UNIVERSITÉ PARIS 1 PANTHÉON SORBONNE

Ecole d'Economie de la Sorbonne

Master 1 Econométrie - Statistiques

ANALYSE DE DONNÉES

Abdoul Aziz Berrada - Baptiste Goumain - Hugo Hamon 14/01/2021

Importation des données et des librairies

Listing 1 – Données et librairies.

```
library(ggplot2)
library(funModeling)
library(caret)
library(e1071)
library(dplyr)
library(MASS)
library(fmsb)
library(FactoMineR)
library(factoextra)
library(stagarzer)
library(xtable)


rm(list=ls())
path = "..." # adapter le lien au dossier parent
setwd(path)

simu <- read.table(paste0(path, '/donnees/simu.txt'),
        header = T)

xsimutest <- read.table(paste0(path, '/donnees/xsimutest.txt'),
            header = T)

chiens <- read.table(paste0(path, '/donnees/chiens.txt'), header = T)
```

EXERCICE 1 - REGRESSION BINAIRE

Statistiques descriptives

Listing 2 – Exploration des données

```
df_status(simu)
```

TABLE 1 – Exploration des données

variable	q-zeros	p-zeros	q-na	p-na	q-inf	p-inf	unique
X_1	0	0	0	0	0	0	2000
X_2	0	0	0	0	0	0	2000
Y	0	0	0	0	0	0	2

On note que la base ne contient pas de données manquantes. Les données sont propres. Nous avons 2000 observations, 2 variables explicatives X_1 et X_2 et une variable de classe Y qui prend les valeurs 1 ou 2.
La variable Y est de type *integer*, elle ne prend que 2 valeurs, on va ainsi la convertir en facteurs et la stocker dans la variable *classes*.

Listing 3 – Détermination des fréquences

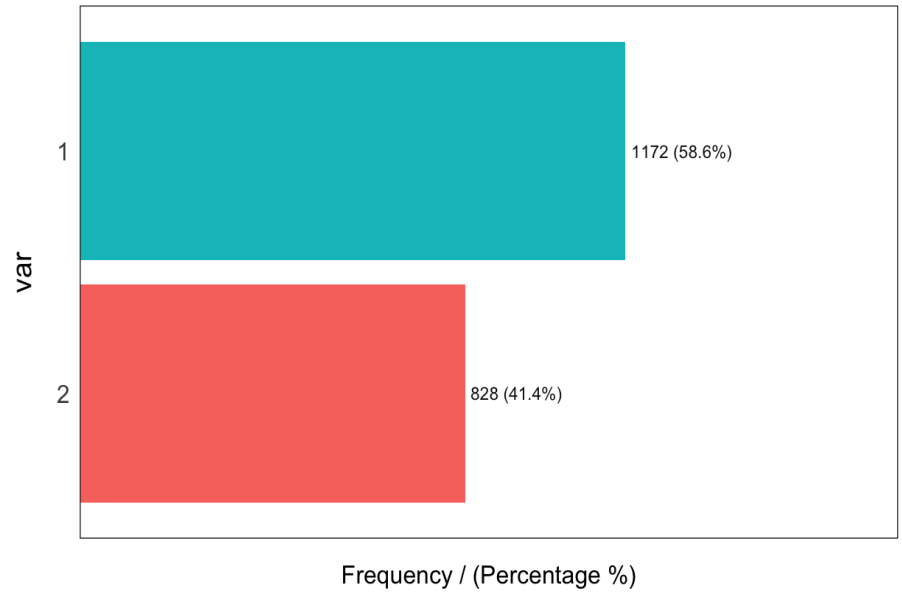
```
# convertir en facteur
simu$classes = as.factor(simu$Y)

# tableau de frequence
freq(simu$classes)
```

On détermine maintenant la part relative des valeurs 1 et 2 dans la variable *classes*.

