

# Réalisation d'une ACM

*Rémi Taniel*

*30/09/2019*

## Contents

<b>Introduction</b>	<b>2</b>
<b>1ère partie : Importation des données</b>	<b>2</b>
<b>2e partie : Mise en oeuvre de l'ACM</b>	<b>2</b>
<b>3e partie : Analyse des résultats</b>	<b>3</b>
Nombre d'axe à retenir . . . . .	3
Analyse des 5 premiers axes en fonction des modalités . . . . .	4
Analyse des dimensions . . . . .	7
Analyse des 5 premiers axes en fonction des individus . . . . .	7
Interprétation du premier plan factoriel . . . . .	10
<b>3e partie : Clasification ascendate hiérarchique sur les axes retenus</b>	<b>11</b>
Réalisation sous R . . . . .	11
Interprétation et analyse des résultats . . . . .	12

# Introduction

## 1ère partie : Importation des données

On commence par importer les données grâce à :

```
data <- read.table("/home/remi/Documents/Cours/AD/data/race.csv", sep = ",", dec = ".", colClasses = "f
```

Puis on visualise les données grâce à la fonction `str(...)` :

```
str(data)

## 'data.frame': 27 obs. of 8 variables:
## $ Race : Factor w/ 27 levels "B.AL","BASS",...: 3 2 1 4 5 6 7 8 9 10 ...
## $ Taille : Factor w/ 3 levels "1","2","3": 3 1 3 2 1 3 1 1 2 3 ...
## $ Poids : Factor w/ 3 levels "1","2","3": 2 1 2 2 1 3 1 1 1 2 ...
## $ Velocite : Factor w/ 3 levels "1","2","3": 3 1 3 2 1 1 2 1 1 3 ...
## $ Intelligence: Factor w/ 3 levels "1","2","3": 3 1 3 2 2 3 3 1 2 2 ...
## $ Affection : Factor w/ 2 levels "1","2": 2 1 2 2 2 1 2 2 2 2 ...
## $ Agressivite : Factor w/ 2 levels "1","2": 2 2 2 2 1 2 1 1 2 1 ...
## $ Fonction : Factor w/ 3 levels "1","2","3": 3 2 3 1 1 3 1 1 1 1 ...
```

Notre jeu de données comporte 27 observations (27 races de chiens) et 8 variables (les 7 modalités et un libellé), on remarque également que toutes les variables sont des variables qualitatives, et ont 3 niveaux chacune :

- Faible : 1
- Moyen : 2
- Fort : 3

On se décide de visualiser les 6 premières lignes de nos données ;

```
head(data)

## Race Taille Poids Velocite Intelligence Affection Agressivite Fonction
## 1 BEAU 3 2 3 3 2 2 3
## 2 BASS 1 1 1 1 1 2 2
## 3 B.AL 3 2 3 3 2 2 3
## 4 BOXE 2 2 2 2 2 2 1
## 5 BULD 1 1 1 2 2 1 1
## 6 BULM 3 3 1 3 1 2 3
```

On remarque que nous devons enlever la colonne `race`, on décide donc de formater nos données pour donner un identifiant aux différentes lignes, dans notre cas, ce sera la variable `race` :

```
rownames(data) <- data$Race
data <- data[,-1]
head(data)

## Taille Poids Velocite Intelligence Affection Agressivite Fonction
## BEAU 3 2 3 3 2 2 3
## BASS 1 1 1 1 1 2 2
## B.AL 3 2 3 3 2 2 3
## BOXE 2 2 2 2 2 2 1
## BULD 1 1 1 2 2 1 1
## BULM 3 3 1 3 1 2 3
```

## 2e partie : Mise en oeuvre de l'ACM

Pour réaliser l'ACM, nous aurons besoin du package `FactoMineR` :

```
library(FactoMineR)
```

Puis on range les résultats de l'ACM (valeurs propres, coordonnées, contribution) dans la variable `data.mca`, dans notre cas, nous ne retenons que les 5 premiers axes et on souhaite que les graphiques ne soient pas générés lors de l'appel de la fonction :

```
data.mca <- FactoMineR::MCA(data, ncp = 5, quali.sup = c(7), graph = FALSE)
```

(Explication de où se trouve les variables ?)

## 3e partie : Analyse des résultats

### Nombre d'axe à retenir

Pour connaître le nombre d'axe que nous devons retenir, nous pouvons utiliser 3 critères :

- Part d'inertie supérieure à la moyenne
- Part d'inertie cumulée supérieure à 80%
- Critère du coude

Pour rappel, les valeurs propres des différents axes sont stockés dans :

```
data.mca$eig
```

```
##          eigenvalue percentage of variance
## dim 1  0.487590671          29.2554403
## dim 2  0.385667752          23.1400651
## dim 3  0.220728360          13.2437016
## dim 4  0.164472270           9.8683362
## dim 5  0.148659735           8.9195841
## dim 6  0.101816275           6.1089765
## dim 7  0.081305114           4.8783069
## dim 8  0.044665793           2.6799476
## dim 9  0.024078978           1.4447387
## dim 10 0.007681718           0.4609031
##          cumulative percentage of variance
## dim 1          29.25544
## dim 2          52.39551
## dim 3          65.63921
## dim 4          75.50754
## dim 5          84.42713
## dim 6          90.53610
## dim 7          95.41441
## dim 8          98.09436
## dim 9          99.53910
## dim 10         100.00000
```

### Part d'inertie supérieure à la moyenne

Nous n'utiliserons qu'un seul critère, le critère de la part d'inertie moyenne, cette moyenne des part d'inertie expliquée par chaque axe peut être obtenue par :

```
100/16
```

```
## [1] 6.25
```

Selon ce critère, nous pouvons retenir les 5 premiers axes, qui possèdent tous une part d'inertie supérieure à la moyenne calculée qui est de 6.25%.

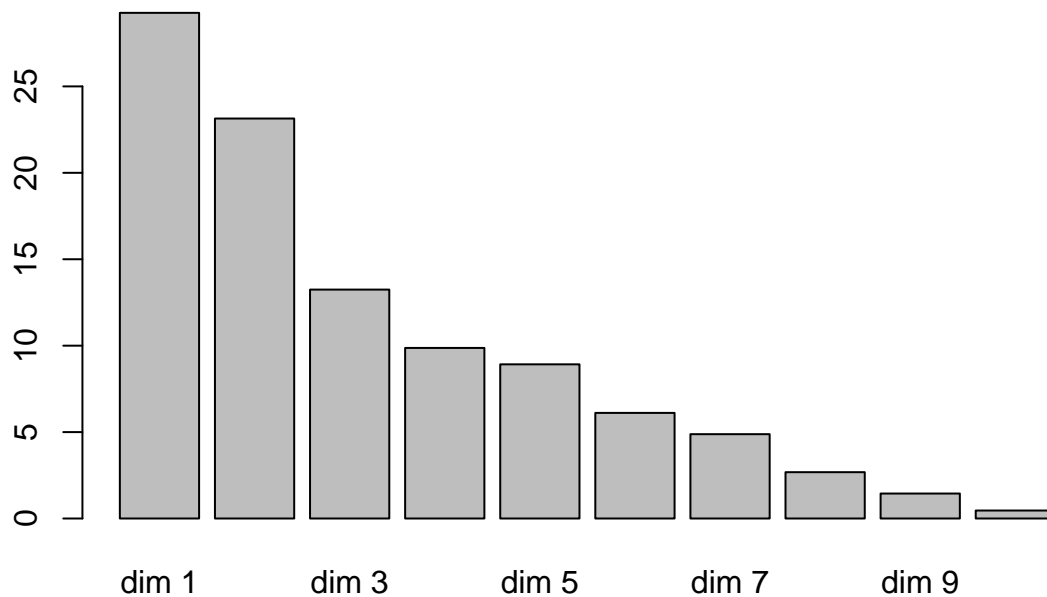
### Part d'inertie cumulée supérieure à 80%

Tout comme le précédent critère, nous retenons les 5 premiers axes, en effet, ces 5 dimensions expliquent 84,43% de l'inertie totale portée par nos données.

