

# R Notebook

Dans ce notebook vous retrouverez l'ensemble de notre travail répondant aux différentes questions sur les bases de données "chiens" et "voitures". Nous y avons incorporé des extraits de codes ainsi que nos réponses.

## Sommaire :

<b>1 - Analyse en composantes principales</b>	<b>2</b>
A) Data cleaning et Data visualisation	2
B) ACP	5
1. Inertie	5
2. Interprétation des axes	6
3. Représentation des individus	7
<b>2 - Classification</b>	<b>10</b>
A) K-means	10
1. K-means à 3 clusters	11
2. K-means à 2 clusters	13
3. K-means à 4 clusters	14
B) Classification Ascendante Hiérarchique (Méthode de Ward)	16
<b>3 - Races de chiens</b>	<b>18</b>
A) Analyse des correspondances multiples à 2 dimensions	21
1. Valeurs propres	22
2. $\text{Cos}^2$	22
3. Contribution des variables à la constitution des dimensions	23
B) Analyse des correspondances multiples à 3 dimensions	24
C) Description des différentes races de chiens	25
1. Interprétation des races	25
2. Interprétation des quadrants pour les dimensions 1 et 2	27

# 1 - Analyse en composantes principales

## A) Data cleaning et Data visualisation

```
library("ggpubr")

library("FactoMineR")
library("heatmaply")

library("factoextra")
```

### Visualisation et Statistiques de base :

```
head(cars)

##           CYL  PUIS  LON  LAR  POIDS  VITESSE  ACCEL  CO2
## ALPHAMITO   875   105 406  172   1130     184   11.4   98
## AUDIA1      999    95 397  174   1065     186   10.9  103
## CITROENC4  1199   130 442  182   1280     196   10.1  115
## JAGUARF     2995   340 447  192   1587     260    5.7  234
## PEUGEOTRCZ  1997   160 428  184   1370     220    8.2  130
## LANDROVER   2993   256 483  191   2570     180    9.3  203

dim(cars)

## [1] 20  8

str(cars)

## 'data.frame':   20 obs. of  8 variables:
## $ CYL      : int  875 999 1199 2995 1997 2993 898 1995 898 1995 ...
## $ PUIS      : int  105 95 130 340 160 256 90 116 90 136 ...
## $ LON       : int  406 397 442 447 428 483 406 462 406 447 ...
## $ LAR       : int  172 174 182 192 184 191 173 181 173 185 ...
## $ POIDS     : int  1130 1065 1280 1587 1370 2570 1092 1570 962 1751 ...
## $ VITESSE   : int  184 186 196 260 220 180 182 198 175 184 ...
## $ ACCEL     : num  11.4 10.9 10.1 5.7 8.2 9.3 12.2 11.1 11.1 10.9 ...
## $ CO2       : int   98 103 115 234 130 203 105 109 116 139 ...

summary(cars)

##           CYL           PUIS           LON           LAR           POIDS
## Min.      : 875   Min.      : 69.0   Min.      :272.0   Min.      :163.0   Min.      : 905
## 1st Qu.:1149   1st Qu.:102.5   1st Qu.:406.0   1st Qu.:174.8   1st Qu.:1168
## Median :1995   Median :133.0   Median :443.5   Median :181.5   Median :1465
## Mean      :2116   Mean      :181.6   Mean      :432.4   Mean      :182.2   Mean      :1497
## 3rd Qu.:2343   3rd Qu.:177.8   3rd Qu.:464.5   3rd Qu.:191.2   3rd Qu.:1728
```

```
## Max. :6262 Max. :660.0 Max. :522.0 Max. :200.0 Max. :2570
## VITESSE ACCEL CO2
## Min. :160.0 Min. : 4.10 Min. : 98.0
## 1st Qu.:181.5 1st Qu.: 9.15 1st Qu.:112.0
## Median :192.5 Median :10.80 Median :134.0
## Mean :203.2 Mean : 9.95 Mean :156.1
## 3rd Qu.:209.0 3rd Qu.:11.43 3rd Qu.:156.5
## Max. :335.0 Max. :13.70 Max. :380.0
```

## Valeurs manquantes :

On vérifie le nombre de valeurs manquantes (que l'on voyait à l'œil nu au vu de la faible dimension des données). Ici il n'y en a pas.

```
sum(is.na(cars))
```

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

## Outliers :

```
boxplot(cars)
```

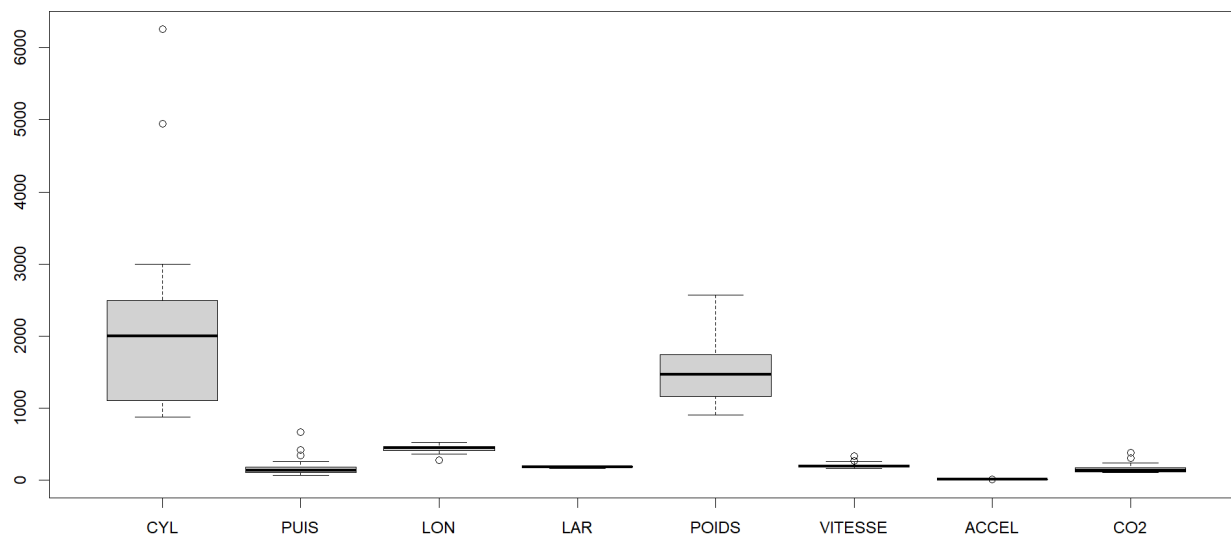
```
boxplot(cars$CYL)
```

```
boxplot(cars$LON)
```

```
boxplot(cars$VITESSE)
```

```
boxplot(cars$POIDS)
```

```
boxplot(cars$CO2)
```



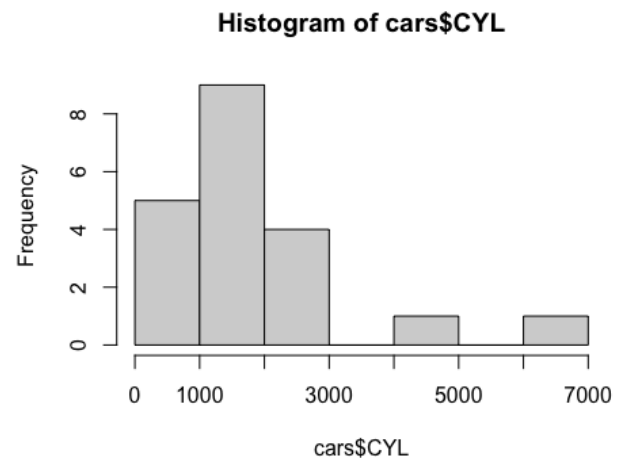
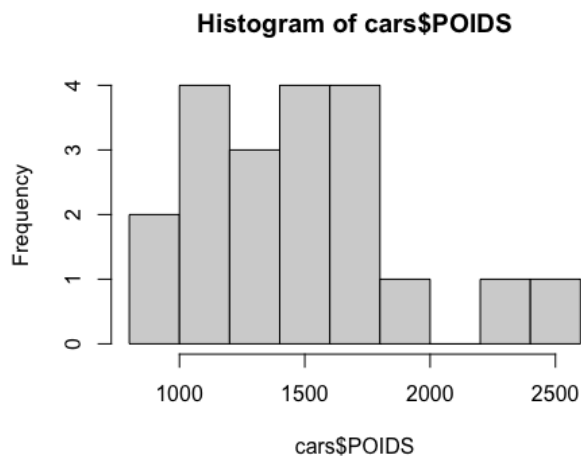
Le boxplot nous montre que concernant les Cylindres et la puissance, les valeurs aberrantes sont vers le haut. Ce qui nous laisse penser qu'il s'agit potentiellement des voitures sportives. Longueur et largeur sont assez uniformes pour toutes les voitures de notre base de données, la seule valeur aberrante étant le Ford Mustang de 272cm. Concernant la vitesse, les valeurs aberrantes correspondent sûrement à des voitures sportives. La même analyse peut être faite pour Accélération mais à l'envers car moindre est le temps d'accélération, meilleure est la performance. Pour finir, concernant le poids, nous n'avons pas de valeur aberrante mais la tranche est assez large avec une valeur minimum de moins de 1000 et maximum de 2500.