TABLE 2 – Fréquences

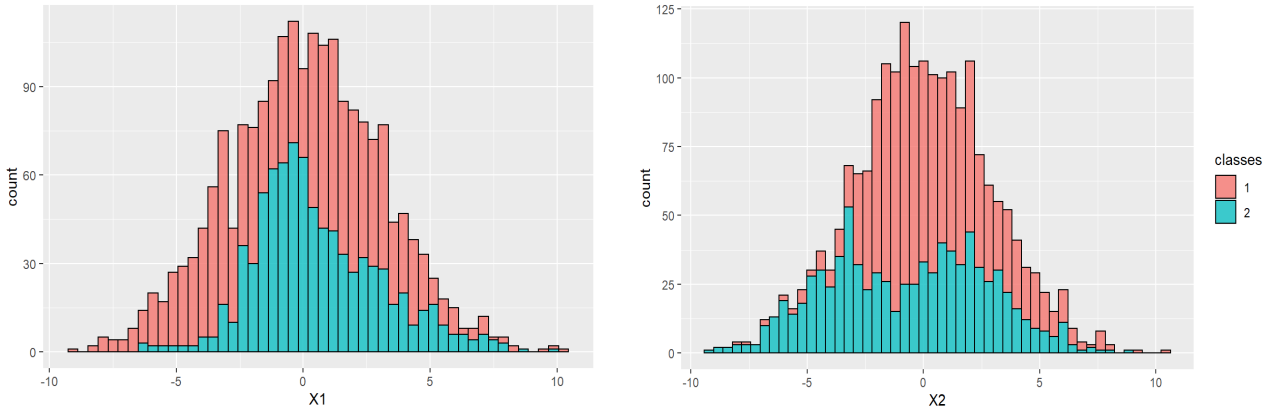
var	frequency	percentage	cumulative perc
1	1172	58.6	58.6
2	828	41.4	100.0



Les 2 populations sont relativement équilibrées (58,6% pour la classe 1 contre 41,4% pour la classe 2).
Déterminons maintenant l'histogramme de chaque variable par classe.

Listing 4 – Histogramme de chaque variable par classe

```
a=for( i in c("X1","X2") ){
  p<-ggplot(simu,aes_string(x=i,fill="classes"))+
  geom_histogram(bins=50,alpha=0.8,colour='black')
  print(p)
}
```



On constate que :

— La distribution de la variable X_1 est relativement similaire dans les deux classes. Les valeurs sont fortement concentrées entre -3 et 3, avec une prédominance de la classe 1 sur la classe 2 ;

— La répartition de la variable X_2 est différente dans les deux classes. Pour la classe 1 on a une distribution centrée en 0. Pour la classe 2 on a deux sous-populations. On note aussi une prédominance de la classe 1.

QUESTION 1 - Détermination du meilleur modèle possible pour modéliser P(Y|X1,X2).

La condition sine qua non pour que des prévisions soient possibles et performantes est que le phénomène à prévoir ne soit pas totalement aléatoire : il doit pouvoir être expliqué au moins partiellement. Pour diminuer la part d'aléatoire apparent, on va intervenir sur les données brutes avant qu'elles ne soient prises en compte par un algorithme apprenant. Nous procédons à la création de 3 nouvelles variables X_1^2, X_2^2 et X_1X_2 respectivement les carrés de X_1, X_2 et le produit de X_1 et X_2.

Listing 5 – Feature engineering

```
create_feature <- function(data){
  data$X1_sq = data$X1 * data$X1 # X1_sq
  data$X2_sq = data$X2 * data$X2 # X2_sq
  data$X1X2 = data$X1 * data$X2 # X1*X2
  return(data)
}
simu = create_feature(simu)
```

On va aussi créer 2 partitions de la variable *classes (Y)*, une 1^{re} **dtrain** qui sera utilisée pour entraîner l'algorithme et va comprendre 80% des observations ainsi qu'une 2^e **dtest** qui aura les 20% qui restent et permettra de tester les prédictions faites par l'algorithme.

Listing 6 – Create partition

```
set.seed(2021)
trainIndex <- createDataPartition(simu$classes, p = .8,
                                  list = FALSE,
                                  times = 1)
dtrain<-simu[trainIndex,]
dtest<-simu[-trainIndex,]
```

Pour déterminer les variables explicatives les plus pertinentes, nous allons faire du stepwise ; en tentant en premier temps la méthode **backward** (car on a peu de regresseurs) et en second lieu avec la **forward** donc en partant du modèle le plus petit (régression avec uniquement la constante) jusqu'au modèle le plus grand (régression avec toutes les variables créées). On prendra le modèle avec le critère AIC le plus faible.

Listing 7 – Feature Selection

```
m = glm(classes ~ 1, data = simu, family = binomial)
#backward
modelb <- glm(classes ~ 1, data = simu,
        family = binomial(link = 'logit')) %>%
        stepAIC( trace = F, direction="backward", scope=list(lower=m,
```

```
                    upper=~X1+X2+X1_sq+X2_sq+X1X2))
#forward
modelf <- glm(classes ~ 1, data = simu,
        family = binomial(link = 'logit')) %>%
        stepAIC( trace = F, direction="forward", scope=list(lower=m,
                    upper=~X1+X2+X1_sq+X2_sq+X1X2))
```

TABLE 3 – Modèle backward

	Dependent variable :
	classes
Constant	-0.347^{***}
	(0.045)
Observations	2,000
Log Likelihood	$-1,356.563$
Akaike Inf. Crit.	2,715.125
Note :	*p$<$0.1 ; **p$<$0.05 ; ***p$<$0.01

