

Réalisation d'une ACM

Taniel Rémi - GIS2A4

30/09/2019

Contents

Partie 1 : Importation et visualisation des données	2
Partie 2 : Mise en oeuvre de l'ACM	3
Partie 3 : Analyse des résultats	3
Nombre d'axe à retenir	3
Analyse des 5 premiers axes en fonction des modalités	5
Analyse des axes par rapport aux variables	8
Analyse des 5 premiers axes en fonction des individus	10
Interprétation du premier plan factoriel	14
Partie 3 : Clasification ascendate hiérarchique sur les axes retenus	15
Réalisation sous R	15
Interprétation et analyse des résultats	16

Partie 1 : Importation et visualisation des données

Nos données sont sauvegardées au format CSV dans le fichiers `race`, nous commençons donc par importer les données dans la variable `data` grâce à la fonction suivante :

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

On visualise ensuite les types de nos données et la forme de celles-ci grâce à la fonction `str` appliquée à notre variable `data` :

```
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, représentant les 27 races de chiens et 8 variables, représentant les 7 modalités et un libellé. On remarque également que toutes les variables sont des variables qualitatives, et ont chacune 3 niveaux qui sont :

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

On se décide alors de visualiser les 6 premières lignes de nos données avec la fonction `head` :

```
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` avant de lancer les calculs et faire notre analyse, on décide donc de formater nos données afin de donner un identifiant aux différentes lignes de nos données, 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
```

Partie 2 : Mise en oeuvre de l'ACM

Pour réaliser l'ACM, nous allons avoir 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)
```

Les résultats seront alors stockés dans les variables suivantes :

- `$eig` pour les informations par rapport aux valeurs propres relatives à chaque dimension
- `$var` pour les résultats concernant les différentes modalités de nos variables
- `$ind` pour les informations par rapport aux différents individus, les races de chiens dans notre cas

Partie 3 : Analyse des résultats

Nombre d'axe à retenir

Pour rappel, on obtient les valeurs propres des dimensions grâce à la variable suivante :

```
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
```

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

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

Nous allons pour chacun de ces critères, déterminer le nombre de composantes principales à retenir.

Part d'inertie supérieure à la moyenne

Pour utiliser ce critère, on commence par calculer la moyenne des pourcentages des valeurs propres, obtenu grâce à :

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

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

Puis nous retenons les composantes principales dont le pourcentage d'inertie expliqué est supérieur à cette moyenne, dans notre cas, nous pouvons retenir les 5 premières composantes principales en utilisant ce critère.

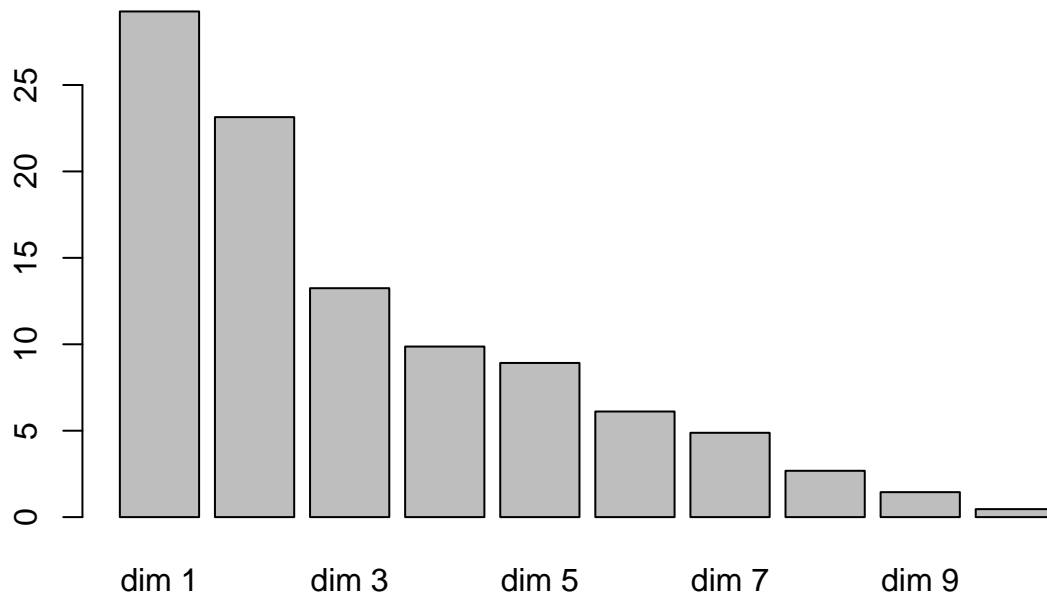
Part d'inertie cumulée supérieure à 80%

Pour ce critère, nous devons additionner les valeurs propres des dimensions jusqu'à obtenir un pourcentage cumulé supérieur à 80%, grâce au tableau précédent, nous pouvons retenir les 5 premières composantes qui à elles seules représentent un peu plus 84% de l'information totale.