## Histogrammes :

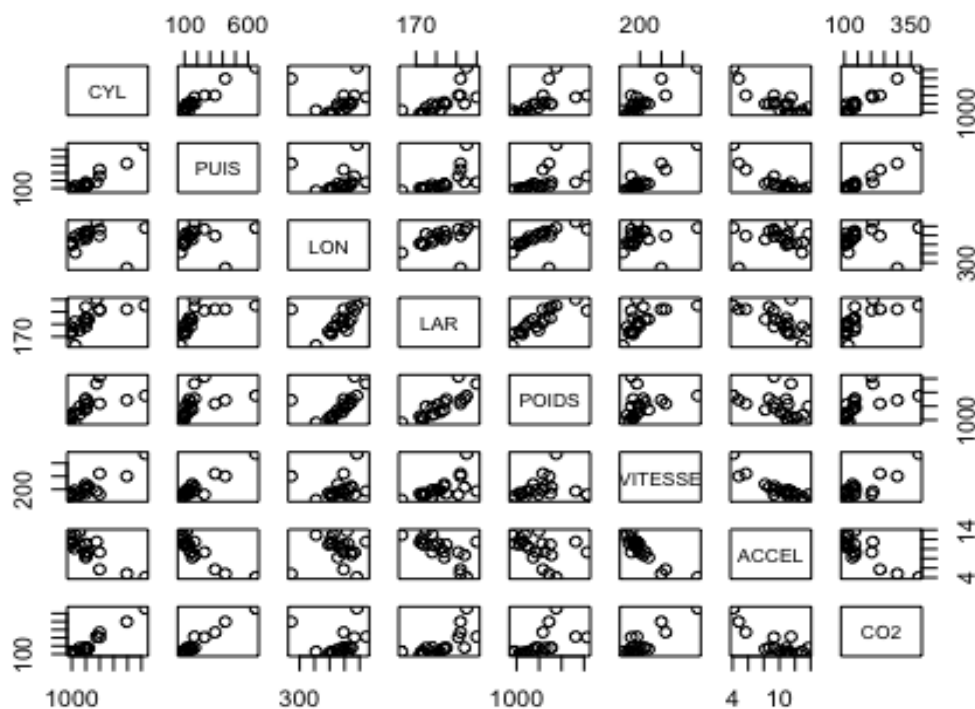
Les histogrammes complètent notre analyse et montrent que la fréquence est faible mais la valeur est importante.

```
hist(cars$POIDS)
```

```
hist(cars$CYL)
```



```
pairs(cars)
```



Le pairplot et la matrice de corrélation sont assez complémentaires et nous montrent certaines relations intéressantes. Les corrélations positives les plus importantes sont entre le cylindrage et la puissance, le cylindrage et le CO<sup>2</sup> ainsi qu'entre la puissance et la vitesse. Les corrélations négatives les plus importantes sont entre le temps d'accélération et le cylindrage et entre le temps d'accélération et la vitesse. Ceci semble logique car plus le temps d'accélération est élevé plus la voiture aura tendance à être lente. Les corrélations les plus faibles sont entre la longueur et le temps d'accélération et entre la longueur et le CO<sup>2</sup>.

## B) ACP

Import des packages nécessaires

- FactoMineR afin de réaliser l'ACP
- factoextra afin de visualiser les résultats issus d l'ACP

```
library('FactoMineR')  
library('factoextra')
```

Nous pouvons réaliser l'ACP à l'aide de la fonction 'PCA' :

```
cars.pca <- PCA(cars, graph=F)
```

Affichons les valeurs propres et % d'inerties expliquée :

```
cars.pca$eig
```

	eigenvalue	% of variance	cumulative % of variance
comp 1	5.549271024	69.36588781	69.36589
comp 2	1.550202725	19.37753406	88.74342
comp 3	0.480655817	6.00819771	94.75162
comp 4	0.280682369	3.50852962	98.26015
comp 5	0.084751837	1.05939796	99.31955
comp 6	0.034770185	0.43462731	99.75417
comp 7	0.013450458	0.16813073	99.92231
comp 8	0.006215585	0.07769482	100.00000

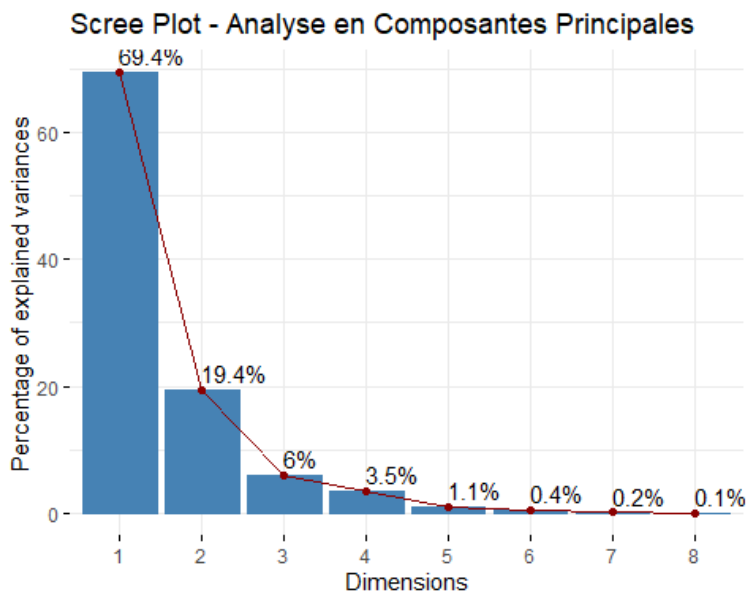
## 1. Inertie

Sans faire d'ACP, si on travaille avec des données centrées normées, le taux d'inertie expliquée par deux variables serait  $(2/p)*100\%$ , dans ce cas 25%.

Les résultats de l'ACP montrent que le premier plan factoriel explique 88,74% de l'inertie totale. On en déduit qu'une analyse en composantes principales s'avère très utile.

Visualisation de l'importance des axes via un Scree Plot

```
fviz_eig(cars.pca, addlabels = TRUE, linecolor="darkred", main = "Scree Plot - Analyse en Composantes Principales")
```

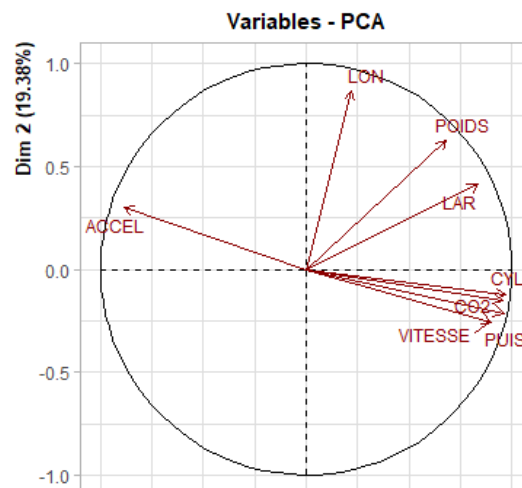


Le graphique permet de mieux visualiser l'importance de chacun des axes en termes de % d'inertie expliquée. On voit clairement que le premier plan factoriel explique au mieux les différences entre les voitures. Les autres dimensions pourraient être cependant utiles pour caractériser des voitures atypiques qui ne seraient pas bien projetées sur ce premier plan.

## 2. Interprétation des axes

Afin d'interpréter les données, on va tout d'abord analyser le cercle des corrélations. Il nous permet de comprendre comment les axes sont corrélés aux variables d'origine.

```
plot(cars.pca, choix = "var", cex = 0.8, col.var = "red4", title = "Variables - PCA")
```



Cette représentation graphique indique une forte corrélation positive du premier axe avec les variables «CYL», «PUIS», «CO2». De plus, on observe une corrélation positive non négligeable du premier axe avec la variable «VITESSE» et une corrélation négative assez importante avec la variable «ACCEL». La variable «ACCEL» est négativement corrélée au premier axe en raison des unités de mesure. D'habitude on mesure l'accélération par le temps nécessaire pour qu'une voiture atteigne une vitesse donnée (100km/h). Une faible valeur de l'accélération indique que la voiture peut développer une vitesse élevée rapidement. Les autres variables, notamment «LAR», «POIDS» et «LON» sont également positivement corrélés au premier axe, mais dans une moindre mesure. On en déduit la présence d'un effet taille sur le premier axe.

- En résultat, dans le quadrant I et IV, on s'attend à voir les voitures puissantes, dont la cylindrée du moteur est particulièrement élevée; elles sont capables de développer une vitesse élevée rapidement et sont très polluantes.
- En revanche, le quadrant II et III contiendra des voitures peu puissantes et moins polluantes, ayant des moteurs moins performants.

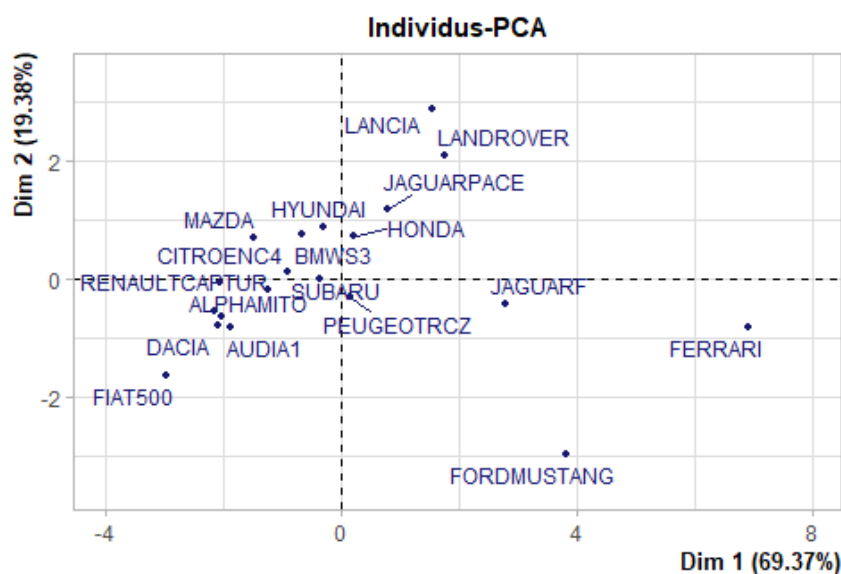
Quant au deuxième axe, le cercle indique une forte corrélation positive avec la variable «LON». De plus, les variables «POIDS» et «LARG» y sont aussi corrélés positivement mais dans une moindre mesure. On voit un effet forme apparaître.