TABLE 4 – Modèle forward

	Dependent variable :
	classes
X2_sq	0.051^{***}
	(0.005)
X2	-0.155^{***}
	(0.018)
X1_sq	-0.041^{***}
	(0.005)
X1	0.141^{***}
	(0.019)
Constant	-0.474^{***}
	(0.069)
Observations	2,000
Log Likelihood	$-1,200.016$
Akaike Inf. Crit.	2,410.033
Note :	*p$<$0.1 ; **p$<$0.05 ; ***p$<$0.01

Le meilleur modèle est alors le 2^{e}, son AIC est de 2410,03 alors que celui du 1^{er} est de 2715,125 :

$$Y = \beta_1 X_2^2 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_1$$

On notera que la variable $X_1 X_2$ ne fait pas partie du modèle choisi.

Méthodologie utilisée

On divise l'échantillon original (dtrain) en 10 échantillons, puis on sélectionne un des 10 échantillons comme ensemble de validation pendant que les 9 autres échantillons constituent l'ensemble d'apprentissage. Puis on répète l'opération en sélectionnant un autre échantillon de validation parmi les blocs prédéfinis. Ce processus d'apprentissage par *Cross-Validation* optimise les paramètres du modèle afin que celui-ci corresponde le mieux possible aux données d'apprentissage.

Listing 8 – Cross-Validation

```
    fitControl <- trainControl(
  ## 10-fold CV
  method = "cv",
  number = 10,
  savePredictions = TRUE
)
```

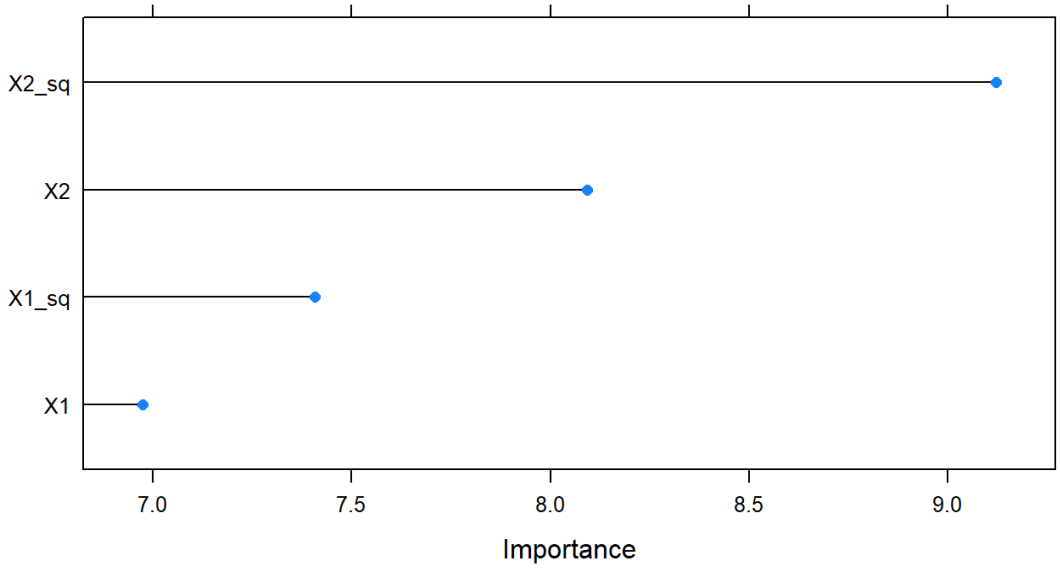
Ainsi pour tous les modèles qu'on va tester ci-dessous, on les entrainera d'abord sur *dtrain*, puis on cherchera l'importance de chaque variable, on fera ensuite une prédiction sur *dtest* et enfin on construira leur matrice de confusion afin de déterminer leur Accuracy, AccuracyLower, Sensitivity et Specificity.

1. **Régression logistique**

Listing 9 – Régression logistique

```
## LR
lreg<-train(classes~X2_sq + X2  + X1 + X1_sq,
            data=dtrain,
            method="glm",
            family=binomial(link = 'logit'),
            trControl=fitControl)

plot(varImp(lreg,scale=F))
```



Listing 10 – Prédiction RL

```
### prediction
lreg_pred<-predict(lreg,dtest)

##resultats
cm = confusionMatrix(lreg_pred,dtest$classes)

lreg_res = data.frame( model = "Logistique",
             Accuracy = cm$overall["Accuracy" ],
            AccuracyLower= cm$overall["AccuracyLower" ],
            AccuracyUpper = cm$overall["AccuracyUpper" ],
            Sensitivity = cm$byClass["Sensitivity"],
            Specificity = cm$byClass["Specificity"], row.names = NULL
            )
lreg_res
```