Critère du coude

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

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



On recherche l'apparition d'un coude, sur notre graphique, le coude apparaît entre le 4e et 5e dimension, donc selon ce critère nous pouvons retenir 4 composantes principales.

Conclusion sur le nombre d'axe à retenir

Selon les critères, précédents, nous décidons de retenir 5 composantes principales pour notre analyse.

Analyse des 5 premiers axes en fonction des modalités

On recupère les données (contribution, qualité de représentation et coordonnées) des modalités sur les différentes dimensions grâce à la variable 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 5 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 sa projection.

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	+
Velocite_3	9.76	0.43	-
Somme	70.38		

Les modalités retenus expliquent 70.38% de l'information portée par la première dimension, de plus cette dimension permet d'opposer les modalités **Poids_1**, **Taille_1**, **Affection_2** aux modalités **Taille_3**, **Affection_1** et **Velocite_3**. Toutes ces modalités sont bien représentés dans cette dimension (qualité > 0.40) et les contributions de celles-ci sont plutôt bonnes.

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		

Les modalités retenus représentent 71.96% de l'information totale expliquée par cet axe, il permet d'opposer les modalités **Velocite_1**, **Taille_1** et **Intelligence_1** aux modalités telles que **Poids_2**, **Taille_2** et **Velocite_2**.

Par rapport à la dimension précédente, seules 2 modalités sont bien représentés sur cet axe (**Velocite_1** et **Poids_2**) dont les contributions sont les plus élevées, les autres modalités ne sont pas très bien représentés par rapport aux 2 autres modalités. Les contributions de ces modalités sont quant à elles dans la moyenne.

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		

Ces 5 modalités expliquent 67.44% de l'information portée par le 3e axe, cet axe permet d'opposer les modalités **Poids_3** et **Intelligence_3** aux modalités **Velocite_3**, **Intelligence_2** et **Taille_1**. A part les 2 modalités **Intelligence_3** et **Taille_1** qui ne sont pas très bien représentés (0.18 et 0.16 de cos2), les autres modalités sont bien représentés et leurs contributions sont bonnes.

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		

Seules 4 modalités permettent d'expliquer plus de 83% de l'information portée par la dimension 4, celle-ci oppose les modalités **Agressivite_2** et **Intelligence_3** aux modalités **Agressivite_1** et **Intelligence_2**. On remarque que 3 des 4 modalités ont une contribution supérieure à 20% sur cet axe, mais seulement 2 de ces modalités, **Agressivite_2** et **Agressivite_1**, sont extrêmement bien représentés (> 0.50), ces 2 modalités expliquent donc l'axe.

La modalité **Intelligence_3** malgré une forte contribution, n'est pas très bien représentée (< 0.30), on pourrait d'ailleurs enlever la modalité **Intelligence_2** du tableau ci-dessus car sa contribution et sa qualité de représentation sur cet axe sont faible par rapport aux autres modalités.

Dimension 5

Modalité	Contribution	Qualité	Signe
Intelligence_1	41.63	0.52	+
Intelligence_3	20.16	0.24	-
Velocite_2	9.40	0.12	+
Poids_3	6.65	0.07	-
Somme	77.84		

Les modalités retenus permettent d'expliquer 77.84% de l'information portée par la dimension 5, cette dimension permet d'opposer les modalités **Intelligence_1** et **Velocite_2** aux modalités **Intelligence_3** et **Poids_3**. La modalité **Intelligence_1** se démarque des autres modalités retenus de par sa contribution élevée ainsi que sa qualité de représentation excellente (> 0.5), la seconde modalité contribuant le plus à cette axe est également une modalité de la variable Intelligence mais sa contribution est 2 fois moins élevée, tout comme sa qualité de représentation.

Les autres modalités retenus sur cette dimension, **Velocite_2** et **Poids_3**, ont une faible contribution et ne sont pas du tout bien représentées, on pourrait décider de les enlever et de ne garder que les modalités de la variable Intelligence.

Analyse des axes par rapport aux variables

Pour chacun des axes que nous avons retenus, nous allons cumuler la somme des contributions des modalités pour chaque variables et voir pour chaque dimension quelles sont les variables qui y contribuent le plus.

Nous avons 6 variables différentes, on ne retiendra donc que les variables ayant une contribution supérieure à la moyenne qui est de :

[1] 16.66667

Dimension 1

Variable	Contribution
Taille	30.4
Poids	21.7
Velocite	14.2
Intelligence	5.9
Affection	20.65
Agressivite	6.1

A part les variables **Intelligence** et **Agressivite**, toutes les autres variables contribuent à cette dimension de manière à peu près égales (à part la variable **Poids** qui prédomine un peu)

Dimension 2

Variable	Contribution
Taille	20.7
Poids	30.9
Velocite	29.3
Intelligence	12.5
Affection	4
Agressivite	1.6

Dans le cas de la dimension 2, ce sont les variables **Poids** et **Velocite** qui y contribuent le plus (~30%) vient ensuite la variable **Taille** puis **Intelligence** avec respectivement 20.7% et 12.5%.