- Le quadrant I et II rassemblera les voitures les plus grandes, longues et massives.
- Le quadrant III et IV contiendra les voitures les plus petites et compactes.

### 3. Représentation des individus

Maintenant, on passe à la représentation des individus (des voitures) sur le premier plan factoriel. Cependant, on ne peut interpréter ni la proximité des voitures sur le graphique, ni leur position sans avoir d'abord vérifié la qualité des projections.

```
plot(cars.pca, choix="ind", cex=0.8, col.ind="midnightblue",
title="Individus-PCA")
```

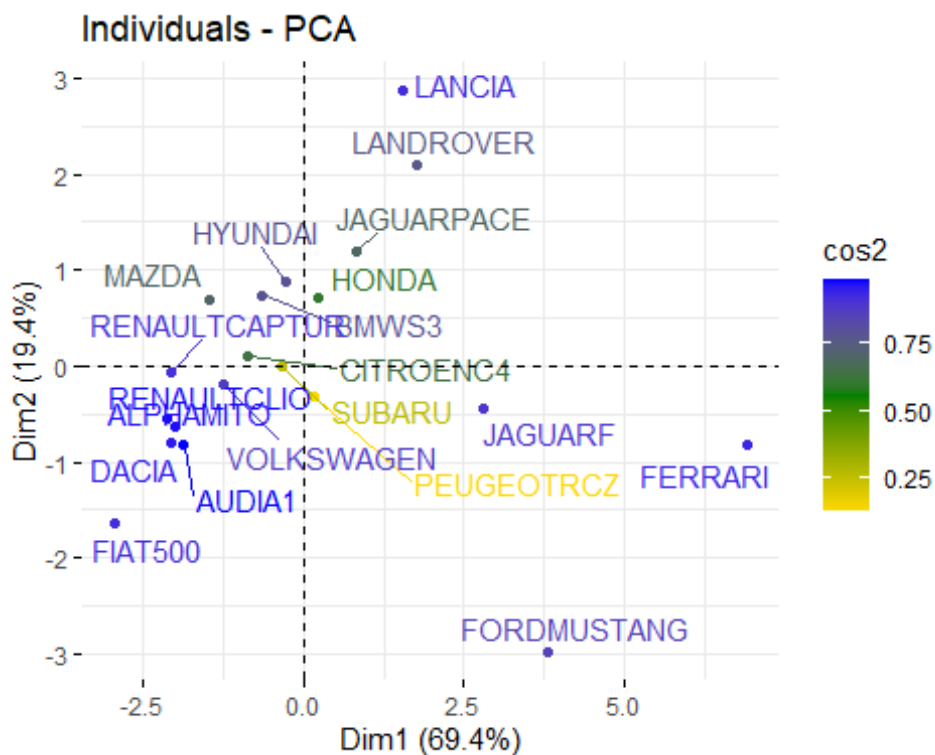


```
cars.pca$ind$cos2[,1:2]
```

	Dim.1	Dim.2
ALPHAMITO	0.88418157	0.0882481546
AUDIA1	0.82447953	0.1603240825
CITROENC4	0.63441116	0.0097161853
JAGUARF	0.87295591	0.0211260379
PEUGEOTRCZ	0.02354612	0.1090728501
LANDROVER	0.31747599	0.4453076078
RENAULTCLIO	0.91588474	0.0619189342
BMWS3	0.34739866	0.4385043460
DACIA	0.83548059	0.1227757130
HYUNDAI	0.07691485	0.7124226564
LANCIA	0.20567125	0.7152890894
RENAULTCAPTUR	0.91218646	0.0011120940
FORDMUSTANG	0.52868209	0.3233183930
FIAT500	0.70301792	0.2181054338
HONDA	0.05681670	0.5506712131
FERRARI	0.92712779	0.0130358167
SUBARU	0.26365614	0.0002406284
MAZDA	0.57219677	0.1274534969
VOLKSWAGEN	0.85640405	0.0198445442
JAGUARPACE	0.21914850	0.4760572094

Pour faciliter l'interprétation, on a ajouté des couleurs indicatrices de la qualité de la projection. Les voitures mal projetées se retrouvent, sans surprise, au centre.

```
fviz_pca_ind(cars.pca,
  col.ind = "cos2",
  gradient.cols = c("#FFD700", "#008000", "#0000FF"),
  repel = TRUE)
```





**a)** Les individus sont-ils bien représentés sur le premier plan factoriel?

L'étude des  $\cos^2$  indique que la plupart des voitures sont bien projetées sur le premier plan factoriel, ayant la somme des  $\cos^2$  pour les deux premiers axes supérieure à 50%. Les seules exceptions sont PEUGEOTRCZ et SUBARU. De plus, la somme des  $\cos^2$  des voitures HONDA, CITROENC4 et MAZDA est comprise entre 60% et 70%.

**b)** Quelles sont les caractéristiques des individus en haut du graphe?

En haut du graphe, on a les voitures les plus massives, dont la longueur et le poids sont très élevés. Parmi les grandes voitures, on distingue en haut à droite les voitures plus puissantes et en haut à gauche celles moins puissantes.

**c)** Quelles sont les caractéristiques des individus à droite du graphe?

A droite du graphe, on observe les voitures les plus puissantes, avec la cylindrée du moteur élevée. Par ailleurs, elles sont capables de développer une vitesse très importante rapidement et sont, en conséquence, les plus polluantes. Parmi les voitures puissantes, on distingue en haut à droite les grandes voitures et en bas à droite les voitures plus compactes.

**d)** Quelles sont les caractéristiques des individus en bas à gauche du graphe?

Le profil de ces voitures indiquerait qu'elles appartiennent à la classe des voitures assez petites et peu puissantes. Le graphique le confirme, on retrouve des voitures telles que FIAT500, DACIA et AUDIA1.

**e)** Peut-on dire que les individus PEUGEOTRCZ et JAGUARF ont un profil semblable? Si oui quel est-il ?

On ne peut rien dire sur la proximité de ces deux voitures sur le graphique car PEUGEOTRCZ est très mal représenté par le premier plan factoriel (la somme des  $\cos^2$  est autour de 12%). De ce fait, la proximité de PEUGEOTRCZ et JAGUARF sur le graphique n'est pas interprétable. PEUGEOTRCZ est bien représenté sur le plan formé par le 3e et 4e axe.

**f)** Peut-on dire que les individus LANCIA et LANDROVER ont un profil semblable? Si oui quel est-il?

Les voitures LANCIA et LANDROVER sont bien représentées sur le premier plan factoriel, on peut donc interpréter leur proximité sur le graphique. Elles appartiennent au profil des voitures massives, ayant un poids et une longueur très élevés. En plus, elles sont assez puissantes.

**g)** Interpréter la représentation graphique des individus.

- En haut à droite, on a des voitures massives qui sont assez puissantes, telles que LANCIA et LANDROVER. Elles sont cependant moins puissantes que les voitures sportives qui se trouvent plus à droite selon l'axe 1.
- En bas à droite, on distingue les voitures sportives : elles sont plus compactes mais extrêmement puissantes : FERRARI, FORDMUSTANG et JAGUARF.

- En haut à gauche, on voit la classe des voitures assez massives mais peu puissantes. Les représentants de cette classe sont les voitures BMW53 et Hyundai.
- En bas à gauche, on reconnaît les voitures de petite taille et peu puissantes, dont FIAT500, DACIA, RENAULTCLIO, AUDIA1 etc.
- Les individus apparaissant en jaune et en vert sur le graphique sont mal représentés sur le premier plan factoriel et, de ce fait, leur position et leur proximité aux autres individus ne sont pas interprétables. Ils ont été écrasés lors de la projection et se retrouvent, en conséquence, plutôt au centre.
- En regardant les contributions des individus aux axes du premier plan factoriel, on a constaté que FERRARI avait une contribution extrêmement forte au premier axe. On a essayé de faire l'ACP sans cet individu et de le projeter comme individu supplémentaire. On a conclu qu'il n'y avait pas d'améliorations importantes de l'ACP.

## 2 - Classification

### A) K-means

Import des packages nécessaires.