### Critère du coude

Afin d'appliquer ce critère, nous devons dans un premier temps, tracer le graphique suivant :

```
barplot(data.mca$eig[,2])
```



Le coude apparaît entre la 4e et 5e dimension, donc en utilisant ce critère nous devons retenir 4 dimensions.

### Conclusion sur le nombre d'axe à retenir

Selon les 3 critères, nous devons seulement retenir les 5 premiers axes.

### Analyse des 5 premiers axes en fonction des modalités

Pour obtenir les données des 5 premiers axes en fonction des modalités, on utilise l'information suivante :

```
data.mca$var
```

```
## $coord
##           Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## Taille_1      1.1902124  0.8814312493 -0.66619486  0.0245082  0.04370012
## Taille_2      0.8464733 -1.2369449391  0.98137849  0.4516478  0.32628695
## Taille_3     -0.8375902  0.0009803967 -0.01623523 -0.1619864 -0.12915570
## Poids_1       1.1761704  0.7857960033 -0.43503416  0.1695709  0.01837861
```

```

## Poids_2      -0.3293556 -0.8124958119 -0.18601927 -0.1564473  0.19161738
## Poids_3      -0.9596770  1.0177146680  1.21690862  0.1667392 -0.56593443
## Velocite_1    0.3536067  1.0417311970  0.35916042  0.0229601 -0.32924689
## Velocite_2    0.5993249 -0.8943513681  0.39416354  0.1983599  0.53176045
## Velocite_3   -0.9256296 -0.3625001139 -0.74943473 -0.2018311 -0.10684608
## Intelligence_1 -0.3298252  0.8162016046 -0.30430278 -0.1139304  1.11946622
## Intelligence_2  0.4660516 -0.2414724706  0.62739888 -0.4483245 -0.26091301
## Intelligence_3 -0.4220026 -0.5188490270 -0.72776633  0.8987625 -0.83211053
## Affection_1   -0.8082282  0.3173466033  0.17128331 -0.2363086  0.14920499
## Affection_2    0.7504976 -0.2946789888 -0.15904879  0.2194294 -0.13854750
## Agressivite_1  0.4080360 -0.1994591783 -0.17920983 -0.7002575 -0.22502503
## Agressivite_2 -0.4394234  0.2148021920  0.19299520  0.7541235  0.24233465
##
## $contrib
##           Dim 1           Dim 2           Dim 3           Dim 4           Dim 5
## Taille_1      12.553839  8.704555e+00  8.68815258  0.01578023  0.05550791
## Taille_2       4.535510  1.224454e+01  13.46697104  3.82791292  2.21034998
## Taille_3      13.322448  2.307632e-05  0.01105694  1.47720499  1.03898691
## Poids_1       14.010709  7.906445e+00  4.23412382  0.86334566  0.01122036
## Poids_2        1.922595  1.479251e+01  1.35478624  1.28604643  2.13446702
## Poids_3        5.829749  8.288840e+00  20.70679054  0.52172002  6.64957461
## Velocite_1     1.582963  1.736933e+01  3.60747890  0.01978518  4.50126929
## Velocite_2     3.637845  1.024184e+01  3.47591976  1.18138110  9.39320246
## Velocite_3     9.762175  1.892912e+00  14.13634026  1.37597681  0.42662991
## Intelligence_1  1.101761  8.530147e+00  2.07170863  0.38972813  41.62973133
## Intelligence_2  3.299734  1.119923e+00  13.20977953  9.05229126  3.39206631
## Intelligence_3  1.578185  3.016142e+00  10.36832982  21.22170349  20.12573312
## Affection_1    10.750805  2.095476e+00  1.06659687  2.72454975  1.20171562
## Affection_2     9.982890  1.945799e+00  0.99041137  2.52993905  1.11587879
## Agressivite_1   2.950901  8.914721e-01  1.25741474  25.76534276  2.94361715
## Agressivite_2   3.177893  9.600469e-01  1.35413895  27.74729221  3.17004923
##
## $cos2
##           Dim 1           Dim 2           Dim 3           Dim 4
## Taille_1      0.49581192  2.719224e-01  0.1553354552  0.0002102282
## Taille_2      0.16284478  3.477347e-01  0.2188872152  0.0463603889
## Taille_3      0.87694667  1.201472e-06  0.0003294784  0.0327995000
## Poids_1       0.58247448  2.599896e-01  0.0796861975  0.0121070633
## Poids_2       0.11681934  7.109302e-01  0.0372649528  0.0263585186
## Poids_3       0.20931365  2.353962e-01  0.3365605898  0.0063186241
## Velocite_1     0.07355161  6.383552e-01  0.0758801234  0.0003100978
## Velocite_2     0.15123804  3.367850e-01  0.0654167993  0.0165670095
## Velocite_3     0.42839511  6.570317e-02  0.2808262078  0.0203679027
## Intelligence_1  0.04580407  2.804990e-01  0.0389895501  0.0054653233
## Intelligence_2  0.17376329  4.664716e-02  0.3149034805  0.1607958967
## Intelligence_3  0.06233016  9.422151e-02  0.1853753391  0.2827209221
## Affection_1     0.60657339  9.351538e-02  0.0272424033  0.0518530622
## Affection_2     0.60657339  9.351538e-02  0.0272424033  0.0518530622
## Agressivite_1   0.17930056  4.284427e-02  0.0345866377  0.5280806741
## Agressivite_2   0.17930056  4.284427e-02  0.0345866377  0.5280806741
##
##           Dim 5
## Taille_1      0.0006683951
## Taille_2      0.0241961758
## Taille_3      0.0208514951