TABLE 5 – Prédiction RL

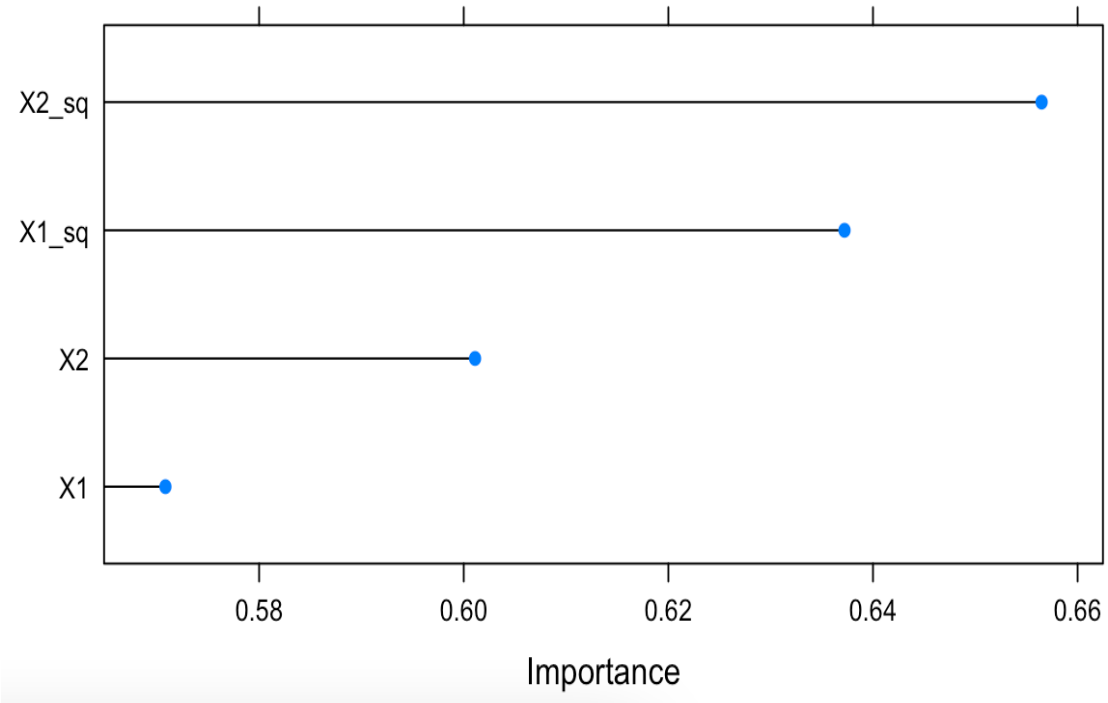
model	Accuracy	AccuracyLower	AccuracyUpper	Sensitivity	Specificity
Logistique	0.6641604	0.6154783	0.7103857	0.8717949	0.369697

Les variables les plus importantes pour le modèle sont X_2^2 et X_2. On a une précision de 66% (+ ou - 5%). La sensibilité de 87,18% est relativement bonne mais une spécificité de seulement 37% laisse à désirer.

2. **Arbre de décision**

Listing 11 – Arbre de décision

```
##DT
dtree<-train(classes~X2_sq + X2  + X1 + X1_sq,
    data=dtrain,
    method="ctree",
    trControl=fitControl)
plot(varImp(dtree,scale=F))
```



Listing 12 – Prédiction AD

```
###prediction
dtree_pred<-predict(dtree,dtest)
##resultats
cm = confusionMatrix(dtree_pred,dtest$classes)

dtree_res = data.frame(model = "Arbre",
                       Accuracy = cm$overall["Accuracy" ],
                      AccuracyLower= cm$overall["AccuracyLower" ],
                      AccuracyUpper = cm$overall["AccuracyUpper" ],
                      Sensitivity = cm$byClass["Sensitivity"],
                      Specificity = cm$byClass["Specificity"], row.↩
                         names = NULL
           )
dtree_res
```

