0.77605	0.0178	1.1e-01	2.6e-01	3.4e-06	4.5e-02
## 49	0.11791	0.2410	5.6e-01	5.9e-01	2.8e-01	1.1e+00

## 50	0.11791	0.2410	5.6e-01	5.9e-01	2.8e-01	1.1e+00
## 51	0.08740	0.0394	5.3e-01	4.3e-03	1.3e-02	1.3e+00
## 52	0.09367	0.1021	1.3e-03	1.5e-01	7.8e-01	1.6e+00
## 53	0.98017	0.4202	5.6e+00	2.8e-01	1.1e+00	3.1e-01
## 54	0.25027	0.1613	3.4e-03	1.3e-01	7.6e-01	3.7e-01
## 55	0.22371	0.1046	2.0e-03	2.2e-01	1.9e+00	9.6e-01
## 56	0.16457	0.3564	1.3e-03	4.0e-01	1.3e+00	1.3e+00
## 57	0.01427	0.2594	2.6e-01	5.7e-01	1.1e+00	1.6e+00
## 58	0.16457	0.3564	1.3e-03	4.0e-01	1.3e+00	1.3e+00
## 59	0.29386	0.4234	4.7e-03	8.8e-02	3.0e-02	1.1e+00
## 60	0.00849	0.3452	3.7e-01	4.1e-01	3.1e-01	7.8e-01
## 61	0.86429	0.0332	5.6e-05	8.1e-01	1.0e+00	3.3e-02
## 62	0.81428	0.0679	1.2e-02	6.2e-01	2.3e+00	3.1e-01
## 63	0.71799	0.0433	1.5e-01	9.5e-01	2.5e+00	5.7e-03
## 64	0.58178	0.0347	1.8e-06	9.6e-01	2.8e-01	2.2e-01
## 65	0.77605	0.0178	1.1e-01	2.6e-01	3.4e-06	4.5e-02
## 66	0.49952	0.0131	4.8e-03	6.7e-03	2.9e-01	8.8e-02
## 67	0.32164	0.0016	1.8e-03	1.2e-02	2.3e+00	1.5e-01
## 68	0.29143	0.0140	3.6e-03	6.4e-06	1.0e+00	1.7e-04
## 69	0.80236	0.1499	3.2e-02	9.1e-01	1.9e+00	1.5e+00
## 70	0.80236	0.1499	3.2e-02	9.1e-01	1.9e+00	1.5e+00
## 71	0.49214	0.0931	7.4e-02	9.8e-01	1.9e+00	3.1e-01
## 72	0.80236	0.1499	3.2e-02	9.1e-01	1.9e+00	1.5e+00
## 73	0.22451	0.0828	1.0e-05	6.5e-01	5.8e+00	3.1e-01
## 74	0.33898	0.1574	6.8e-01	9.0e-01	6.6e-01	8.8e-01
## 75	2.31268	0.4076	6.1e-01	4.0e-03	4.2e-02	1.1e+00
## 76	0.11791	0.2410	5.6e-01	5.9e-01	2.8e-01	1.1e+00
## 77	5.50969	0.4902	6.1e-01	3.8e-03	6.3e-01	8.5e+00
## 78	1.28485	0.1196	3.8e+00	8.3e-01	1.4e+00	5.4e+00
## 79	5.90295	0.4549	1.5e-01	2.0e-04	1.9e+00	9.1e+00
## 80	1.10499	0.1380	2.4e+00	9.7e-01	3.9e-01	4.9e+00
## 81	3.58901	0.5240	2.0e+00	5.3e-02	3.1e-02	5.2e-01
## 82	4.77915	0.7470	1.4e+00	2.1e-02	3.7e-01	1.2e+00
## 83	1.28485	0.1196	3.8e+00	8.3e-01	1.4e+00	5.4e+00
## 84	2.48309	0.4774	8.0e-02	1.6e-02	8.6e-02	3.0e-01
## 85	1.28485	0.1196	3.8e+00	8.3e-01	1.4e+00	5.4e+00
## 86	0.67295	0.1971	2.0e+00	1.8e-01	1.5e-01	1.5e+00
## 87	4.24933	0.5185	2.0e+00	9.6e-02	9.6e-02	1.0e+00
## 88	0.31961	0.2912	1.6e+00	2.5e-01	1.8e-02	7.4e-01
## 89	0.67295	0.1971	2.0e+00	1.8e-01	1.5e-01	1.5e+00
## 90	3.26306	0.6654	2.5e+00	5.7e-02	4.8e-02	4.0e-01
## 91	3.16520	0.7991	2.2e+00	1.2e-01	7.6e-02	6.6e-02
## 92	0.54466	0.2205	1.1e+00	2.5e-01	9.5e-01	1.8e+00
## 93	0.67295	0.1971	2.0e+00	1.8e-01	1.5e-01	1.5e+00
## 94	1.01763	0.0406	3.0e+00	5.5e-01	8.0e-02	5.1e-02
## 95	0.67295	0.1971	2.0e+00	1.8e-01	1.5e-01	1.5e+00
## 96	0.67295	0.1971	2.0e+00	1.8e-01	1.5e-01	1.5e+00
## 97	4.13755	0.6372	1.7e+00	1.8e-01	6.5e-01	3.9e-01
## 98	0.93170	0.0512	2.1e-01	9.2e-02	2.1e-01	1.0e-02
## 99	0.02032	0.1375	2.4e-01	5.0e-01	6.2e+00	1.6e-02
## 100	0.02032	0.1375	2.4e-01	5.0e-01	6.2e+00	1.6e-02
## 101	0.02032	0.1375	2.4e-01	5.0e-01	6.2e+00	1.6e-02
## 102	0.02032	0.1375	2.4e-01	5.0e-01	6.2e+00	1.6e-02
## 103	1.28181	0.0529	2.2e-01	1.5e-01	7.4e-04	1.5e-01