```

```

## Poids_1      0.0001422203
## Poids_2      0.0395416207
## Poids_3      0.0727913127
## Velocite_1   0.0637667735
## Velocite_2   0.1190607053
## Velocite_3   0.0057080420
## Intelligence_1 0.5276651063
## Intelligence_2 0.0544604772
## Intelligence_3 0.2423427783
## Affection_1  0.0206719783
## Affection_2  0.0206719783
## Agressivite_1 0.0545313614
## Agressivite_2 0.0545313614
##
## $v.test
##           Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## Taille_1      3.590419  2.658943687 -2.00965714  0.07393195  0.13182668
## Taille_2      2.057660 -3.006842664  2.38559586  1.09789349  0.79315860
## Taille_3     -4.774999  0.005589121 -0.09255506 -0.92346467 -0.73630080
## Poids_1       3.891572  2.599948123 -1.43938915  0.56105583  0.06080894
## Poids_2     -1.742786 -4.299323719 -0.98432148 -0.82784146  1.01394385
## Poids_3     -2.332843  2.473924090  2.95813714  0.40531991 -1.37570859
## Velocite_1    1.382874  4.073970535  1.40459361  0.08979167 -1.28760868
## Velocite_2    1.982975 -2.959123171  1.30416133  0.65630957  1.75942557
## Velocite_3   -3.337405 -1.307012748 -2.70212535 -0.72771249 -0.38523901
## Intelligence_1 -1.091286  2.700550551 -1.00684075 -0.37695942  3.70395637
## Intelligence_2  2.125522 -1.101283907  2.86137912 -2.04467438 -1.18994639
## Intelligence_3 -1.273022 -1.565170677 -2.19539491  2.71122555 -2.51016179
## Affection_1   -3.971260  1.559294642  0.84160708 -1.16111137  0.73312443
## Affection_2    3.971260 -1.559294642 -0.84160708  1.16111137 -0.73312443
## Agressivite_1  2.159124 -1.055438765 -0.94828929 -3.70541462 -1.19072054
## Agressivite_2 -2.159124  1.055438765  0.94828929  3.70541462  1.19072054
##
## $eta2
##           Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## Taille      0.8897105  0.48476391  0.29356228  0.05250841  0.02947784
## Poids      0.6366877  0.71705966  0.34825241  0.02635943  0.07845008
## Velocite   0.4383338  0.68272633  0.28102789  0.02543211  0.12773827
## Intelligence 0.1749381  0.29309697  0.33969853  0.30259993  0.58108888
## Affection  0.6065734  0.09351538  0.02724240  0.05185306  0.02067198
## Agressivite 0.1793006  0.04284427  0.03458664  0.52808067  0.05453136

```

Pour chacune des 4 dimensions, nous allons retenir les modalités dont la contribution est supérieure à la moyenne (soit 6.25), puis pour chacune des modalités retenues, nous allons noter la qualité de leur représentation sous cet axe, ainsi que le signe de ses coordonnées.

## Dimension 1

Modalité	Contribution	Qualité	Signe
Poids_1	14.01	0.58	+
Taille_3	13.32	0.87	-
Taille_1	12.56	0.49	+
Affection_1	10.75	0.60	-
Affection_2	9.98	0.60	+

Modalité	Contribution	Qualité	Signe
Velocite_3	9.76	0.43	-
Somme	70.38		

## Dimension 2

Modalité	Contribution	Qualité	Signe
Velocite_1	17.36	0.64	+
Poids_2	14.89	0.71	-
Taille_2	12.24	0.34	-
Velocite_2	10.24	0.34	-
Taille_1	8.70	0.27	+
Intelligence_1	8.53	0.28	+
Somme	71.96		

## Dimension 3

Modalité	Contribution	Qualité	Signe
Poids_3	20.71	0.37	+
Velocite_3	14.14	0.28	-
Intelligence_2	13.21	0.32	-
Intelligence_3	10.37	0.18	+
Taille_1	8.68	0.16	-
Somme	67.44		

## Dimension 4

Modalité	Contribution	Qualité	Signe
Agressivite_2	27.74	0.53	
Agressivite_1	25.76	0.53	
Intelligence_3	21.22	0.28	
Intelligence_2	9.01	0.16	
Somme	83.73		

## Analyse des dimensions

### Analyse des 5 premiers axes en fonction des individus

Nous allons reprendre la même méthode que précédemment mais en l'appliquant aux individus et non plus aux modalités, dans ce cas on utilisera les données suivantes :

```
data.mca$ind
```

```
## $coord
##      Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## BEAU -0.5259384 -0.47575949 -0.58374071  0.55564235 -0.33401607
## BASS  0.2726981  1.08888034 -0.24197092  0.25517645  0.53766998
## B.AL -0.5259384 -0.47575949 -0.58374071  0.55564235 -0.33401607
## BOXE  0.4519622 -0.87628238  0.65659133  0.41868418  0.38581511
## BULD  1.0369751  0.52959725 -0.16067532 -0.29265208 -0.38543246
```

```

## BULM -0.7430945 0.55653626 0.42440108 0.59354942 -0.63323133
## CANI 0.8836606 -0.06443988 -0.62899988 0.33303322 -0.26015724
## CHIH 0.8470130 0.81345055 -0.49119441 -0.15522858 0.21125992
## COCK 0.7526566 0.07225541 0.55583657 0.48058272 -0.06125534
## COLL -0.1117012 -0.51249596 -0.23503778 -0.59565654 -0.28913039
## DAL 0.6542361 -0.98745978 0.52455242 -0.17901219 0.18379095
## DOBE -0.8979794 -0.31150711 -0.46655613 0.36835102 -0.20963017
## DA.L -1.0264251 0.53796989 0.18135217 0.08498958 0.30650741
## E.BR 0.4422729 -1.06190072 0.04381058 0.37459034 -0.06311888
## E.FR -0.1197618 -0.49097917 0.28783575 -0.61848428 0.11130403
## FX.H -0.8759782 0.04678712 -0.31633338 -0.04782801 0.63397205
## FX.T 0.8933499 0.12117846 -0.01621913 0.37712706 0.18877675
## GBLG -0.5119978 -0.09594847 0.08935556 0.11663557 0.91002057
## LABR 0.6542361 -0.98745978 0.52455242 -0.17901219 0.18379095
## LEVR -0.6737043 -0.06439028 -0.44837229 -0.64552437 0.43194788
## MAST -0.7210934 0.91483050 0.57462384 0.17737039 0.21037088
## PEKI 0.8470130 0.81345055 -0.49119441 -0.15522858 0.21125992
## POIN -0.6957054 -0.42268452 -0.59859504 -0.22934534 -0.41165433
## ST-B -0.5311313 0.63097720 0.90514293 0.03994690 -0.38632150
## SETT -0.4837422 -0.34824358 -0.11785320 -0.78294787 -0.16474449
## TECK 1.0369751 0.52959725 -0.16067532 -0.29265208 -0.38543246
## T.NE -0.3288574 0.51979980 0.77310402 -0.55774946 -0.58834566
##
## $contrib
## Dim 1 Dim 2 Dim 3 Dim 4 Dim 5
## BEAU 2.10111844 2.17369113 5.71765657 6.95239166 2.77956990
## BASS 0.56486551 11.38631564 0.98243645 1.46630641 7.20235393
## B.AL 2.10111844 2.17369113 5.71765657 6.95239166 2.77956990
## BOXE 1.55161864 7.37413469 7.23383138 3.94745005 3.70852623
## BULD 8.16803372 2.69348375 0.43318803 1.92862044 3.70117360
## BULM 4.19439182 2.97447169 3.02225475 7.93336291 9.99005033
## CANI 5.93132626 0.03987790 6.63864170 2.49757571 1.68622192
## CHIH 5.44955505 6.35455623 4.04841413 0.54260890 1.11192907
## COCK 4.30303195 0.05013761 5.18408763 5.20091611 0.09348260
## COLL 0.09477557 2.52234098 0.92694388 7.98979010 2.08271755
## DAL 3.25124622 9.36400725 4.61695759 0.72162030 0.84157173
## DOBE 6.12510519 0.93187754 3.65246368 3.05539455 1.09483944
## DA.L 8.00267820 2.77932193 0.55185326 0.16265782 2.34058723
## E.BR 1.48580379 10.82905948 0.03220594 3.15977886 0.09925708
## E.FR 0.10894759 2.31498957 1.39016885 8.61392102 0.30864885
## FX.H 5.82864293 0.02102208 1.67906747 0.05151197 10.01343537
## FX.T 6.06211255 0.14101775 0.00441401 3.20271975 0.88785063
## GBLG 1.99121023 0.08840951 0.13397424 0.30634135 20.63218850
## LABR 3.25124622 9.36400725 4.61695759 0.72162030 0.84157173
## LEVR 3.44762033 0.03981654 3.37330515 9.38358594 4.64842239
## MAST 3.94969799 8.03718894 5.54045425 0.70844447 1.10259017
## PEKI 5.44955505 6.35455623 4.04841413 0.54260890 1.11192907
## POIN 3.67647489 1.71575624 6.01235105 1.18446820 4.22190300
## ST-B 2.14281499 3.82340445 13.74714214 0.03593434 3.71826755
## SETT 1.77749654 1.16463313 0.23305630 13.80413634 0.67618368
## TECK 8.16803372 2.69348375 0.43318803 1.92862044 3.70117360
## T.NE 0.82147815 2.59474760 10.02891524 7.00522151 8.62398496
##
## $cos2