TABLE 6 – Prédiction AD

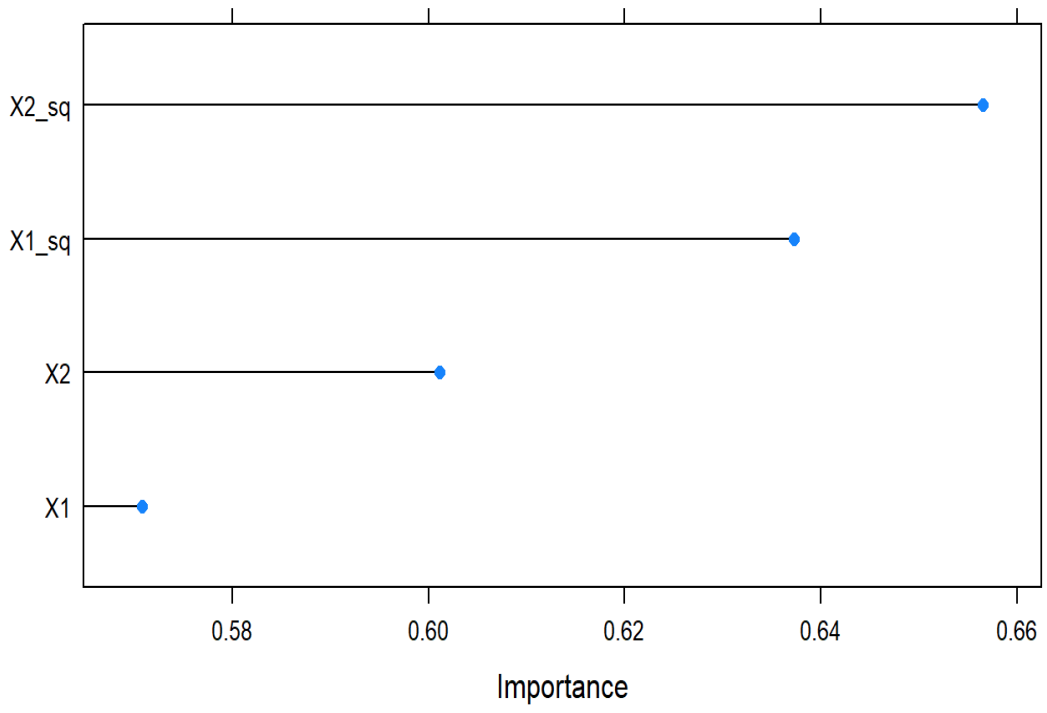
model	Accuracy	AccuracyLower	AccuracyUpper	Sensitivity	Specificity
Arbre	0.7894737	0.7461384	0.8284539	0.8547009	0.6969697

Les variables les plus importantes pour l'arbre de décision sont X_2^2 et X_1^2. On a une précision de 79% (+ ou - 4%). La sensibilité est meilleure et la spécificité l'est beaucoup plus comparée à la régression logistique.

3. **Linear discriminant analysis (LDA)**

Listing 13 – Linear discriminant analysis

```
1     da<-train(classes~X2_sq + X2  + X1 + X1_sq,
2               data=dtrain,
3               method="lda",
4               trControl=fitControl)
5
6
7  plot(varImp(lda,scale=F))
```



Listing 14 – Prédiction LDA

```
### predict on test dataset
lda_pred<-predict(lda,dtest)


##results
cm = confusionMatrix(lda_pred,dtest$classes)

lda_res = data.frame(model = "LDA",
                         Accuracy = cm$overall["Accuracy" ],
                       AccuracyLower= cm$overall["AccuracyLower" ],
                       AccuracyUpper = cm$overall["AccuracyUpper" ],
                       Sensitivity = cm$byClass["Sensitivity"],
                       Specificity = cm$byClass["Specificity"], row.↩
                          names = NULL
            )
lda_res
```

TABLE 7 – Prédiction LDA

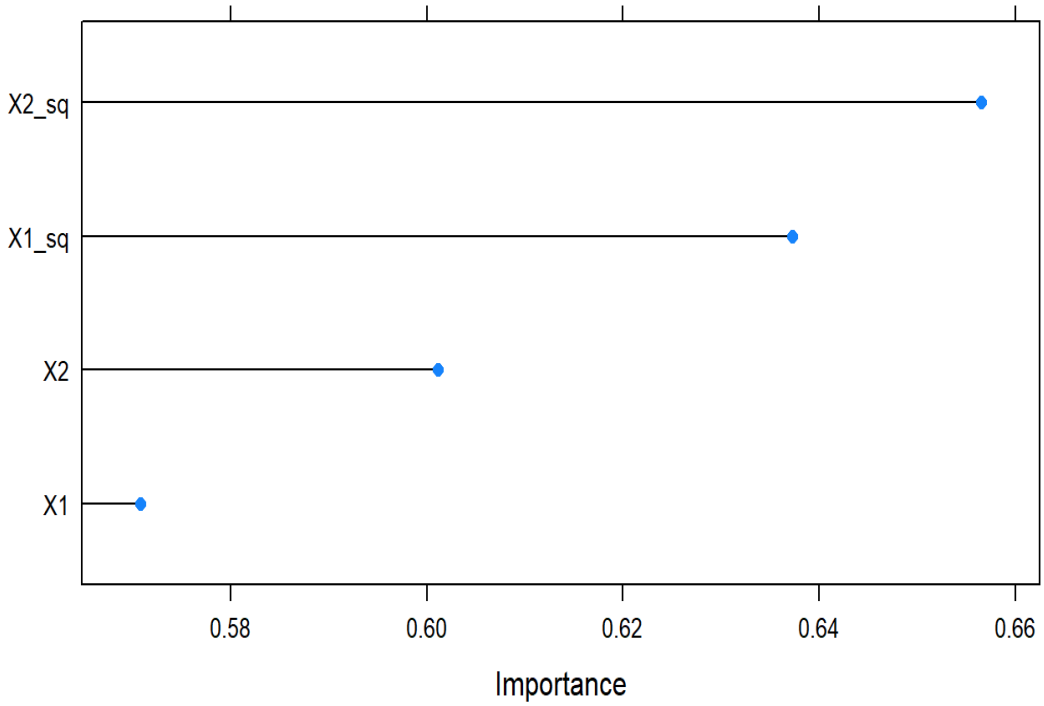
model	Accuracy	AccuracyLower	AccuracyUpper	Sensitivity	Specificity
LDA	0.6716792	0.623202	0.7175866	0.9059829	0.3393939