Dimension 3

Variable	Contribution
Taille	22
Poids	26.2
Velocite	21.1
Intelligence	25.5
Affection	2
Agressivite	2.5

Les variables **Affection** et **Agressivite** ne contribuent quasiment pas à la dimension 3, tandis que les autres variables contribuent quasiment de manière égale.

Dimension 4

Variable	Contribution
Taille	5.2
Poids	2.5
Velocite	2.3

Variable	Contribution
Intelligence	30.6
Affection	5.2
Agressivite	53.4

Cette fois-ci, seules les variables **Agressivite** et **Intelligence** contribuent à cette dimension, nous pouvons négliger le reste des variables étant donné que leurs contributions sont inférieures à la moyenne calculée précédemment.

Dimension 5

Variable	Contribution
Taille	3
Poids	8.7
Velocite	13.8
Intelligence	65.1
Affection	2.3
Agressivite	6

Tout comme l'axe précédent, nous avons seulement 2 variables qui contribuent d'une manière significative, dont une qui explique 65.1% de l'information portée par cet axe, la variable **Intelligence**.

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	0.64	+
TECK	8.17	0.64	+
DA.L	8	0.53	-
DOBE	6.13	0.55	-
FX.T	6.06	0.44	+
CANI	5.93	0.38	+
CHIH	5.45	0.38	+
PEKI	5.45	0.38	+
FX.H	5.83	0.55	-
COCK	4.30	0.28	+
BULM	4.20	0.27	-
Somme	67.69		

Les races retenus permettent d'expliquer plus de 67% de l'information portée par la première dimension, sur cette dimension, les races BULD, TECK, FX.T, CANI, CHIH, PEKI et COCK aux races DA.L, DOBE, FX.H et BULM. Sur cette dimension, les cos2 sont plutôt bon et les contributions sensiblement égales.

Dimension 2

Individu	Contribution	Qualité	Signe
BASS	11.38	0.62	+
E.BR	10.83	0.54	-
LABR	9.36	0.54	-
DAL	9.36	0.54	-
MAST	8.04	0.43	+
BOXE	7.37	0.42	-
PEKI	6.35	0.35	+
CHIH	6.35	0.35	+
Somme	69.04		

Ces 8 races canines expliquent un peu moins de 70% de l'information totale portée par le second axe, cet axe oppose les races BASS, MAST, PEKI et CHIH aux races E.BR, LABR, DAL, BOXE. A part les races PEKI et CHIH qui sont un peu moins bien représentés par rapport aux autres individus, le reste des qualités de représentations sont plutôt bonnes.

Dimension 3

Individu	Contribution	Qualité	Signe
ST-B	13.75	0.47	+
T.NE	10.03	0.35	+
BOXE	7.23	0.23	+
CANI	6.64	0.19	-
POIN	6.01	0.25	-
BEAU	5.72	0.23	-
B.AL	5.72	0.23	-
MAST	5.54	0.17	+
COCK	5.18	0.15	+
LABR	4.62	0.15	+
Somme	70.44		

Les races retenus au-dessus permettent d'expliquer 70.44% de l'information portée par la dimension 3, cette dimension met en avant l'opposition entre les races ST-B, T.NE, BOXE, MAST, COCK, LABR aux races canines CANI, POIN, BEAU et B.AL. On remarque que par rapport aux races ST-B et T.NE, les autres races canines ne contribuent pas significativement et ne sont pas très bien représentés sur cet axe.

Dimension 4

Individu	Contribution	Qualité	Signe
SETT	13.8	0.52	-
LEVR	9.3	0.30	-
E.FR	8.6	0.31	-
BULM	7.9	0.17	+
COLL	7.9	0.31	-
T.NE	7	0.18	-
BEAU	6.9	0.21	+
B.AL	6.9	0.21	+
COCK	5.2	0.11	+
Somme	73.5		

Les 9 races retenus expliquent 73.5% de l'information portée par la dimension 4, sur cette dimension les races BULM, BEAU, B.AL et COCK sont négativement corrélés aux races SETT, LEVR, E.FR, COLL et T.NE.

On remarque que les races négativement représentés dans cette dimension, ont des contributions élevés et une bonne qualité de représentation (sauf pour la race T.NE mais cela est du à sa forte contribution à l'axe précédent), tandis que les races positivement représentés ont une contribution faible et une mauvaise qualité de représentation (ces races apparaissent avec une meilleur représentation dans d'autres axes).

Dimension 5

Individu	Contribution	Qualité	Signe
GBLG	20.6	0.57	+

Individu	Contribution	Qualité	Signe
FX.H	10	0.29	+
BULM	10	0.20	-
T.NE	8.6	0.20	-
BASS	7.2	0.15	+
Somme	56.4		

Ces 5 races expliquent 56.4% de l'information totale portée par le dernier axe que nous retenons, il met en opposition les races GBLG, FX.H et BASS aux races canines BULM, T.NE.