```



```
##          Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## BEAU 0.19318200 0.158078170 0.2379780727 0.2156193132 0.077916905
## BASS 0.03893080 0.620711031 0.0306517673 0.0340886922 0.151342440
## B.AL 0.19318200 0.158078170 0.2379780727 0.2156193132 0.077916905
## BOXE 0.11183606 0.420403058 0.2360304269 0.0959733835 0.081495980
## BULD 0.64266568 0.167625423 0.0154293198 0.0511860542 0.088786106
## BULM 0.27815801 0.156023623 0.0907311457 0.1774668333 0.201989233
## CANI 0.38024293 0.002022086 0.1926598936 0.0540088961 0.032958090
## CHIH 0.38556757 0.355617092 0.1296663063 0.0129498182 0.023985817
## COCK 0.28975347 0.002670396 0.1580262244 0.1181329992 0.001919211
## COLL 0.01095173 0.230541033 0.0484889401 0.3114290851 0.073376137
## DAL  0.23755593 0.541172466 0.1527125209 0.0177853352 0.018747576
## DOBE 0.55359781 0.066618921 0.1494408973 0.0931505474 0.030169577
## DA.L 0.53895247 0.148051191 0.0168244731 0.0036951098 0.048059352
## E.BR 0.09451162 0.544844311 0.0009273902 0.0677981326 0.001924967
## E.FR 0.01169406 0.196541690 0.0675488576 0.3118788278 0.010100673
## FX.H 0.55756256 0.001590595 0.0727104867 0.0016621553 0.292043282
## FX.T 0.44081779 0.008110859 0.0001453019 0.0785581701 0.019683984
## GBLG 0.18220281 0.006398746 0.0055496040 0.0094554212 0.575600583
## LABR 0.23755593 0.541172466 0.1527125209 0.0177853352 0.018747576
## LEVR 0.33582962 0.003067758 0.1487503335 0.3083228462 0.138052133
## MAST 0.27298068 0.439369729 0.1733469229 0.0165162378 0.023233792
## PEKI 0.38556757 0.355617092 0.1296663063 0.0129498182 0.023985817
## POIN 0.33802415 0.124775596 0.2502437311 0.0367347283 0.118348388
## ST-B 0.16426904 0.231835114 0.4770745045 0.0009292189 0.086905976
## SETT 0.20103461 0.104185949 0.0119323415 0.5266336570 0.023316573
## TECK 0.64266568 0.167625423 0.0154293198 0.0511860542 0.088786106
## T.NE 0.06389477 0.159632893 0.3531230245 0.1837928020 0.204510346
```

Comme précédemment, pour chaque dimension, on ne garde que les races canines ayant une contribution supérieure à la moyenne qui est de :

```
mean(data.mca$ind$contrib[,1])
```

```
## [1] 3.703704
```

## Dimension 1

Individu	Contribution	Qualité	Signe
BULD	8.17		
TECK	8.17		
DA.L	8		
DOBE	6.13		
FX.T	6.06		
CANI	5.93		
CHIH	5.45		
PEKI	5.45		
FX.H	5.83		
COCK	4.30		
BULM	4.20		
Somme	67.69		

## Dimension 2

Individu	Contribution	Qualité	Signe
BASS	11.38		
E.BR	10.83		
LABR	9.36		
DAL	9.36		
MAST	8.04		
BOXE	7.37		
PEKI	6.35		
CHIH	6.35		
Somme	69.04		

### Dimension 3

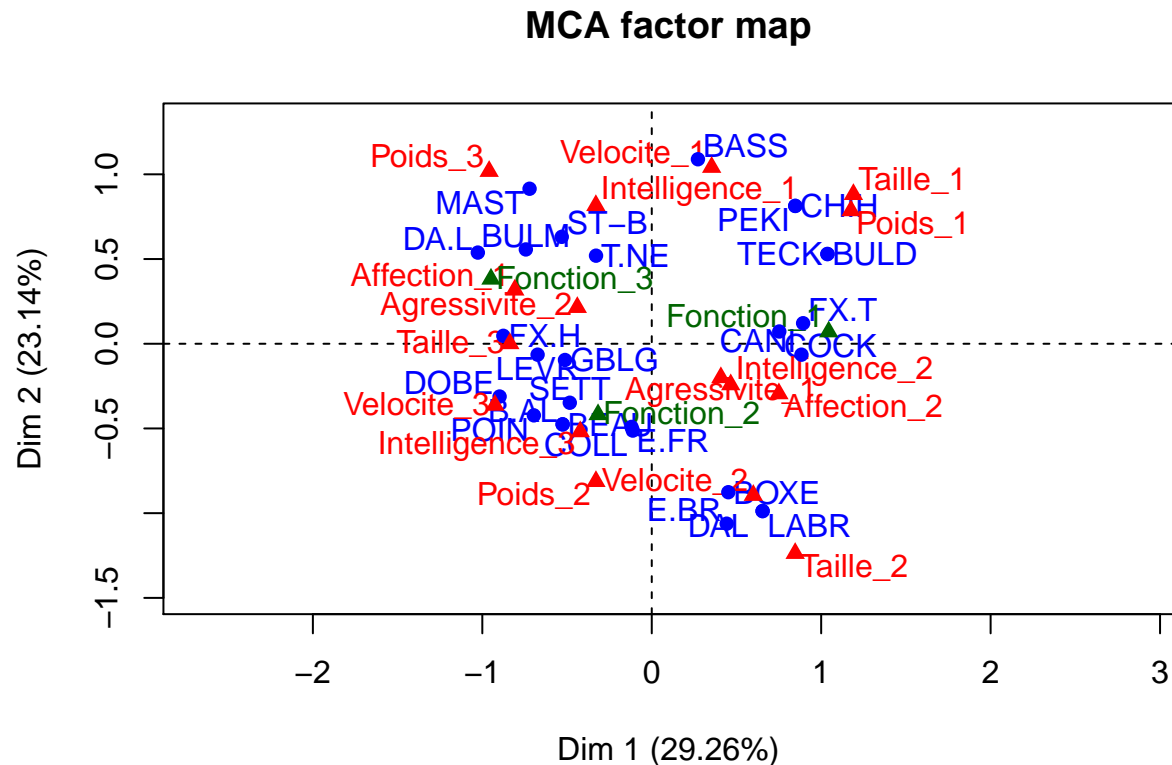
Individu	Contribution	Qualité	Signe
ST-B	13.75		
T.NE	10.03		
BOXE	7.23		
CANI	6.64		
POIN	6.01		
BEAU	5.72		
B.AL	5.72		
MAST	5.54		
COCK	5.18		
LABR	4.62		
Somme	70.44		