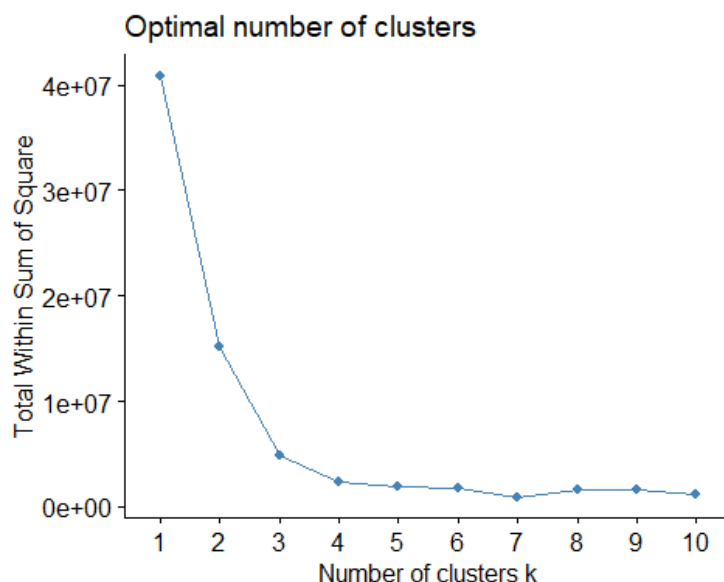
- *factoextra* afin de trouver le nombre optimal de clusters.
- *cluster* afin de réaliser l'algorithme k-means

```
library('factoextra')
```

```
library('cluster')
```

Avant de réaliser la classification K-means, il est nécessaire de savoir combien de classes l'algorithme va devoir créer. Nous allons ainsi employer la « méthode du coude ».

```
fviz_nbclust(cars, kmeans, method='wss')
```



Nous voyons que le coude apparaît au niveau de 3 clusters. Nous allons donc indiquer à notre machine de réaliser un k-means avec 3 classes. Nous réaliserons par la suite d'autres classifications qui varieront par leurs nombre de clusters. En effet, leurs études peuvent aussi être pertinentes, et nous permettre de comprendre des relations diverses.

Commençons tout d'abord par centrer et réduire nos données.

```
cars2<-scale(cars)
```

Maintenant que cela ait été réalisé, nous pouvons passer à la classification.

### 1. K-means à 3 clusters

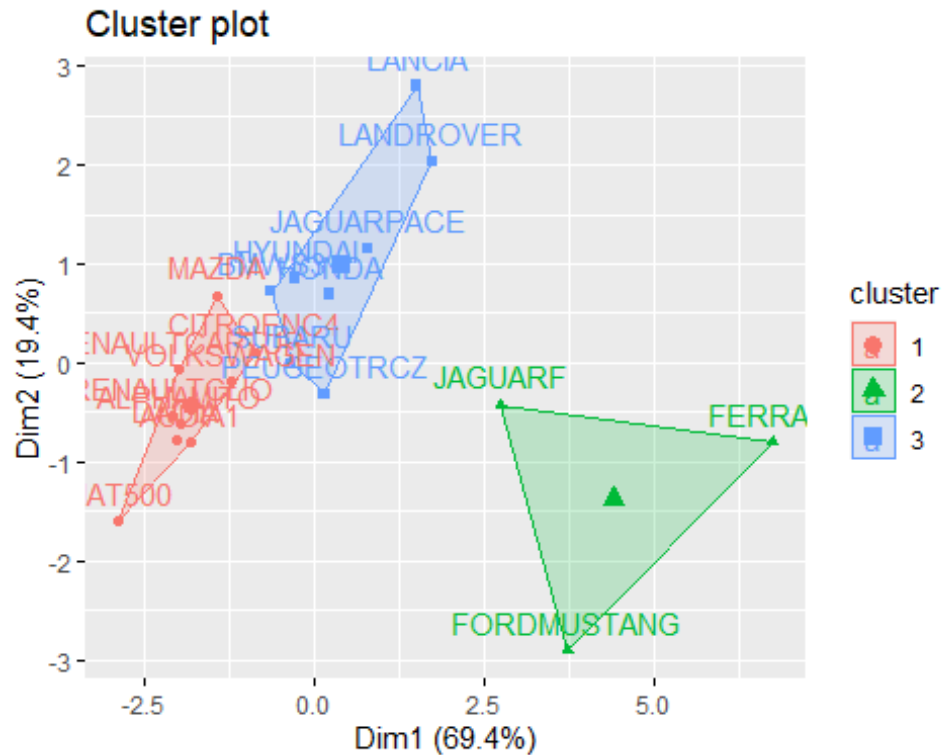
On commence avec 3 clusters et on réalise l'algorithme 10 fois afin de faire varier les centroïdes initiaux. En effet, les premiers centroïdes sont choisis de manière aléatoire et en fonction de ces derniers, la répartition finale pourra être impactée. C'est pourquoi il est important d'effectuer plusieurs fois l'algorithme.

```
cars.kmeans<-kmeans(cars2,centers=3,nstart=10)
cars.kmeans

## K-means clustering with 3 clusters of sizes 9, 3, 8
##
## Cluster means:
##          CYL          PUIS          LON          LAR          POIDS          VITESSE
## 1 -0.70991645 -0.5780425 -0.3821539 -0.8340633 -0.8080689 -0.5693279
## 2  1.88483231  2.0384256 -0.5411696  1.1325701  0.5369576  1.9825411
## 3  0.09184389 -0.1141118  0.6328617  0.5136074  0.7077184 -0.1029591
##          ACCEL          CO2
## 1  0.69542538 -0.60241503
## 2 -1.93408579  1.99764687
## 3 -0.05707138 -0.07140066
##
## Clustering vector:
##          ALPHAMITO          AUDIA1          CITROENC4          JAGUARF          PEUGEOTRCZ
##          1              1              1              2              3
##          LANDROVER          RENAULTCLIO          BMWS3          DACIA          HYUNDAI
##          3              1              3              1              3
##          LANCIA          RENAULTCAPTUR          FORDMUSTANG          FIAT500          HONDA
##          3              1              2              1              3
##          FERRARI          SUBARU          MAZDA          VOLKSWAGEN          JAGUARPACE
##          2              3              1              1              3
##
## Within cluster sum of squares by cluster:
## [1]  9.246285 19.947844 17.929894
## (between_SS / total_SS =  69.0 %)
##
## Available components:
##
## [1] "cluster"          "centers"          "totss"            "withinss"         "tot.withinss"
## [6] "betweenss"        "size"             "iter"             "ifault"
```

Nous voici donc avec nos 3 clusters, composés respectivement de 9, 3 et 8 voitures. Nous pouvons réaliser un affichage visuel afin de mieux observer nos groupes.

```
fviz_cluster(cars.kmeans,data=cars)
```



Nous distinguons bien le 2<sup>e</sup> groupe, composé de JaguarF, Ferrari et FordMustang. Il se différencie des 2 autres clusters, qui sont en effet plus proches l'un de l'autre. On peut déjà penser que le cluster vert sera celui de voitures extrêmement puissantes, sportives, à l'inverse des véhicules présents dans les autres groupes. Afin désormais de passer à une analyse plus précise des groupes, nous allons afficher les valeurs moyennes de chaque variable pour chacun d'entre eux.

```
aggregate(cars, by=list(cluster=cars.kmeans$cluster), mean)
```

```
## cluster CYL PUIS LON LAR POIDS VITESSE ACCEL
## 1 1 1129.667 98.77778 411.8889 174.3333 1147.111 180.6667 11.77778
## 2 2 4736.000 473.66667 403.3333 193.0000 1729.000 281.6667 4.866667
## 3 3 2244.000 165.25000 466.5000 187.1250 1802.875 199.1250 9.800000
## CO2
## 1 111.3333
## 2 304.3333
## 3 150.7500
```

Nous pouvons finalement réaliser une description plus détaillée :

- Le 1<sup>er</sup> groupe est composé des voitures les moins puissantes. En effet, elles disposent d'un moteur de petite cylindrée, d'une vitesse maximale peu élevée et d'une accélération lente. C'est donc logiquement aussi celles qui polluent le moins.
- Le 2<sup>e</sup> groupe est bien celui qui va réunir les véhicules les plus puissants, sportifs, disposant d'une grosse cylindrée, de vitesses de pointe élevées ainsi que d'un temps d'accélération bien inférieur aux autres voitures. C'est aussi ce groupe qui émet le plus de CO<sub>2</sub>.

- Le 3<sup>e</sup> groupe représente des véhicules que l'on pourrait qualifier «d'intermédiaire» en termes de puissance. En revanche, les voitures s'y trouvant sont relativement imposantes. Elles sont en effet les plus lourdes et les plus longues. Les véhicules s'y trouvant peuvent donc correspondre à des voitures familiales.