Les variables les plus importantes pour l'analyse discriminante linéaire sont comme pour l'arbre de décision, X_2^2 et X_1^2. On a une précision proche de la régression logistique. La sensibilité est de 90% mais la spécificité est très faible, on beaucoup de faux positifs.

4. **Quadratic discriminant analysis (QDA)**

Listing 15 – Quadratic discriminant analysis

```
    qda<-train(classes~X2_sq + X2  + X1 + X1_sq,
            data=dtrain,
            method="qda",
            trControl=fitControl)
plot(varImp(qda,scale=F))
```



Listing 16 – Prédiction QDA

```
### predict on test dataset
qda_pred<-predict(qda,dtest)

##results
cm = confusionMatrix(qda_pred,dtest$classes)

qda_res = data.frame(model = "QDA",
                       Accuracy = cm$overall["Accuracy" ],
                     AccuracyLower= cm$overall["AccuracyLower" ],
                     AccuracyUpper = cm$overall["AccuracyUpper" ],
                     Sensitivity = cm$byClass["Sensitivity"],
                     Specificity = cm$byClass["Specificity"], row.↩
                        names = NULL
           )
qda_res
```

TABLE 8 – Prédiction QDA

model	Accuracy	AccuracyLower	AccuracyUpper	Sensitivity	Specificity
QDA	0.6817043	0.6335208	0.7271671	0.9273504	0.3333333

Les résultats de l'analyse discriminante quadratique sont similaires avec ceux de l'analyse discriminante linéaire.

5. **Comparaison des modèles**

Listing 17 – Comparaison des modèles

```
    comparaison = rbind(lreg_res,
                        dtree_res,
                        lda_res,
                        qda_res)
rownames(comparaison) <- comparaison$model
comparaison
```

TABLE 9 – Récapitulatif

model	Accuracy	AccuracyLower	AccuracyUpper	Sensitivity	Specificity
Logistique	0.6641604	0.6154783	0.7103857	0.8717949	0.369697
Arbre	0.7894737	0.7461384	0.8284539	0.8547009	0.6969697
LDA	0.6716792	0.623202	0.7175866	0.9059829	0.3393939
QDA	0.6817043	0.6335208	0.7271671	0.9273504	0.3333333

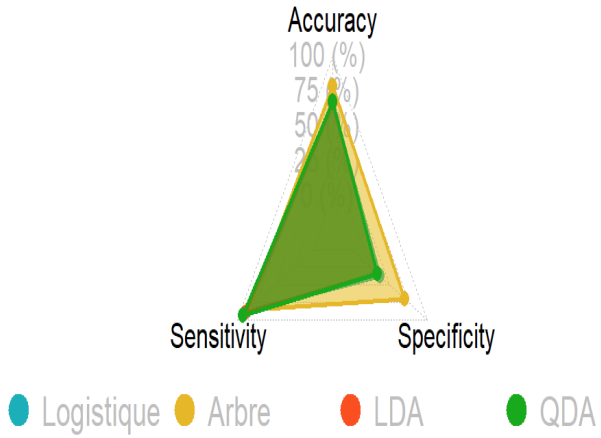
À travers ce tableau on remarque que concernant la *précision* l'arbre de décision a un meilleur score, du point de vue de la *sensibilité* le QDA est devant suivi du LDA mais concernant la capacité d'un test à donner un résultat négatif lorsque l'hypothèse n'est pas vérifiée à savoir la *spécificité* l'arbre de décision est meilleur.
Pour mieux voir les différences, on va faire un radar map.