### Dimension 4

### Interprétation du premier plan factoriel

On peut obtenir les projections des marques et des attributs dans le premier plan factoriel grâce à :

```
plot.MCA(data.mca, axes = c(1, 2))
```



### 3e partie : Clasification ascendate hiérarchique sur les axes retenus

Pour rappel, nous avons retenus les 4 premiers axes factoriels qui représentent 75% de l'information totale. Notre classification se portera donc seulement sur ces axes.

#### Réalisation sous R

Pour notre étude, nous aurons besoins des librairies suivantes :

```
library(plyr)
library(philentropy)
library(factoextra)
```

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

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library(ggplot2)
```

On stocke ensuite les résultats du CAH dans la variable `data.hcpc` et on décide de ne pas afficher les graphiques tout de suite :

```
data.hcpc <- FactoMineR::HCPC(data.mca, nb.clust = 4, proba = 1, graph = FALSE)
```

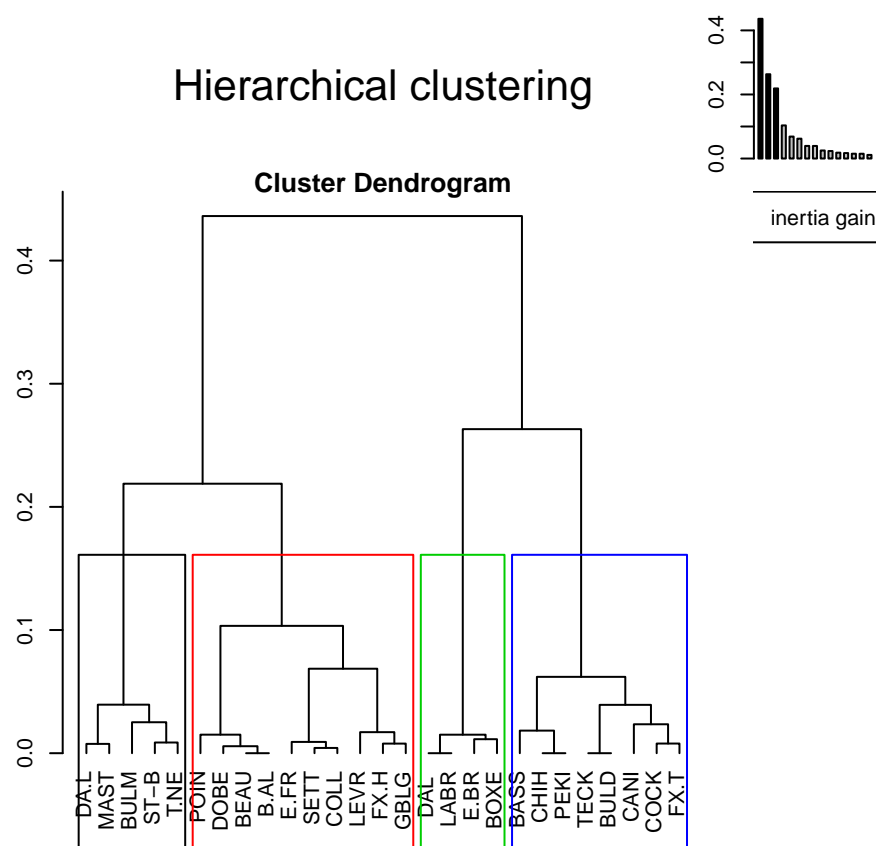
## Interprétation et analyse des résultats

### Typologie en 4 classes

Classification de départ, chaque individu = 1 groupe, tout ceux qui s'accroche avant sont des éléments qui se ressemblent, ont des proximités, mesure le degré de regroupement des branches (indice d'inertie), la somme est égale à l'inertie totale et nous permet de mesurer le niveau d'aggrégation des branches, plus l'indice est élevé plus les branches sont distinctes et inversement, le nombre de classe dépend du niveau de coupure de notre indice d'inertie

Pour visualiser la coupure de l'arbre en 4 classes, on utilise la fonction suivante :

```
plot(data.hcpc, choice = "tree")
```



On distingue 4 groupes principaux : \* Le premier groupe formé des races : DA.L, MAST, BULM, ST-B et T.NE \* Un second formé des races : POIN, DOBE, BEAU, B.AL, E.FR, SETT, COLL, LEVR, FX.H, GBL.G \* Un autre par les races : DAL, LABR, E.BR et BOXE \* Puis le dernier par : BASS, CHIH, PEKI, TECK, BULD, CANI, COCK, FX.T

Pour chaque classe déterminée au dessus, on décide de les caractériser en fonction des modalités actives, c'est à dire en fonction de leur taille, poids, vitesse, intelligence, affection et agressivité :

```
data.hcpc$desc.var$test.chi2
```

##	p.value	df
## Poids	7.377151e-10	6
## Taille	3.099411e-08	6
## Velocite	3.717407e-05	6
## Fonction	1.724164e-04	6
## Affection	1.746539e-03	3

```
## Agressivite 3.521994e-01 3
## Intelligence 6.346470e-01 6
```

p.value faible -> variable utile pour l'explication en 4 classes, dans notre cas Poids, Taille, Velocite, Fonction et Affection On retrouve ce test dans tout les analyses de construction d'un modèle