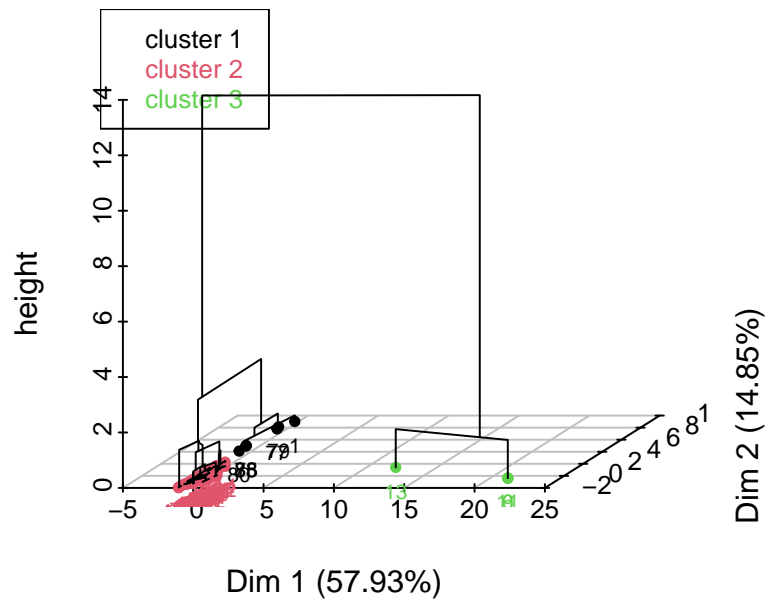
```
## 104 0.54466 0.2205 1.1e+00 2.5e-01 9.5e-01 1.8e+00
## 105 0.58569 0.1532 8.6e-01 3.8e-01 2.3e-01 2.9e+00
## 106 0.54466 0.2205 1.1e+00 2.5e-01 9.5e-01 1.8e+00
## 107 1.28181 0.0529 2.2e-01 1.5e-01 7.4e-04 1.5e-01
## 108 0.93170 0.0512 2.1e-01 9.2e-02 2.1e-01 1.0e-02
## 109 0.67018 0.0574 1.2e+00 1.4e-01 3.6e-01 1.1e-02
```

16-Application du CAH au tableau des contributions des individus aux axes du sous espace