## 2. K-means à 2 clusters

On peut désormais réaliser un k-means pour 2 clusters.

```
cars.kmeans2<-kmeans(cars2,centers=2,nstart=10)
cars.kmeans2

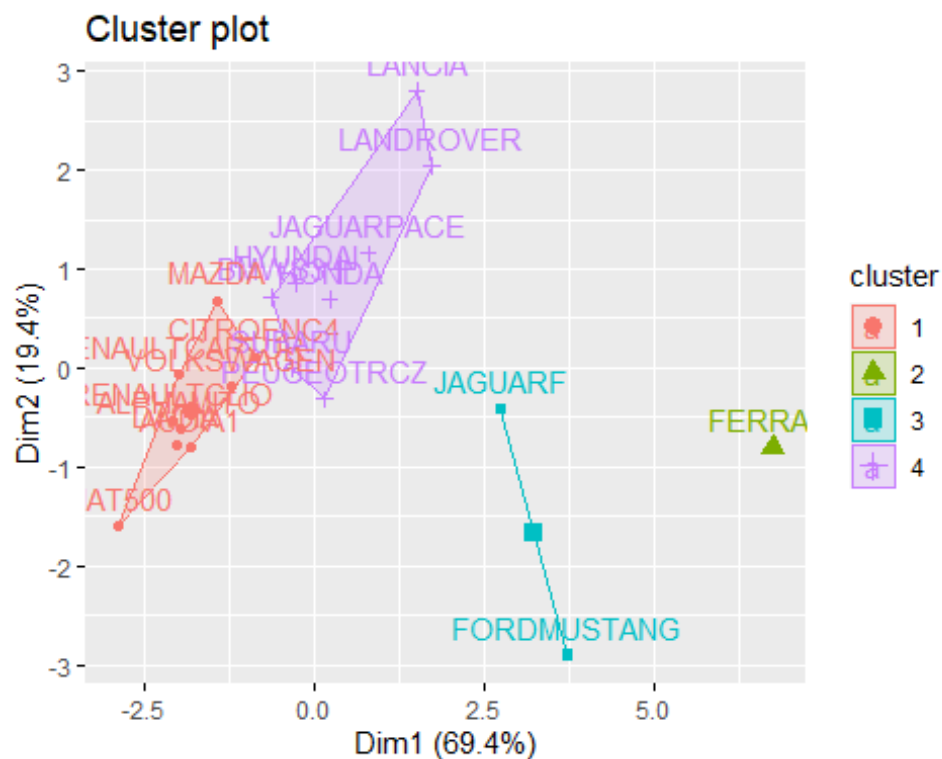
## K-means clustering with 2 clusters of sizes 17, 3
##
## Cluster means:
##      CYL      PUIS      LON      LAR      POIDS      VITESSE
## 1 -0.3326175 -0.3597222  0.09550051 -0.1998653 -0.09475722 -0.3498602
## 2  1.8848323  2.0384256 -0.54116956  1.1325701  0.53695760  1.9825411
##      ACCEL      CO2
## 1  0.3413093 -0.3525259
## 2 -1.9340858  1.9976469
##
## Clustering vector:
##      ALPHAMITO      AUDIA1      CITROENC4      JAGUARF      PEUGEOTRCZ
##      1      1      1      2      1
##      LANDROVER      RENAULTCLIO      BMWS3      DACIA      HYUNDAI
##      1      1      1      1      1
##      LANCIA      RENAULTCAPTUR      FORDMUSTANG      FIAT500      HONDA
##      1      1      2      1      1
##      FERRARI      SUBARU      MAZDA      VOLKSWAGEN      JAGUARPACE
##      2      1      1      1      1
##
## Within cluster sum of squares by cluster:
## [1] 57.11066 19.94784
## (between_SS / total_SS =  49.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"      "withinss"      "tot.withinss"
## [6] "betweenss"    "size"      "iter"      "ifault"
```

```
fviz_cluster(cars.kmeans2,data=cars)
```



```
## 1  0.69542538 -0.60241503
## 2 -2.22578397  3.01701483
## 3 -1.78823669  1.48796289
## 4 -0.05707138 -0.07140066
##
## Clustering vector:
##      ALPHAMITO      AUDIA1      CITROENC4      JAGUARF      PEUGEOTRCZ
##           1           1           1           3           4
##      LANDROVER  RENAULTCLIO      BMWS3      DACIA      HYUNDAI
##           4           1           4           1           4
##      LANCIA  RENAULTCAPTUR  FORDMUSTANG      FIAT500      HONDA
##           4           1           3           1           4
##      FERRARI      SUBARU      MAZDA  VOLKSWAGEN  JAGUARPACE
##           2           4           1           1           4
##
## Within cluster sum of squares by cluster:
## [1]  9.246285  0.000000  6.960978 17.929894
## (between_SS / total_SS =  77.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

`fviz_cluster(cars.kmeans4,data=cars)`



```
aggregate(cars, by=list(cluster=cars.kmeans4$cluster), mean)
```

##	cluster	CYL	PUIS	LON	LAR	POIDS	VITESSE	ACCEL
## 1	1	1129.667	98.77778	411.8889	174.3333	1147.111	180.6667	11.77778
## 2	2	6262.000	660.00000	491.0000	195.0000	1880.000	335.0000	4.10000
## 3	3	3973.000	380.50000	359.5000	192.0000	1653.500	255.0000	5.25000
## 4	4	2244.000	165.25000	466.5000	187.1250	1802.875	199.1250	9.80000
##	CO2							
## 1	1	111.3333						
## 2	2	380.0000						
## 3	3	266.5000						
## 4	4	150.7500						

Les groupes sont similaires aux cas de 3 clusters à l'exception de la Ferrari qui quitte celui des véhicules puissants. En effet, cette voiture se distingue par sa puissance encore plus grande. Elle dispose de la plus grosse cylindrée, mais aussi d'une vitesse de pointe bien plus élevée que les autres. Elle se place donc en haut du panier dans ce cadre.

En conclusion, le K-means à 3 clusters reste la méthode permettant de classifier de manière optimale nos véhicules.

## **B) Classification Ascendante Hiérarchique (Méthode de Ward)**

On commence par importer les packages nécessaires.

- *dendextend* pour manipuler les dendrogrammes
- *factoextra* afin de trouver le nombre optimal de clusters
- *cluster* afin de réaliser la classification ascendante hiérarchique via 'agnes'

```
library('dendextend')
library('factoextra')
library('cluster')
```

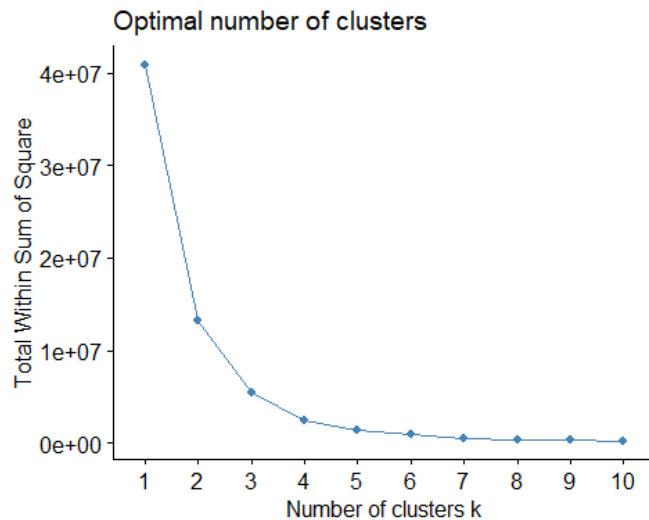
Centrons les données :

```
cars2<-scale(cars)
```

Étudions maintenant le nombre optimal de clusters à l'aide de la méthode du coude.

```
fviz_nbclust(cars, FUN=hcut, method = "wss")
```

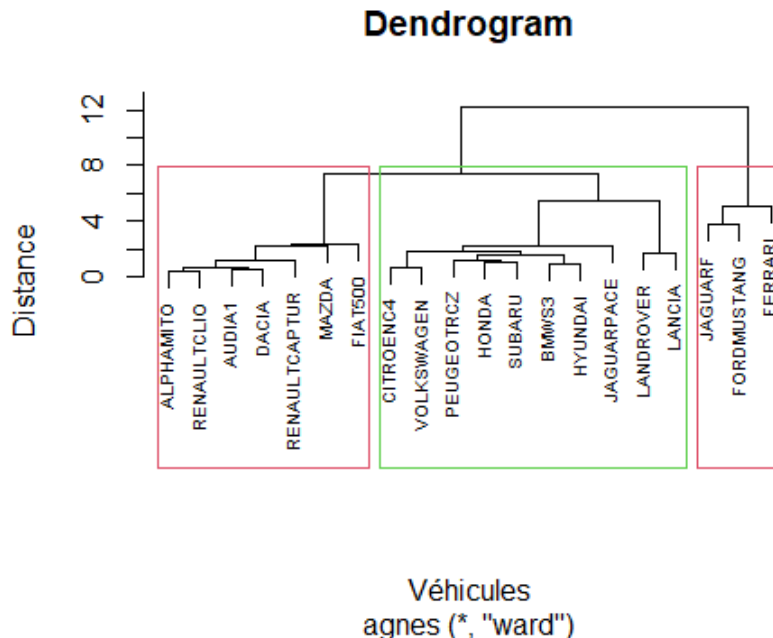




A l'instar des k-means, on trouve le coude au niveau de 3 clusters. Il est maintenant temps de lancer l'algorithme de la classification ascendante hiérarchique avec la méthode de Ward pour 3 clusters.

```
cars.ward<-agnes(cars2,method="ward")

pltree(cars.ward, cex = 0.6, main = "Dendrogram",ylab='Distance',xlab='Véhicules')
rect.hclust(cars.ward, k = 3, border = 2:3)
```



On reconnaît les groupes trouvés avec les k-means. Les véhicules s'y trouvant sont presque les mêmes. L'exception est pour CitroenC4 et Volkswagen, qui passent du groupe des véhicules les moins puissants à celui des véhicules dits «intermédiaires» ou familiaux. Observons plus en détail ces groupes.