```
data.hcpc$desc.var$category
```

```
## $`1`
## Cla/Mod Mod/Cla Global p.value
## Poids=Poids_3 100.000000 100 18.51852 1.238697e-05
## Fonction=Fonction_3 62.500000 100 29.62963 6.936703e-04
## Affection=Affection_1 38.461538 100 48.14815 1.594203e-02
## Taille=Taille_3 33.333333 100 55.55556 3.719807e-02
## Velocite=Velocite_1 40.000000 80 37.03704 5.046451e-02
## Agressivite=Agressivite_2 30.769231 80 48.14815 1.558776e-01
## Intelligence=Intelligence_1 25.000000 40 29.62963 6.076551e-01
## Intelligence=Intelligence_2 16.666667 40 44.44444 8.521739e-01
## Intelligence=Intelligence_3 14.285714 20 25.92593 8.041992e-01
## Velocite=Velocite_3 11.111111 20 33.33333 5.534002e-01
## Taille=Taille_2 0.000000 0 18.51852 3.261984e-01
## Taille=Taille_1 0.000000 0 25.92593 1.920476e-01
## Agressivite=Agressivite_1 7.142857 20 51.85185 1.558776e-01
## Velocite=Velocite_2 0.000000 0 29.62963 1.440357e-01
## Poids=Poids_1 0.000000 0 29.62963 1.440357e-01
## Fonction=Fonction_2 0.000000 0 33.33333 1.061315e-01
## Fonction=Fonction_1 0.000000 0 37.03704 7.665056e-02
## Affection=Affection_2 0.000000 0 51.85185 1.594203e-02
## Poids=Poids_2 0.000000 0 51.85185 1.594203e-02
## v.test
## Poids=Poids_3 4.3706633
## Fonction=Fonction_3 3.3920688
## Affection=Affection_1 2.4102400
## Taille=Taille_3 2.0835835
## Velocite=Velocite_1 1.9560055
## Agressivite=Agressivite_2 1.4190734
## Intelligence=Intelligence_1 0.5134234
## Intelligence=Intelligence_2 -0.1863454
## Intelligence=Intelligence_3 -0.2479163
## Velocite=Velocite_3 -0.5926727
## Taille=Taille_2 -0.9817999
## Taille=Taille_1 -1.3045458
## Agressivite=Agressivite_1 -1.4190734
## Velocite=Velocite_2 -1.4609263
## Poids=Poids_1 -1.4609263
## Fonction=Fonction_2 -1.6158278
## Fonction=Fonction_1 -1.7704599
## Affection=Affection_2 -2.4102400
## Poids=Poids_2 -2.4102400
## $`2`
## Cla/Mod Mod/Cla Global p.value
## Poids=Poids_2 71.42857 100 51.85185 0.0001186541
## Velocite=Velocite_3 88.88889 80 33.33333 0.0001674908
## Taille=Taille_3 66.66667 100 55.55556 0.0003559624
```

```

## Fonction=Fonction_2      66.66667      60 33.33333 0.0377633046
## Affection=Affection_1    53.84615      70 48.14815 0.1042461226
## Intelligence=Intelligence_3 57.14286      40 25.92593 0.2463024898
## Agressivite=Agressivite_2 38.46154      50 48.14815 0.8909229596
## Fonction=Fonction_3      37.50000      30 29.62963 0.9667105841
## Intelligence=Intelligence_1 37.50000      30 29.62963 0.9667105841
## Agressivite=Agressivite_1 35.71429      50 51.85185 0.8909229596
## Velocite=Velocite_2      25.00000      20 29.62963 0.4479578393
## Intelligence=Intelligence_2 25.00000      30 44.44444 0.2834477498
## Affection=Affection_2     21.42857      30 51.85185 0.1042461226
## Poids=Poids_3            0.00000       0 18.51852 0.0766505636
## Taille=Taille_2          0.00000       0 18.51852 0.0766505636
## Fonction=Fonction_1      10.00000      10 37.03704 0.0334265616
## Taille=Taille_1          0.00000       0 25.92593 0.0219001610
## Poids=Poids_1            0.00000       0 29.62963 0.0109500805
## Velocite=Velocite_1      0.00000       0 37.03704 0.0023052801
##
##                                v.test
## Poids=Poids_2              3.84889031
## Velocite=Velocite_3        3.76359093
## Taille=Taille_3            3.57074864
## Fonction=Fonction_2        2.07741466
## Affection=Affection_1       1.62460795
## Intelligence=Intelligence_3  1.15937714
## Agressivite=Agressivite_2   0.13713643
## Fonction=Fonction_3         0.04173421
## Intelligence=Intelligence_1  0.04173421
## Agressivite=Agressivite_1  -0.13713643
## Velocite=Velocite_2        -0.75882401
## Intelligence=Intelligence_2 -1.07260640
## Affection=Affection_2      -1.62460795
## Poids=Poids_3              -1.77045987
## Taille=Taille_2            -1.77045987
## Fonction=Fonction_1        -2.12692204
## Taille=Taille_1            -2.29209501
## Poids=Poids_1              -2.54428785
## Velocite=Velocite_1        -3.04779385
##
## $`3`
##
## Cla/Mod Mod/Cla Global p.value
## Taille=Taille_2      80.000000      100 18.51852 0.0002849003
## Velocite=Velocite_2   50.000000      100 29.62963 0.0039886040
## Affection=Affection_2 28.571429      100 51.85185 0.0570370370
## Poids=Poids_2         28.571429      100 51.85185 0.0570370370
## Intelligence=Intelligence_2 25.000000      75 44.44444 0.2444444444
## Agressivite=Agressivite_1 21.428571      75 51.85185 0.3837037037
## Fonction=Fonction_2    22.222222      50 33.33333 0.5005128205
## Fonction=Fonction_1    20.000000      50 37.03704 0.6051282051
## Intelligence=Intelligence_3 14.285714      25 25.92593 0.9931623932
## Poids=Poids_3          0.000000       0 18.51852 0.4168091168
## Agressivite=Agressivite_2 7.692308      25 48.14815 0.3837037037
## Taille=Taille_1        0.000000       0 25.92593 0.2760683761
## Fonction=Fonction_3     0.000000       0 29.62963 0.2208547009
## Intelligence=Intelligence_1 0.000000       0 29.62963 0.2208547009
## Poids=Poids_1          0.000000       0 29.62963 0.2208547009

```

```

## Velocite=Velocite_3      0.000000      0 33.33333 0.1743589744
## Velocite=Velocite_1      0.000000      0 37.03704 0.1356125356
## Affection=Affection_1    0.000000      0 48.14815 0.0570370370
## Taille=Taille_3         0.000000      0 55.55556 0.0282051282
##
##                               v.test
## Taille=Taille_2         3.628656260
## Velocite=Velocite_2     2.879061615
## Affection=Affection_2   1.903026886
## Poids=Poids_2          1.903026886
## Intelligence=Intelligence_2 1.163949582
## Agressivite=Agressivite_1 0.871092400
## Fonction=Fonction_2     0.673683081
## Fonction=Fonction_1     0.517040045
## Intelligence=Intelligence_3 -0.008569774
## Poids=Poids_3          -0.811969528
## Agressivite=Agressivite_2 -0.871092400
## Taille=Taille_1        -1.089193928
## Fonction=Fonction_3     -1.224258569
## Intelligence=Intelligence_1 -1.224258569
## Poids=Poids_1          -1.224258569
## Velocite=Velocite_3     -1.358330065
## Velocite=Velocite_1     -1.492330459
## Affection=Affection_1   -1.903026886
## Taille=Taille_3        -2.194421409
##
## $`4`
##
## Cla/Mod Mod/Cla Global p.value
## Poids=Poids_1          100.000000    100.0 29.62963 4.504352e-07
## Taille=Taille_1        100.000000     87.5 25.92593 9.008705e-06
## Fonction=Fonction_1     70.000000     87.5 37.03704 9.594270e-04
## Velocite=Velocite_1     60.000000     75.0 37.03704 1.474275e-02
## Affection=Affection_2   50.000000     87.5 51.85185 2.280193e-02
## Agressivite=Agressivite_1 35.714286     62.5 51.85185 5.118196e-01
## Intelligence=Intelligence_1 37.500000     37.5 29.62963 5.910692e-01
## Intelligence=Intelligence_2 33.333333     50.0 44.44444 7.275362e-01
## Velocite=Velocite_2     25.000000     25.0 29.62963 7.734279e-01
## Taille=Taille_2        20.000000     12.5 18.51852 6.721665e-01
## Agressivite=Agressivite_2 23.076923     37.5 48.14815 5.118196e-01
## Intelligence=Intelligence_3 14.285714     12.5 25.92593 3.579068e-01
## Fonction=Fonction_2     11.111111     12.5 33.33333 1.684321e-01
## Poids=Poids_3           0.000000      0.0 18.51852 1.440357e-01
## Fonction=Fonction_3     0.000000      0.0 29.62963 3.404480e-02
## Affection=Affection_1    7.692308     12.5 48.14815 2.280193e-02
## Velocite=Velocite_3     0.000000      0.0 33.33333 1.971014e-02
## Poids=Poids_2           0.000000      0.0 51.85185 5.797101e-04
## Taille=Taille_3         0.000000      0.0 55.55556 2.229654e-04
##
##                               v.test
## Poids=Poids_1           5.0463054
## Taille=Taille_1         4.4396936
## Fonction=Fonction_1     3.3021618
## Velocite=Velocite_1     2.4386376
## Affection=Affection_2   2.2767371
## Agressivite=Agressivite_1 0.6560070
## Intelligence=Intelligence_1 0.5372872

```