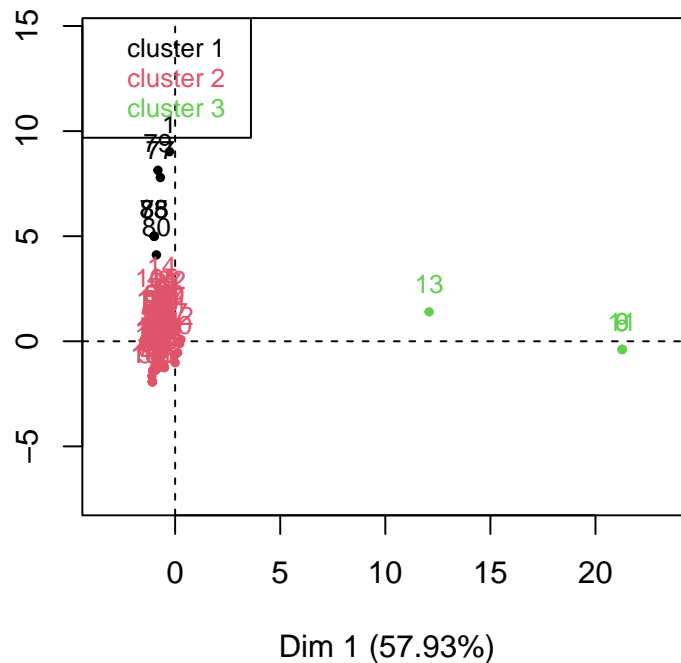
```
cah_ind<-HCPC(contrib_ind,nb.clust=-1)
```



## Hierarchical clustering on the factor map



## Factor map



```
cah_ind$desc.var
```

```
##
## Link between the cluster variable and the quantitative variables
## =====
##          Eta2      P-value
## Dim.2 0.9464357 4.269904e-68
## Dim.6 0.8020207 5.258388e-38
## Dim.1 0.1201343 1.132582e-03
## Dim.4 0.1091069 2.191573e-03
## Dim.3 0.1046698 2.851752e-03
##
## Description of each cluster by quantitative variables
## =====
## $'1'
##          v.test Mean in category Overall mean sd in category Overall sd
## Dim.6 9.301739          6.977878    0.9174312    2.081329    1.773787
## Dim.1 3.546896          2.368517    0.9174312    2.143061    1.113796
## Dim.3 3.331647          2.588367    0.9174312    1.476074    1.365405
##          p.value
## Dim.6 1.381675e-20
## Dim.1 3.897980e-04
## Dim.3 8.633364e-04
##
## $'2'
```

```
##          v.test Mean in category Overall mean sd in category Overall sd
## Dim.4 -2.055279      0.8097077    0.9174312    1.4114304    1.625805
## Dim.3 -2.886975      0.7903515    0.9174312    1.2551362    1.365405
## Dim.1 -3.164710      0.8037965    0.9174312    0.9314676    1.113796
## Dim.2 -5.960497      0.1947533    0.9174312    0.1907586    3.760892
## Dim.6 -7.080237      0.5125566    0.9174312    0.6139877    1.773787
##          p.value
## Dim.4 3.985210e-02
## Dim.3 3.889646e-03
## Dim.1 1.552377e-03
## Dim.2 2.514726e-09
## Dim.6 1.439080e-12
##
## $'3'
##          v.test Mean in category Overall mean sd in category Overall sd
## Dim.2 10.109771      19.662411    0.9174312    4.431771    3.760892
## Dim.4 3.432029       3.668315    0.9174312    3.782603    1.625805
##          p.value
## Dim.2 5.000116e-24
## Dim.4 5.990833e-04
```

#les ind de '\$1' contribuent à l'axe 5

#les ind de '\$2' ne contribuent ni à l'axe 1, ni à l'axe 2, ni à l'axe 4, ni à l'axe 5 et ni à l'axe 6.

#les ind de '\$3' contribuent à l'axe 4

#les ind de '\$4' contribuent à l'axe 6 et à l'axe 1

#les ind de '\$5' contribuent à les axes 2 ,3 et 4 .

#les ind de '\$6' contribuent à les axes 1 et 2.

### Nuage des variables

17-Calculer les coefficients de corrélation des variables avec les projections sur les axes du sous espace

```
res$var$eta2
```