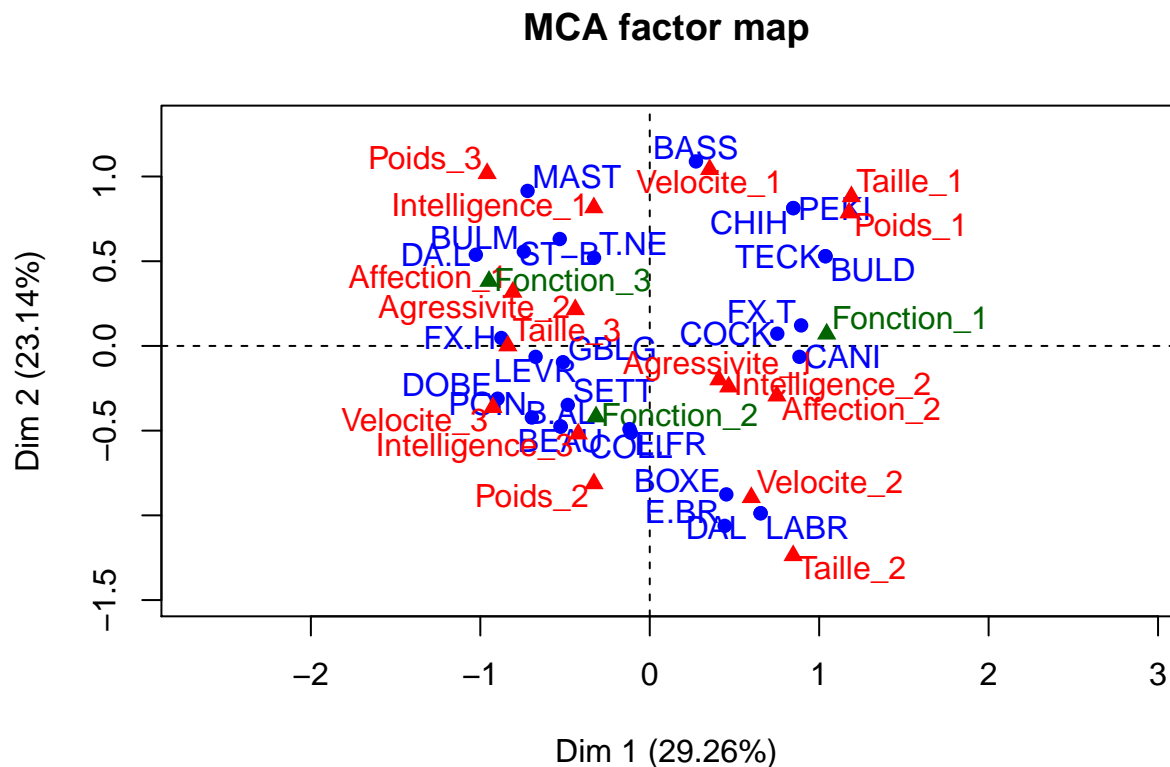
On décide de ne retenir que 5 races malgré le fait qu'elles ne représentent que 56% de l'information totale de notre dimension car les autres races contribuent très faiblement et leurs qualités de représentations n'est pas assez bonne pour être retenu.

On remarque qu'une race se démarque des autres dans cette dimension, c'est la race GBLG qui contribue à plus de 20% et qui est très bien représentée, c'est la première fois que cette race apparaît dans l'analyse des axes factoriels que nous avons retenu, dans le cas des autres races retenus, elles ne sont pas forcément très bien représentées car elles apparaissent dans d'autres axes. On pourrait donc choisir de ne retenir que 4 axes et de mettre la race GBLG en individu supplémentaire lors de l'ACM.