```
## Intelligence=Intelligence_2  0.3484048
## Velocite=Velocite_2         -0.2878939
## Taille=Taille_2             -0.4231765
## Agressivite=Agressivite_2   -0.6560070
## Intelligence=Intelligence_3 -0.9193609
## Fonction=Fonction_2         -1.3772591
## Poids=Poids_3               -1.4609263
## Fonction=Fonction_3         -2.1195407
## Affection=Affection_1       -2.2767371
## Velocite=Velocite_3         -2.3318203
## Poids=Poids_2               -3.4409339
## Taille=Taille_3             -3.6914657
```

On analyse ce tableau de façon suivante :

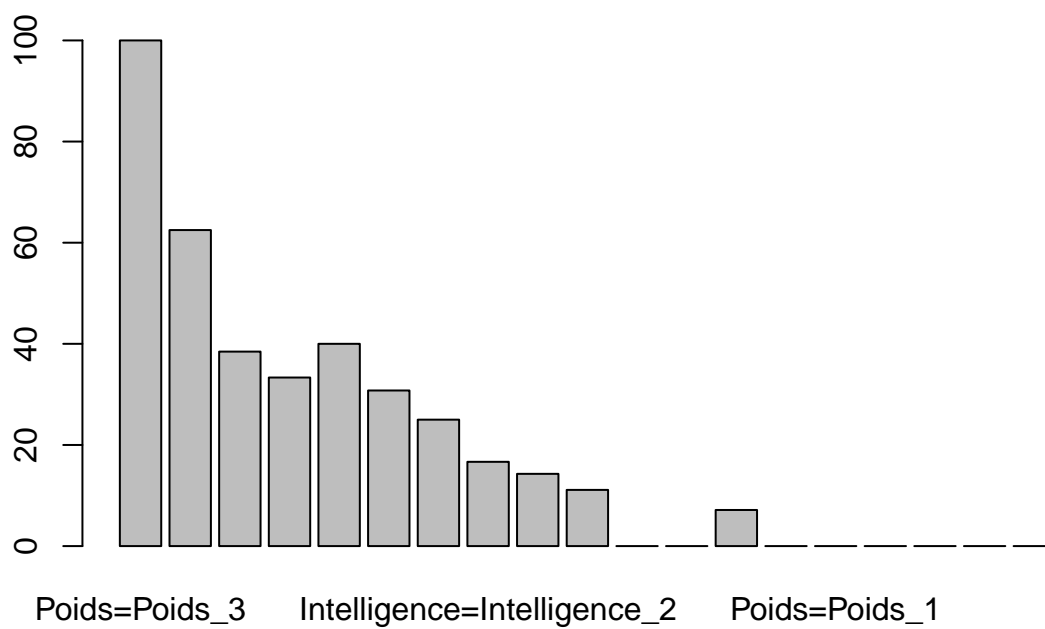
- Pour la première classe, on remarque que ce sont des races de chiens de grande taille et lourd (100% sont représentés), et qui dans les autres modalités sont plutôt moyen, ce sont des chiens qui ne sont pas rapide et ne sont pas affectueux
- La seconde quant à elle regroupe des chiens de grande taille, de poids moyen (proportion de la classe supérieure à la proportion totale), mais possédant une intelligence et une vitesse élevée, concernant l'affection, on ne peut pas retenir cette modalité car la proportion par rapport à la classe est quasiment égale à la proportion totale de cette modalité
- La troisième regroupe les races canines dont les modalités sont toutes moyennes dans l'ensemble (surtout pour la taille, la vitesse, l'affection et le poids)
- La dernière classe est caractérisée par les races de petites tailles et donc de poids faible (100% des individus de cette classe ont ces modalités), de vitesse faible et plutôt affectif, les autres modalités ne peuvent être retenus à cause de la p.value qui est trop élevée ( $> 5\%$ )

On représente ensuite l'histogramme des pourcentages des modalités dans la classe et dans l'échantillon afin de donner une première tendance de la typologie de chaque classe :

## Dimension 1

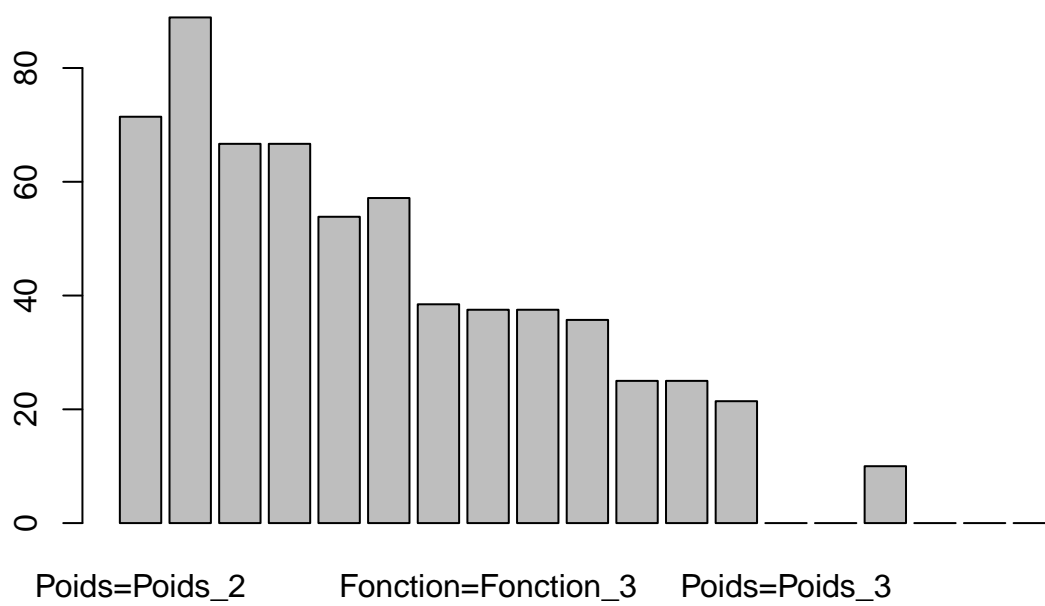
```
barplot(data.hcpc$desc.var$category$`1`[,1])
```





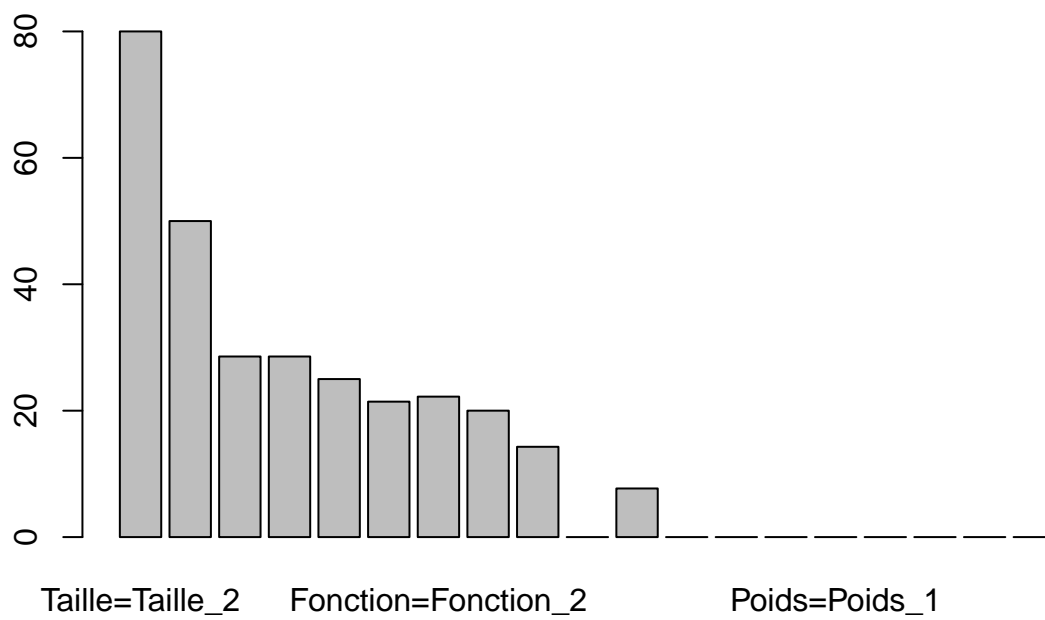
## Dimension 2