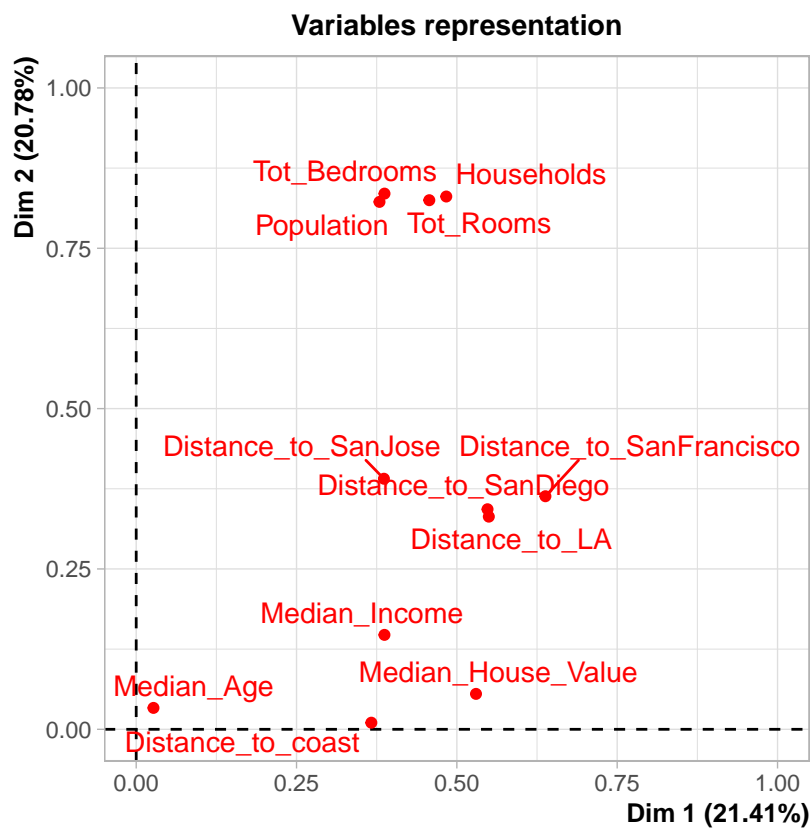
```
##          Dim 1      Dim 2      Dim 3      Dim 4
## Median_House_Value 0.52985487 0.05521745 0.11060677 0.01169693
## Median_Income      0.38708843 0.14723639 0.08108647 0.05274093
## Median_Age         0.02694424 0.03337307 0.31859442 0.03654916
## Tot_Rooms          0.45700738 0.82495458 0.38270089 0.17984830
## Tot_Bedrooms       0.38716016 0.83522636 0.39751138 0.13588159
## Population         0.37934388 0.82233092 0.40736854 0.18990875
## Households         0.48349605 0.83070679 0.41711378 0.16914025
## Distance_to_coast  0.36664987 0.01027285 0.11763909 0.05567292
## Distance_to_LA     0.54982351 0.33157357 0.13038516 0.47220013
## Distance_to_SanDiego 0.54780846 0.34310443 0.13121021 0.48659807
## Distance_to_SanJose 0.38628549 0.39064390 0.13404031 0.41217716
## Distance_to_SanFrancisco 0.63807739 0.36337852 0.17553213 0.22767817
##          Dim 5      Dim 6
## Median_House_Value 0.286714834 0.562736617
## Median_Income      0.270039990 0.381633453
## Median_Age         0.266943246 0.098034158
## Tot_Rooms          0.001766034 0.025793644
```



```
## Tot_Bedrooms      0.005395423 0.006271400
## Population        0.007534597 0.006955846
## Households        0.001579612 0.013687637
## Distance_to_coast  0.389864894 0.078421785
## Distance_to_LA     0.204196793 0.102896031
## Distance_to_SanDiego 0.204898535 0.108588918
## Distance_to_SanJose 0.128197600 0.003740545
## Distance_to_SanFrancisco 0.171608054 0.050084343
```

18-Tracer le graphique des coefficients de corrélation des variables avec les projections sur le 1<sup>er</sup> plan factoriel

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



## 19-Conclusion

Après l'application de l'ACM, nous avons réduit les 36 modalités en 6 composantes. Cette réduction de dimensionnalité nous a permis de visualiser les modalités et les individus sur un plan factoriel et nous a permis de connaître les individus et les modalités qui contribuent le plus dans le sous espace obtenu.

On peut conclure que l'ACM a un rôle très important résumer l'information contenu dans un grand nombre de variables afin de faciliter l'interprétation des corrélations existantes entre ces différentes variables qui sont qualitatives.

# Bibliographie

- [1] <<https://stackoverflow.com/questions/11308367/error-in-my-code-object-of-type-closure-is-not-subset>>
- [2] <[http://factominer.free.fr/factomethods/index\\_fr.html](http://factominer.free.fr/factomethods/index_fr.html)>.