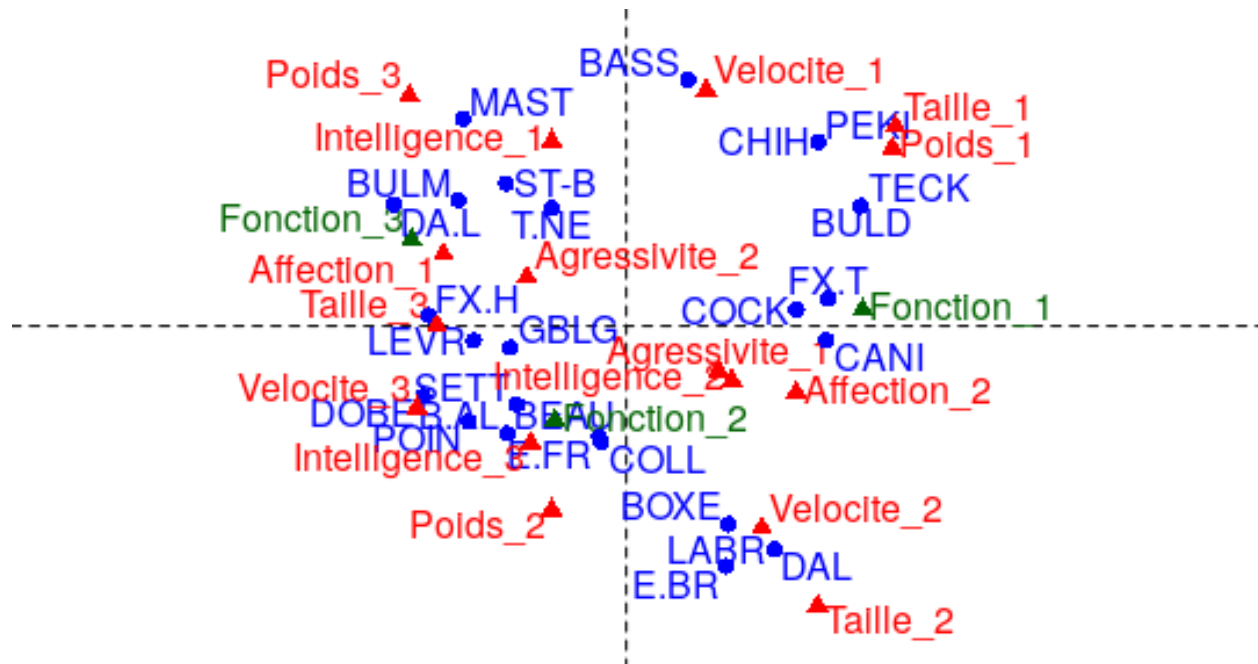
Interprétation du premier plan factoriel

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))
```



D'après ce plan, on pourrait regrouper les races canines en 4 groupes différents :



Voici le détail de chaque groupe :

- Un premier regroupant les chiens de petites tailles et de poids faibles et pas très rapide avec des races comme BASS, PEKI, CHIH, BULD, FX.T, COCK et CANI. Plutôt des chiens de compagnies
- Un second regroupant les races BOXE, DAL, E.BR, LABR qui sont des races canines de taille et de rapidité moyenne
- Un troisième regroupant les races canines de grande taille et de poids important mais peu affectif, tel que MAST, DA.L ou ST-B
- Puis un dernier groupe avec des races comme LEVR, DOBE ou BEAU, qui sont des races rapides, intelligentes et de poids moyen

Partie 3 : Clasifcation ascendate hiérarchique sur les axes retenus

Pour rappel, nous avons retenus les 5 premiers axes factoriels qui représentent 84% 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(ggplot2)
library(plyr)
library(philentropy)
library(factoextra)
```

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

On stocke ensuite les résultats de la CAH dans la variable `data.hcpc` et on décide de ne pas afficher les graphiques lors de l'appel à la fonction :

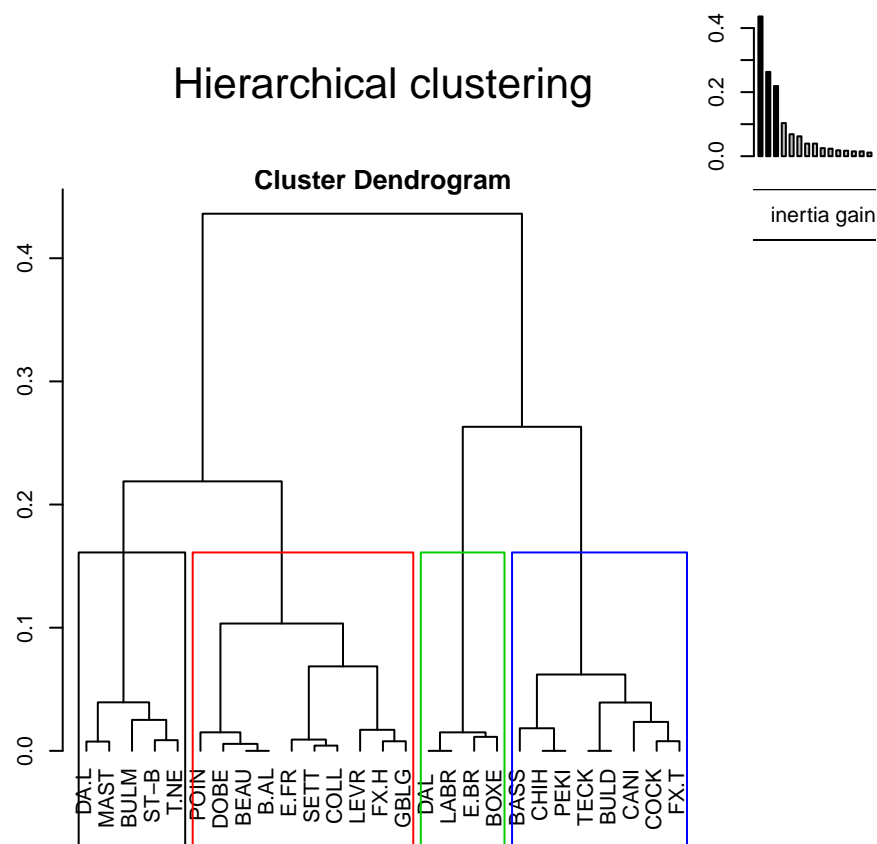
```
data.hcpc <- FactoMineR::HCPC(data.mca, nb.clust = 4, proba = 1, graph = FALSE)
```

Interprétation et analyse des résultats

Typologie en 4 classes

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

Analyse des variables les plus explicatives pour les 4 classes

On se décide d'abord de déterminer quelles seront les variables qui seront les plus explicatives pour l'analyse des 4 classes, pour cela on utilise la p.value de nos variables :

```
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
```

On remarque que les races canines seront divisés en classe principalement par rapport à la variable Poids, Taille et Vitesse, les autres variables.