```
cars.groups<-cutree(cars.ward,k=3)
table(cars.groups)

## cars.groups
## 1 2 3
## 7 10 3

cbind(cars,cars.groups)

##          CYL  PUIS  LON  LAR  POIDS  VITESSE  ACCEL  CO2  cars.groups
## ALPHAMITO    875   105  406  172   1130     184   11.4   98           1
## AUDIA1       999    95  397  174   1065     186   10.9  103           1
## CITROENC4   1199   130  442  182   1280     196   10.1  115           2
## JAGUARF     2995   340  447  192   1587     260    5.7  234           3
## PEUGEOTRCZ  1997   160  428  184   1370     220    8.2  130           2
## LANDROVER   2993   256  483  191   2570     180    9.3  203           2
## RENAULTCLIO   898    90  406  173   1092     182   12.2  105           1
## BMW53       1995   116  462  181   1570     198   11.1  109           2
## DACIA        898    90  406  173    962     175   11.1  116           1
## HYUNDAI     1995   136  447  185   1751     184   10.9  139           2
## LANCIA      2776   177  522  200   2315     193   11.5  207           2
## RENAULTCAPTUR 898    90  412  178   1180     171   13.0  113           1
## FORDMUSTANG 4951   421  272  192   1720     250    4.8  299           3
## FIAT500     1242    69  355  163    905     160   12.9  115           1
## HONDA       2199   150  472  184   1632     212    9.4  138           2
## FERRARI     6262   660  491  195   1880     335    4.1  380           3
## SUBARU      1998   147  445  178   1440     198    9.3  141           2
## MAZDA       1560   115  458  175   1490     180   13.7  138           1
## VOLKSWAGEN  1598   105  425  179   1220     192   10.7   99           2
## JAGUARPACE  1999   180  473  194   1775     208    8.7  139           2

aggregate(cars, by=list(cluster=cars.groups), mean)

##   cluster    CYL    PUIS    LON    LAR    POIDS  VITESSE    ACCEL
## 1      1 1052.857  93.42857 405.7143 172.5714 1117.714 176.8571 12.171429
## 2      2 2074.900 155.70000 459.9000 185.8000 1692.300 198.1000  9.920000
## 3      3 4736.000 473.66667 403.3333 193.0000 1729.000 281.6667  4.866667
##          CO2
## 1 112.5714
## 2 142.0000
## 3 304.3333
```

On retrouve donc bien des groupes proches de ceux trouvés dans la question précédente. Les 3 voitures les plus puissantes, puis un groupe intermédiaire et enfin les véhicules les moins puissants et moins polluants.

### **3 - Races de chiens**

Importation des librairies pertinentes lors de notre exercice.

- *factoextra* afin de trouver le nombre optimal de clusters.
- *FactoMineR* afin de
- *fbasics* afin de réaliser les statistiques descriptives

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

## Statistiques descriptives et visualisation des données

```
View(chiens)
```

```
basicStats(chiens)
```

##	TAI	POI	VEL	INT	AFF	AGR
## nobs	27.000000	27.000000	27.000000	27.000000	27.000000	27.000000
## NAs	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
## Minimum	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
## Maximum	3.000000	3.000000	3.000000	3.000000	2.000000	2.000000
## 1. Quartile	1.500000	1.000000	1.000000	1.000000	1.000000	1.000000
## 3. Quartile	3.000000	2.000000	3.000000	2.000000	2.000000	2.000000
## Mean	2.296296	1.888889	1.962963	1.925926	1.518519	1.481481
## Median	3.000000	2.000000	2.000000	2.000000	2.000000	1.000000
## Sum	62.000000	51.000000	53.000000	52.000000	41.000000	40.000000
## SE Mean	0.167220	0.134327	0.164356	0.140471	0.097991	0.097991
## LCL Mean	1.952571	1.612776	1.625125	1.637185	1.317096	1.280059
## UCL Mean	2.640021	2.165002	2.300801	2.214667	1.719941	1.682904
## Variance	0.754986	0.487179	0.729345	0.532764	0.259259	0.259259
## Stdev	0.868899	0.697982	0.854017	0.729907	0.509175	0.509175
## Skewness	-0.573102	0.137157	0.065905	0.103736	-0.070045	0.070045
## Kurtosis	-1.474168	-1.031098	-1.676546	-1.190684	-2.067607	-2.067607
##	FON					
## nobs	27.000000					
## NAs	0.000000					
## Minimum	1.000000					
## Maximum	3.000000					
## 1. Quartile	1.000000					
## 3. Quartile	3.000000					
## Mean	1.925926					
## Median	2.000000					
## Sum	52.000000					
## SE Mean	0.159468					
## LCL Mean	1.598135					
## UCL Mean	2.253717					
## Variance	0.686610					
## Stdev	0.828619					
## Skewness	0.128768					
## Kurtosis	-1.586063					

```
str(chiens)
```

```
## 'data.frame': 27 obs. of 7 variables:
## $ TAI: int 3 1 3 2 1 3 1 1 2 3 ...
## $ POI: int 2 1 2 2 1 3 1 1 1 2 ...
## $ VEL: int 3 1 3 2 1 1 2 1 1 3 ...
## $ INT: int 2 1 3 2 2 3 3 1 2 2 ...
## $ AFF: int 2 1 2 2 2 1 2 2 2 2 ...
```

```
## $ AGR: int  2 2 2 2 1 2 1 1 2 1 ...
## $ FON: int  3 2 3 1 1 3 1 1 1 1 ...

dim(chiens)

## [1] 27  7

summary(chiens)

##          TAI          POI          VEL          INT
## Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
## 1st Qu.:1.500   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
## Median :3.000   Median :2.000   Median :2.000   Median :2.000
## Mean   :2.296   Mean   :1.889   Mean   :1.963   Mean   :1.926
## 3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:2.000
## Max.   :3.000   Max.   :3.000   Max.   :3.000   Max.   :3.000
##          AFF          AGR          FON
## Min.   :1.000   Min.   :1.000   Min.   :1.000
## 1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
## Median :2.000   Median :1.000   Median :2.000
## Mean   :1.519   Mean   :1.481   Mean   :1.926
## 3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:3.000
## Max.   :2.000   Max.   :2.000   Max.   :3.000
```

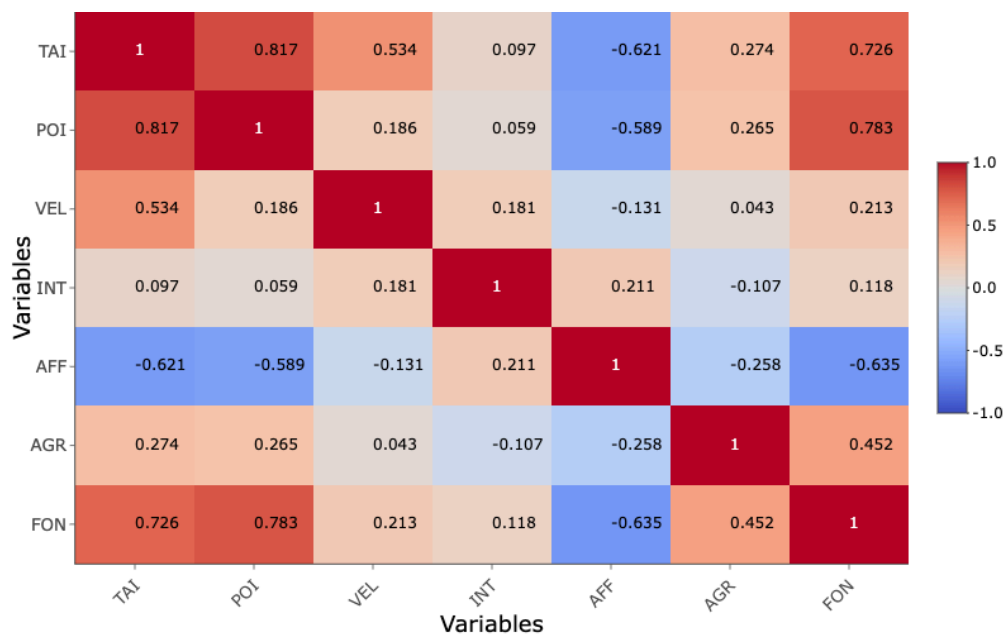
### Valeurs manquantes et outliers :

```
sum(is.na(chiens))
```

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

### Matrice de corrélation :

```
heatmaply_cor(cor(chiens),xlab = "Variables", ylab = "Variables",k_col =
               ,k_row = 2,dendrogram=FALSE, cellnote = cor(chiens))
```



La matrice de corrélation nous montre logiquement que le poids et la taille ont la corrélation la plus forte parmi toutes les variables. La seconde corrélation la plus importante est celle entre la fonction du chien et son poids. Cela nous indique donc que les chiens de garde seront des chiens lourds contrairement aux chiens de compagnie.

De l'autre côté, les corrélations négatives les plus fortes sont celles entre affection et fonction et entre taille et affection. D'une part, les chiens sélectionnés pour la chasse ou la garde ne le seraient donc pas pour leur affection. D'autre part, plus les chiens sont petits, plus ils seront affectueux.

Les corrélations les plus faibles sont entre agressivité et vélocité et intelligence et poids.

### Normalisation des données :

```
chiens2 <- scale(chiens)
```

Test d'indépendance non faisable avec données normalisées. Nous garderons les données non transformées qui semblent déjà normalisées.

```
chisq.test(chiens)
```

```
## Warning in chisq.test(chiens): Chi-squared approximation may be incorrect
```

```
##
```

```
## Pearson's Chi-squared test
```

```
##
```

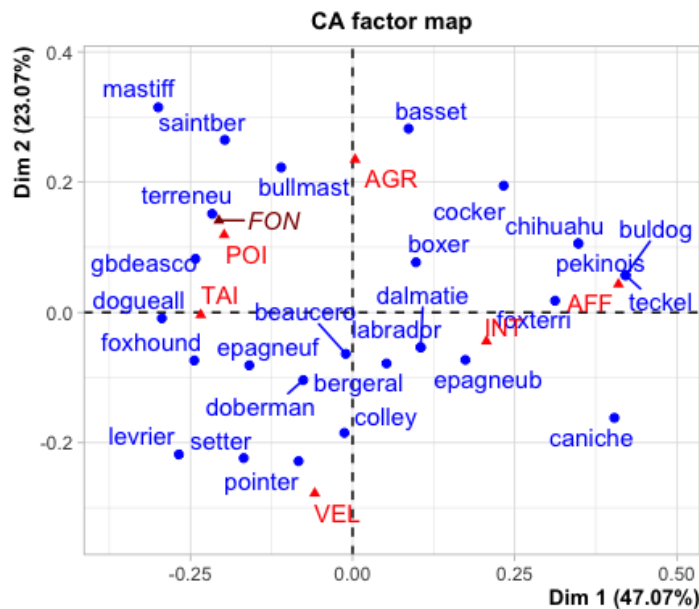
```
## data: chiens
```

```
## X-squared = 36.234, df = 156, p-value = 1
```

## A) Analyse des correspondances multiples à 2 dimensions

Commande permettant de réaliser l'Analyse des Correspondances Multiples, en mettant FON comme variable supplémentaire, et permettant d'obtenir les graphiques associés.