```
barplot(data.hcpc$desc.var$category$2[,1])
```



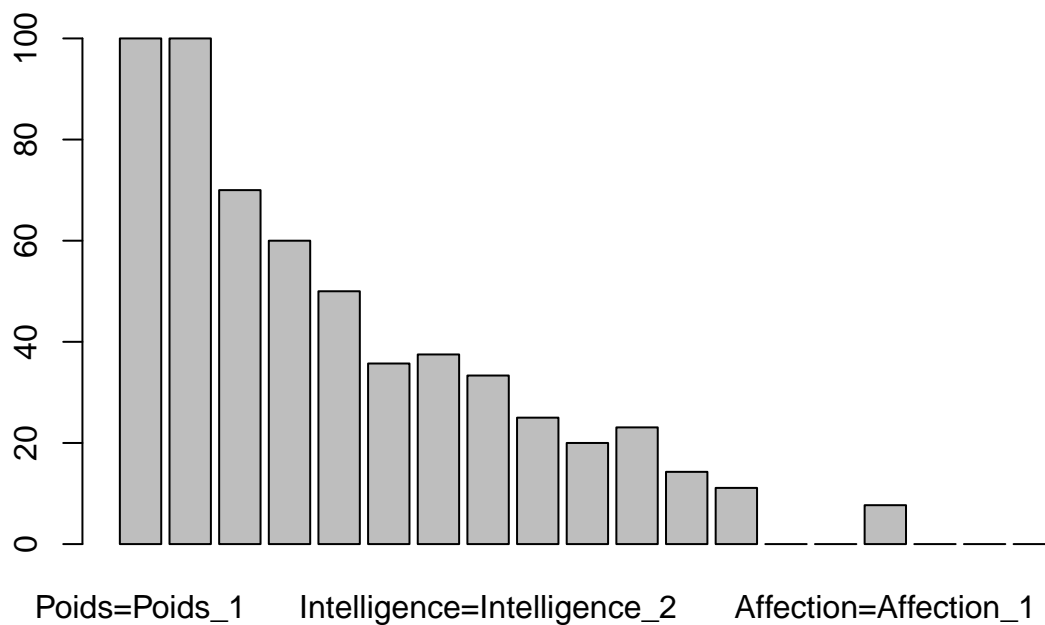
### Dimension 3

```
barplot(data.hcpc$desc.var$category$3[,1])
```



#### Dimension 4

```
barplot(data.hcpc$desc.var$category$`4`[,1])
```



La variance totale est l'inertie totale de notre tableau, on décompose cette variance en autant de proportion que de valeurs propres, soit 16 dans notre cas. le choix du nombre de classe fonctionne avec le critère du coude, il recherche la différence de descente la plus importante. Dans notre cas c'est 4 classes.

On analyse ensuite les variables les plus significatives par classe grâce à

On récupère ensuite les races canines les plus proches du centre de gravité grâce à :

```
data.hcpc$desc.ind
```

```
## $para
## Cluster: 1
##      ST-B      MAST      BULM      DA.L      T.NE
## 0.3993946 0.5275492 0.6939566 0.7508752 0.8352853
## -----
## Cluster: 2
##      POIN      DOBE      SETT      LEVR      COLL
## 0.5781682 0.6822483 0.7068663 0.7241162 0.7399578
## -----
## Cluster: 3
##      DAL      LABR      BOXE      E.BR
## 0.3183983 0.3183983 0.4579262 0.5475634
## -----
## Cluster: 4
##      CHIH      PEKI      FX.T      BULD      TECK
## 0.5298141 0.5298141 0.5506816 0.5786682 0.5786682
##
## $dist
```

```
## Cluster: 1
##      ST-B      T.NE      MAST      BULM      DA.L
## 1.596170 1.567528 1.559442 1.514469 1.145663
## -----
## Cluster: 2
##      BEAU      B.AL      POIN      LEVR      SETT
## 1.683641 1.683641 1.615030 1.567443 1.482398
## -----
## Cluster: 3
##      BOXE      DAL      LABR      E.BR
## 1.624937 1.607351 1.607351 1.381973
## -----
## Cluster: 4
##      BULD      TECK      CHIH      PEKI      BASS
## 1.827669 1.827669 1.809158 1.809158 1.538262
```

Pour chaque groupe, on obtient :

- Pour le groupe 1, ce sont les races canines BULM, ST-B, MAST

On représente graphiquement sur le plan factoriel 1 & 2, la projection :

- Des 4 classes
- Des 3 modalités de la variable FONCTION
- De l'ensemble des races canines

```
pl1 <- factoextra::fviz_cluster(data.hcpc, ellipse = FALSE)
factoextra::fviz_add(pl1, data.mca$quali.sup$coord)
```