Listing 18 – Radar map

```
#plot a radar map!
data <- rbind(rep(1,5) , rep(0,5) , comparaison[,c(2,5,6)])

color = c("#00AFBB", "#E7B800", "#FC4E07","#00AC00")
radarchart(
    data, axistype = 1,
    # Personnaliser le polygone
    pcol = color, pfcol = scales::alpha(color, 0.5), plwd = 2, plty =↩
        1,
    # Personnaliser la grille
    cglcol = "grey", cglty = 3, cglwd = 0.2,
    # Personnaliser l'axe
    axislabcol = "grey",
    # Etiquettes des variables
     vlabels = colnames(data))
legend(x="bottom", legend = rownames(data[-c(1,2),]), col =color, ↩
```

```
        horiz = T,
16          bty = "n", pch=20 , text.col = "grey", cex=1.2, pt.cex=3)
```



Dans tous les modèles testés, Y=1 constitue la classe 1 et Y=2 constitue la classe 0. Le tableau présente les métriques des 4 modèles testés :

- AccuracyLower et AccuracyUpper sont les bornes de l'intervalle de confiance de la métrique Accuracy, qui est le taux de précision des prédictions ;
- On voit que le triangle constitué par les métriques de l'arbre de décision est le plus à l'extérieur, sauf pour le sommet attribué à la sensibilité où le QDA est le meilleur ;
- On peut en déduire que globalement l'arbre est le meilleur modèle pour modéliser les probabilité.

QUESTION 2 - Prédictions à partir du fichier xsimutest

Listing 19 – Nouvelles données

```
1 #entrainer le modele sur toute la donnee
2 dtree<-train(classes~X2_sq + X2  + X1 + X1_sq,
3              data=simu,
4              method="ctree",
5              trControl=fitControl)
6 # creer les variables sur les nouvelles donnees
7 xsimutest = create_feature(xsimutest)
```

Listing 20 – Prédiction

```
1 ### predict on test dataset
```

```
2 xsimutest$predicted  = predict(dtree,xsimutest)
3 xsimutest$probability_of_class1  = predict(dtree,
4                     xsimutest, type="prob")[,1]
5 write.table(xsimutest[,c(1,2,6,7)], paste0(path, "/donnees/prediction.txt"↩
    ), row.names = F, col.names = F)
```

Le fichier stockant la prédiction de 1000 observations de la variable Y se nomme ***prediction***.

EXERCICE 2 - RACES DE CHIENS

Statistiques descriptives

Listing 21 – Exploration des données

```
df_status(chiens)
```

TABLE 10 – Exploration des données

variable	q-zeros	p-zeros	q-na	p-na	q-inf	p-inf	unique
TAI	0	0	0	0	0	0	3
POI	0	0	0	0	0	0	3
VEL	0	0	0	0	0	0	3
INT	0	0	0	0	0	0	3
AFF	0	0	0	0	0	0	2
AGR	0	0	0	0	0	0	2
FON	0	0	0	0	0	0	3

Les données communiquées décrivent les caractéristiques de 27 races de chiens au moyen de variables qualitatives, les 6 premières ont été considérées comme actives, la septième *fonction*, comme supplémentaire : ses trois modalités sont « compagnie », « chasse » et « garde» ».
La base de données ne contient aucune valeur manquante, aucune observation n'est nulle et toutes les variables sont de type *integer*.
On remarquera que les paires d'individus suivant : (buldog, teckel), (chihuahu, pekinois) et (dalmatie, labrador) ont des valeurs identiques pour les 7 variables, il y aura donc des observations confondues.

QUESTION 1 - Analyse des correspondances multiples des données

Nous allons transformer les types en *facteur* et rendre les modalités identifiables afin de pas les confondre.

Listing 22 – ACM avec FON en var supp

```
 for (var in colnames(chiens)) {
   chiens[,var] <-  as.factor(chiens[,var])
   chiens[,var] <- paste(var,chiens[,var],sep = "_")
 }
grp <- as.factor(chiens[, "FON"])
mca <- MCA (chiens,  quali.sup = c(7), graph = FALSE)
```

```
7 summary(mca, nbelements = 100 ,nb.dec = 2, ncp = 2)
```

On affiche les valeurs propres :