```
sol.mca <- CA(chiens, col.sup = 7, graph = TRUE)
```



Le premier plan factoriel explique 70,14% de la variance. Lorsque nous prenons en compte les 3 premières dimensions, nous expliquons environ 87,87% de la variance.

## 1. Valeurs propres

```
sol.mca$eig
```

```
##          eigenvalue percentage of variance cumulative percentage of variance
## dim 1 0.04910249          47.066103          47.06610
## dim 2 0.02406373          23.065751          70.13185
## dim 3 0.01852523          17.756948          87.88880
## dim 4 0.01015600           9.734806          97.62361
## dim 5 0.00247921           2.376392         100.00000
```

## 2. Cos<sup>2</sup>

Les commandes suivantes calculant les cos<sup>2</sup> nous permettent de connaître la qualité de représentation de nos variables et individus par rapport aux dimensions. Autrement dit, elles nous laissent voir la qualité de représentation des races selon l'axe correspondant. En général lorsque le cos<sup>2</sup> est inférieur à 0.5 nous disons que la variable s'est faite écraser.

```
sol.mca$row$cos2
```

```
##          Dim 1          Dim 2          Dim 3          Dim 4          Dim 5
## beaucero 0.005807173 0.2139892906 0.7316594970 8.834864e-06 4.853520e-02
## basset 0.035525890 0.3800812140 0.3433560018 2.290755e-01 1.196136e-02
## bergeral 0.178706111 0.3999829664 0.0008718915 3.726180e-01 4.782103e-02
## boxer 0.407175036 0.2529269987 0.2536908467 4.149595e-03 8.205752e-02
## bulldog 0.947418604 0.0174288877 0.0250503695 9.310197e-03 7.919422e-04
## bullmast 0.094280369 0.3851078148 0.4388063801 8.052358e-02 1.281855e-03
## caniche 0.752640585 0.1209541512 0.0649674668 4.662875e-02 1.480905e-02
## chihuahua 0.610522070 0.0565707349 0.1117881965 2.197596e-01 1.359397e-03
## cocker 0.421736882 0.2944777894 0.0313052559 5.192732e-02 2.005528e-01
## colley 0.003257470 0.6980114830 0.0063987124 2.607813e-01 3.155103e-02
## dalmatie 0.337576226 0.0878056279 0.0605365110 4.652717e-01 4.880993e-02
## doberman 0.107286494 0.1999787827 0.0260579355 6.666032e-01 7.354429e-05
## dogueall 0.699274260 0.0006470683 0.1905202588 1.249658e-02 9.706183e-02
## epagneub 0.448788047 0.0787214892 0.4456071990 5.373303e-03 2.150996e-02
## epagneuf 0.495477098 0.1287244149 0.2737430085 4.067145e-03 9.798833e-02
## foxhound 0.517410345 0.0471876829 0.4255649968 9.596967e-03 2.400089e-04
## foxterri 0.636305524 0.0021193474 0.2159948147 1.191945e-01 2.638583e-02
## gbdeasco 0.625017587 0.0722111170 0.2554144009 3.113473e-03 4.424342e-02
## labrador 0.337576226 0.0878056279 0.0605365110 4.652717e-01 4.880993e-02
## levrier 0.506092239 0.3353213512 0.0897666542 6.750223e-02 1.317526e-03
## mastiff 0.457160299 0.5051713041 0.0003475691 3.658947e-02 7.313571e-04
## pekinois 0.610522070 0.0565707349 0.1117881965 2.197596e-01 1.359397e-03
## pointer 0.077375785 0.5813962362 0.2484331542 9.191986e-02 8.749657e-04
## saintber 0.310342000 0.5614213285 0.1248714422 2.125502e-03 1.239728e-03
## setter 0.355991695 0.6315013064 0.0107238097 2.377294e-04 1.545460e-03
## teckel 0.947418604 0.0174288877 0.0250503695 9.310197e-03 7.919422e-04
## terreneu 0.305234990 0.1493392854 0.4571643317 8.805342e-02 2.079687e-04
```

```
sol.mca$col$cos2
```

##		Dim 1	Dim 2	Dim 3	Dim 4	Dim 5
##	TAI	0.8671179806	0.0002171733	0.01587190	0.016810593	9.998236e-02
##	POI	0.5200622076	0.1883508845	0.10734685	0.111514077	7.272598e-02
##	VEL	0.0344164508	0.7806099606	0.16993696	0.004563839	1.047279e-02
##	INT	0.4084511690	0.0187117722	0.44900100	0.123828704	7.353419e-06
##	AFF	0.8402406571	0.0093966550	0.03112363	0.117465846	1.773209e-03
##	AGR	0.0001332396	0.4821428878	0.32440007	0.193158778	1.650216e-04

Ici nous pouvons tirer comme conclusion que notre premier axe oppose les chiens affectueux et intelligents aux chiens plutôt imposants. Cela peut nous laisser comprendre que plus un chien est affectueux, plus il sera de petite taille. C'est une conclusion similaire à celle trouvée à l'aide de la matrice de corrélation.

Quant à l'axe 2, nous voyons qu'il oppose la vélocité à l'agressivité des chiens.

### 3. Contribution des variables à la constitution des dimensions

```
sol.mca$row$contrib
```

##		Dim 1	Dim 2	Dim 3	Dim 4	Dim 5
##	beaucero	0.01044938	0.78570268	3.489590743	7.686106e-05	1.729707369
##	basset	0.35499504	7.74986896	9.094145201	1.106718e+01	2.367271045
##	bergeral	0.27983790	1.27805420	0.003618841	2.821057e+00	1.483119854
##	boxer	0.78071791	0.98957526	1.289311105	3.846801e-02	3.116173422
##	bulldog	9.65749480	0.36252076	0.676825382	4.588411e-01	0.159884475
##	bullmast	1.07449620	8.95584524	13.255510819	4.436982e+00	0.289342872
##	caniche	11.09717794	3.63903585	2.538988747	3.323989e+00	4.324564234
##	chihuahua	5.77576557	1.09204529	2.803135331	1.005164e+01	0.254709194
##	cocker	3.69561812	5.26548875	0.727114307	2.199998e+00	34.806820052
##	colley	0.01412059	6.17412778	0.073519902	5.465499e+00	2.708795576
##	dalmatie	0.82630281	0.43856136	0.392757682	5.506238e+00	2.366278187
##	doberman	0.55174292	2.09853530	0.355198929	1.657449e+01	0.007490847
##	dogueall	7.65151288	0.01444743	5.525618032	6.611072e-01	21.034804402
##	epagneub	2.47169340	0.88468213	6.504972143	1.430789e-01	2.346298031
##	epagneuf	1.89845961	1.00642019	2.780102055	7.534394e-02	7.436051096
##	foxhound	4.86647037	0.90562472	10.609252896	4.364092e-01	0.044709156
##	foxterri	6.64062781	0.04513217	5.974845745	6.014237e+00	5.453865418
##	gbdeasco	4.39759322	1.03673442	4.763302056	1.059129e-01	6.165406799
##	labrador	0.82630281	0.43856136	0.392757682	5.506238e+00	2.366278187
##	levrier	5.37293738	7.26413570	2.526021635	3.464824e+00	0.277033084
##	mastiff	6.72362627	15.16052951	0.013549276	2.601789e+00	0.213036867
##	pekinois	5.77576557	1.09204529	2.803135331	1.005164e+01	0.254709194
##	pointer	0.61400213	9.41407522	5.225336795	3.526591e+00	0.137513730
##	saintber	3.17108839	11.70569869	3.381978541	1.050050e-01	0.250890583
##	setter	2.30258836	8.33472610	0.183850776	7.434302e-03	0.197981397
##	teckel	9.65749480	0.36252076	0.676825382	4.588411e-01	0.159884475
##	terreneu	3.51111781	3.50530486	13.938734667	4.897092e+00	0.047380454

```
sol.mca$col$contrib
```

##		Dim 1	Dim 2	Dim 3	Dim 4	Dim 5
##	TAI	23.133110335	0.01182232	1.122340	2.1683022	52.828638777
##	POI	13.614027492	10.06096486	7.448358	14.1137349	37.706058292