Caractérisation des 4 classes en fonction des différentes modalités

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$desc.var$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
## Vitesse=Vitesse_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
## Vitesse=Vitesse_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
## Vitesse=Vitesse_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
## Vitesse=Vitesse_1 1.9560055
## Agressivite=Agressivite_2 1.4190734
## Intelligence=Intelligence_1 0.5134234
## Intelligence=Intelligence_2 -0.1863454
## Intelligence=Intelligence_3 -0.2479163
## Vitesse=Vitesse_3 -0.5926727
## Taille=Taille_2 -0.9817999
## Taille=Taille_1 -1.3045458
## Agressivite=Agressivite_1 -1.4190734
## Vitesse=Vitesse_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              80.00000      100 18.51852 0.0002849003
## Velocite=Velocite_2          50.00000      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  25.00000       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
```

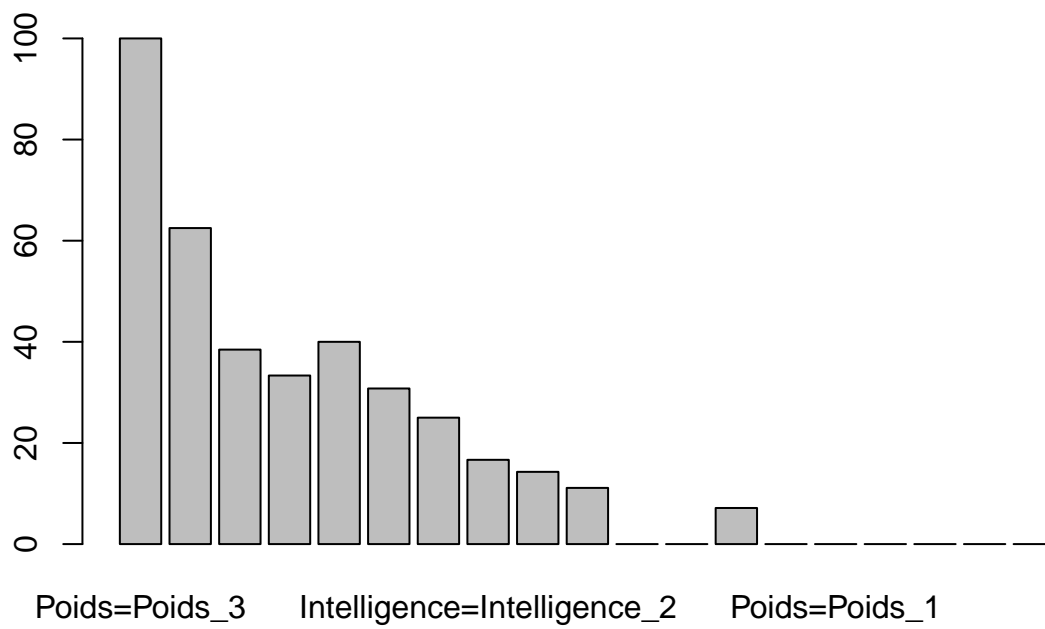
On analyse ce tableau de façon suivante :

- 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 :

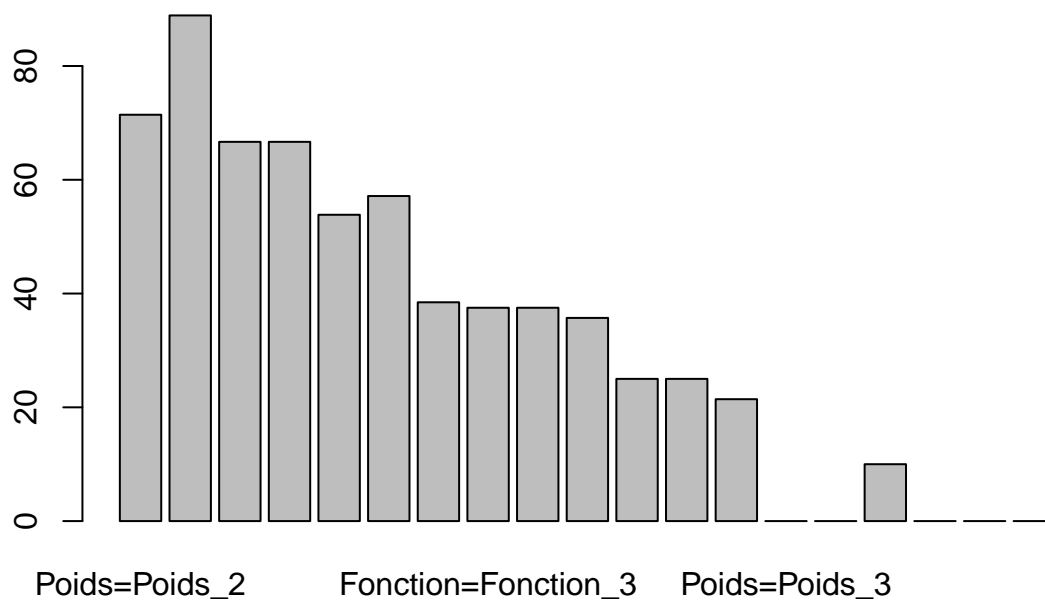
Dimension 1

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



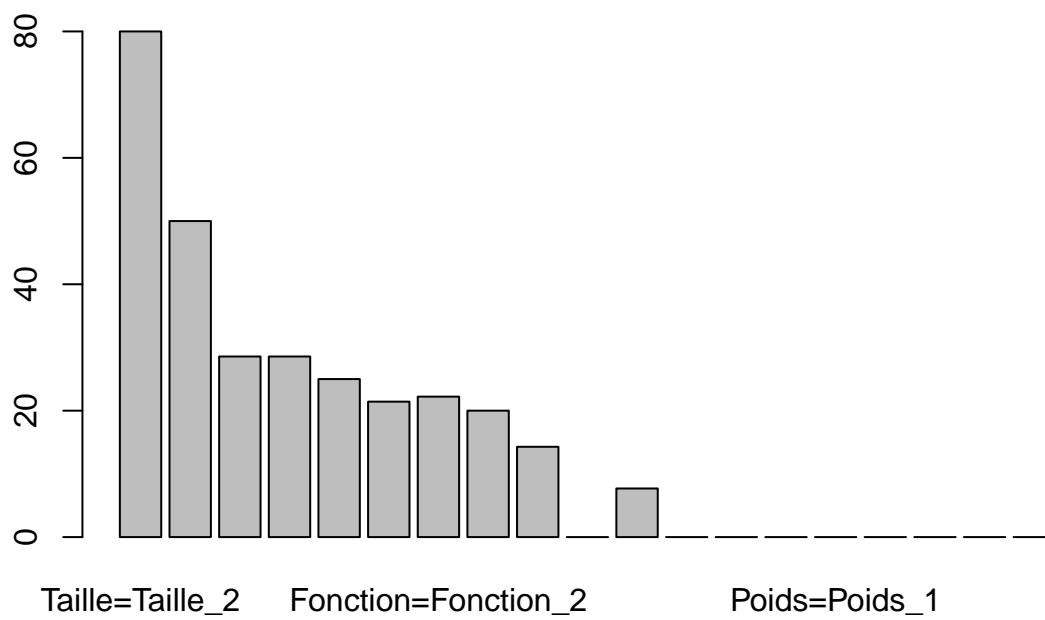
Dimension 2

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



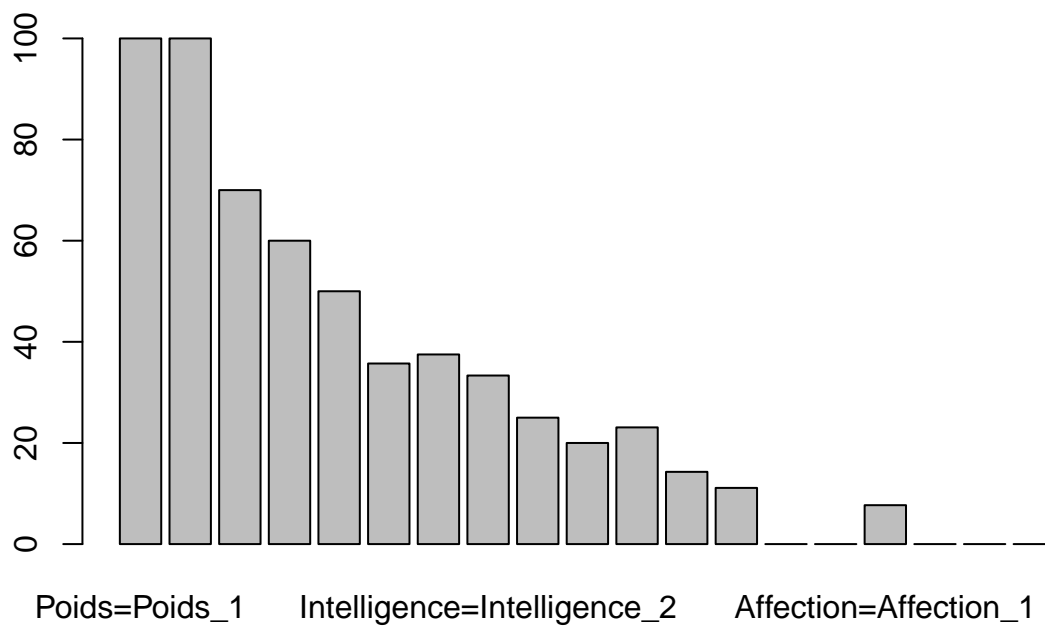
Dimension 3

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



Dimension 4

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

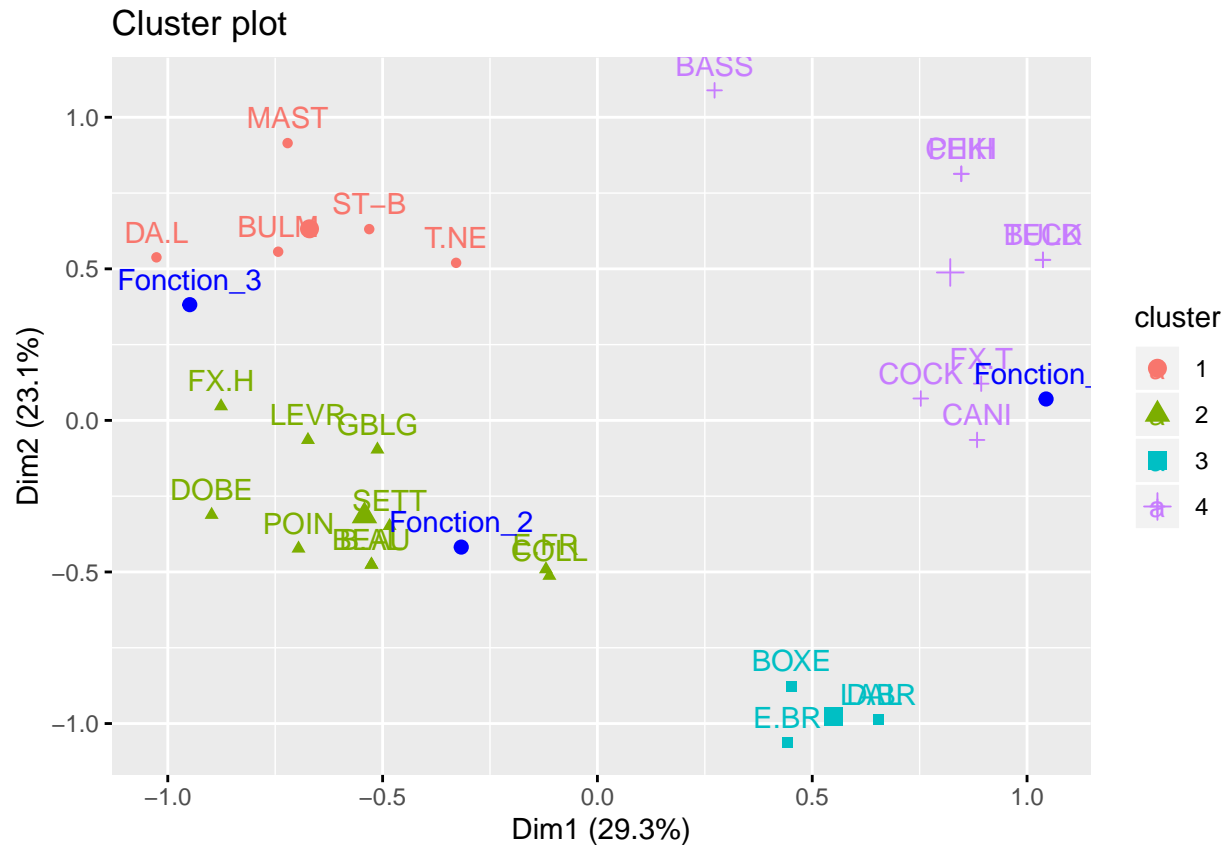


On représente graphiquement sur le plan factoriel 1 & 2, la projection :

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

Pour cela, on utilise le code suivant :

```
p11 <- factoextra::fviz_cluster(data.hcpc, ellipse = FALSE)
factoextra::fviz_add(p11, data.mca$quali.sup$coord)
```

On remarque qu'à part le groupe 3, tout les autres groupes sont caractérisés par une modalité de la variable Fonction :

- X pour le groupe 1, qui regroupe des chiens de grande taille et lourd
- X pour le groupe 2, ce groupe regroupe les races canines de grande taille et de poids moyens, mais qui possède une vitesse élevée
- Enfin X pour le groupe 4, ce groupe est formé de chiens de petite taille et de poids faible, qui ne sont pas très rapide mais plutôt affectueux.

A l'aide du graphique précédent, on décide de noter les races canines les plus proches du centre de gravité de chaque classe, on obtient les résultats suivants :

- Pour la classe 1, ce sont les races canines BULM, ST-B, MAST
- Pour la classe 3, toutes les races canines sont proches du centre de gravité de la classe