```
## VEL 1.227292277 56.80108708 16.062372 0.7868521 7.396644047
## INT 15.093009269 1.41088383 43.976730 22.1226911 0.005381648
## AFF 46.928410445 1.07089299 4.607468 31.7193806 1.961473171
## AGR 0.004150183 30.64434892 26.782731 29.0890391 0.101804065
```

Le code suivant nous permet d'avoir l'information sur la contribution et cosinus carré de notre variable supplémentaire FON qui n'a pas été prise en compte lors de la construction de notre plan factoriel.

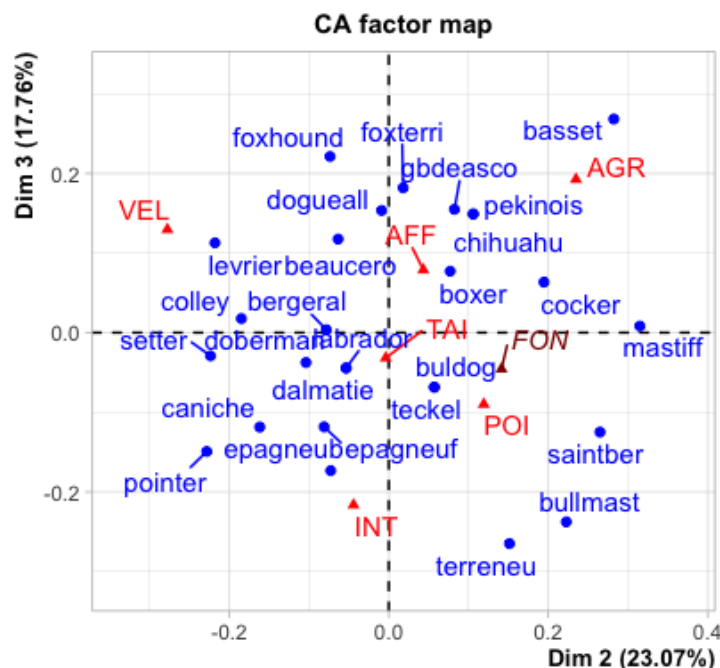
```
sol.mca$col.sup

## $coord
##          Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## FON -0.2061405  0.1413847 -0.04560565  0.04578968  0.05184196
##
## $cos2
##          Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
## FON 0.3717421  0.174872  0.01819502  0.01834216  0.02351138
```

## B) Analyse des correspondances multiples à 3 dimensions

L'objectif est d'avoir une analyse plus détaillée et précise des données et de couvrir une plus grande variance. Par ailleurs, les variables AGR et INT étaient mal représentées par le 1er plan factoriel. Nous pencher sur ces nouvelles dimensions nous semblent donc pertinents pour mettre en lumière ces 2 variables.

```
sol2.mca <- CA(chiens, col.sup = 7, graph = TRUE, axes = c(2,3))
```





Graphiquement et à l'aide du tableau de contribution des variables réalisé précédemment, la dimension 3 oppose l'agressivité à l'intelligence. Cela comble ainsi notre manque d'information pour 2 variables.

Ce plan factoriel explique par ailleurs 41 % de la variance.

## C) Description des différentes races de chiens

### 1. Interprétation des races

**Beaucero** : D'après les  $\text{Cos}^2$ , cette race de chien ne serait pas si bien représentée dans notre premier axe principal. Graphiquement cela se confirme du fait de son positionnement près du centre. Ainsi nous pouvons conclure que l'information se ferait écraser. Toutefois, nous pouvons interpréter notre deuxième graphique. Cela nous permet de dire que ce serait un chien affectif et assez agressif.

**Basset** : Le  $\text{Cos}^2$  serait aussi inférieur à 0.5 et donc la représentation ne serait pas si bonne sur notre premier plan factoriel. Cette race serait apparemment agressive, ce qui est en accord avec l'autre graphique où l'on a une proximité avec AGR aussi. Nous aurons recours à notre deuxième graphique et donc la Dim3 et nous pouvons dire que ce sera une race de chien agressive avec une intelligence faible et un poids bas.

**Bergeral** : Bien que graphiquement il soit situé au centre, le  $\text{Cos}^2$  nous permet de savoir qu'il est bien représenté dans notre premier plan factoriel. Nous pouvons conclure que c'est un chien ayant un équilibre entre l'intelligence et une affection moyenne. Il est aussi assez rapide. Il dispose de notes moyennes quant aux autres caractéristiques.

**Boxer** : D'après le  $\text{Cos}^2$  il est bien représenté donc on peut conclure que c'est un chien moyen dans les caractéristiques donc c'est un chien qui, selon l'ACM serait de taille moyen/ grand et assez affectif. La troisième dimension nous confirme son affection.

**Bulldog** : Cette race de chien est particulièrement bien représentée dans notre premier plan factoriel avec un  $\text{Cos}^2$  supérieur à 0.95 Il semblerait être un chien affectif, intelligent et dont le poids n'est pas si important.

**Bullmast** : D'après la Dim3, ce serait un type de chien intelligent. Selon l'axe deux, cela semble un chien agressif. Cependant ce chien est majoritairement représenté dans l'axe 3, on peut ainsi conclure qu'il est plus fiable de dire que c'est un chien d'une très grande intelligence et d'une agressivité modérée.

**Caniche** : Le  $\text{Cos}^2$  nous permet de dire que le premier plan factorielle explique bien cette race de chien. Cette catégorie de chien serait très affective, de petite taille et donc d'un poids modéré. Les autres dimensions sont moins significatives. Toutefois, dans le premier plan factoriel, le deuxième axe nous permet de penser que ce serait un chien rapide.

**Chihuahua** : Ce sont des chiens très affectifs. Leur poids, taille et fonction est peu significatif. Quant à la vitesse, nous pouvons penser que ce seraient des chiens moyens en vitesse et agressivité. Cependant, étant des chiens dont la taille n'est pas importante, nous pourrions conclure que la vitesse ne sera pas une des caractéristiques clé. Nous remarquons aussi que leur qualité de représentation dans la Dim2 est très basse.

**Cocker** : D'après le premier plan factoriel, où la somme des  $\text{Cos}^2$  pour les deux premières dimensions s'élève à 0.7, nous pouvons dire que ce seront des chiens qui seront à la fois affectifs et agressifs. Du fait de l'opposition avec la variable TAILLE, nous pouvons dire que ce sont des chiens d'une taille modérée. Concernant la deuxième dimension, ce seront des chiens dont la vitesse est faible. Notre deuxième plan factoriel et donc la dimension 3 nous permettent de dire que l'intelligence de ce type de chien est moyenne.

**Colley** : Dans notre premier plan factoriel il est plutôt représenté plutôt par le 2ème axe . Nous pouvons dire que c'est un chien qui jouit d'une grande vitesse et d'une agressivité minimale.

**Dalmatien** : En analysant le  $\text{Cos}^2$  nous pouvons nous rendre compte qu'il est mal représenté dans le premier plan factoriel. Accompagné du fait que graphiquement il est au centre, nous pouvons conclure que l'information s'est fait écraser lors de l'ACM

**Doberman** : Nous avons le même cas que pour les dalmatiens, ce type de chiens est mal représenté dans notre premier axe factoriel . L'information semble être écrasée du fait de sa proximité au centre et le  $\text{Cos}^2$  en dessous de 0.5 quand nous faisons la somme.

**Dogueall** : En analysant le graphique, cela semble être un chien de grande taille et donc de poids. Par rapport à l'axe 2 il est moins bien représenté mais en le projetant sur le centre nous pourrions dire qu'il est moyennement rapide et moyennement agressif. Si nous prenions en compte la dimension 3, nous pouvons conclure que c'est un chien peu intelligent.

**Epagneub** : C'est un chien assez bien représenté. Il semblerait qu'il soit intelligent et plutôt affectueux. Il semble aussi peu agressif et assez rapide.

**Epagneuf** : C'est un chien de taille et poids important. On constate qu'il est aussi peu affectif et agressif mais assez intelligent.

**Foxhound** : Grand en taille et avec un poids important. On pourrait dire que c'est aussi un chien peu affectif. Grâce à la dimension 3 nous constatons qu'il est moyennement agressif. Il est aussi peu intelligent.

**Foxterri** : Cette race correspond plutôt à des chiens affectifs et intelligents.

**Labrador** : Il semblerait que son information ait été écrasée lors de la réalisation de l'analyse de correspondance. Le 1er axe nous permet tout de même de conclure que c'est un chien affectif et intelligent.

**Levrier** : Chien de taille importante et d'une grande vitesse. Il est peu affectif et agressif et il est très bien représenté dans notre 1er plan factoriel.

**Mastiff** : Le premier plan factoriel explique presque tout et nous pouvons dire que c'est un chien de taille importante et d'une grande agressivité. Il semblerait qu'ils sont peu affectifs et peu intelligents.

**Pékinois** : C'est une race de chien très affective et de taille moindre.

**Pointer** : Il est majoritairement expliqué par la deuxième dimension. Nous pouvons dire que ce sont des chiens rapides et d'une basse agressivité. Ils sont aussi assez intelligents.

**Saintber** : Chien plutôt agressif et imposant de par sa taille et son poids. Il s'avère peu affectif et peu ou moyennement intelligent.

**Setter** : Chien rapide. Entre moyen et grand pour la taille. Il est peu agressif.

**Teckel** : Ce type de chien est fortement affectif, petit ainsi que relativement léger.

**Terreneu** : Chien intelligent, grand et d'un poids important. Il semble en revanche peu affectif.

## **2. Interprétation des quadrants pour les dimensions 1 et 2**

**Côté Supérieur-Gauche** : On y retrouve des chiens imposants que ce soit en taille ou en poids et pour lesquels l'agressivité est élevée.

**Côté Supérieur-Droit** : Ce sont des chiens plutôt agressifs, intelligents et affectifs. Par ailleurs, ils sont peu imposants.

**Côté Inférieur-Droit** : Ce sont des chiens qui se caractérisent par leur grande vitesse, leurs affections et intelligence. Ils sont de taille modérée ou petite et peu agressifs.

**Côté Inférieur-Gauche** : Chiens de grande vitesse. De taille et poids grand ou modéré. Ils sont aussi peu agressifs.