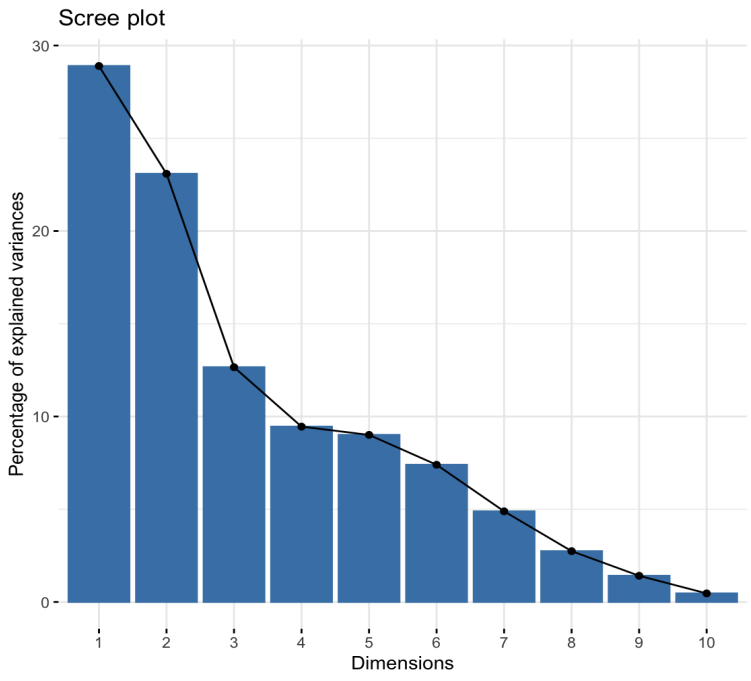
Listing 23 – Valeurs propres

```
1 round(mca$eig,3)
```

	eigenvalue	percentage.of.variance	cumulative.percentage.of.variance
dim 1	0.48	28.90	28.90
dim 2	0.38	23.08	51.98
dim 3	0.21	12.66	64.64
dim 4	0.16	9.45	74.09
dim 5	0.15	9.01	83.10
dim 6	0.12	7.40	90.50
dim 7	0.08	4.89	95.38
dim 8	0.05	2.74	98.12
dim 9	0.02	1.41	99.54
dim 10	0.01	0.46	100.00

Listing 24 – Gains d'inertie

```
1 fviz_eig(mca, barcolor = "firebrick4", barfill="firebrick", title="Gain d'↩
    inertie)
```



Le diagramme des valeurs propres montre cependant une chute après μ_2. On interprètera donc uniquement les deux premiers axes. L'inertie projetée cumulée nous indique que les deux premiers axes expliquent à eux seuls 51.98% (28.90% + 23.08%) des variations observées dans notre échantillon.

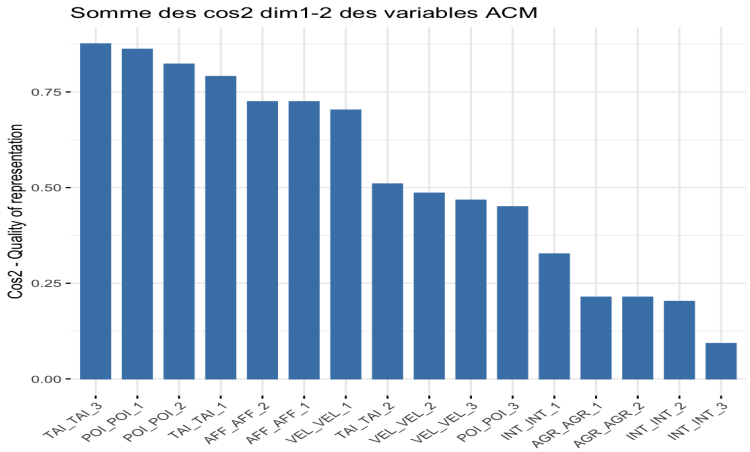
Listing 25 – cos2 des variables

```
mca$var$cos2
cosvar=mca$var$cos2
cosvar=data.frame(cosvar)
print(xtable(cosvar, type ="latex"), file = "cosvar1.tex")
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
TAI_1	0.49	0.30	0.13	0.01	0.00
TAI_2	0.16	0.34	0.23	0.03	0.02
TAI_3	0.88	0.00	0.00	0.04	0.02
POI_1	0.58	0.29	0.05	0.01	0.00
POI_2	0.10	0.72	0.06	0.02	0.04
POI_3	0.23	0.22	0.34	0.00	0.09
VEL_1	0.06	0.64	0.09	0.00	0.06
VEL_2	0.15	0.33	0.05	0.06	0.06
VEL_3	0.40	0.07	0.29	0.03	0.00
INT_1	0.05	0.28	0.05	0.00	0.45
INT_2	0.13	0.08	0.23	0.34	0.02
INT_3	0.03	0.06	0.10	0.46	0.32
AFF_1	0.65	0.08	0.00	0.01	0.00
AFF_2	0.65	0.08	0.00	0.01	0.00
AGR_1	0.17	0.04	0.10	0.28	0.13
AGR_2	0.17	0.04	0.10	0.28	0.13

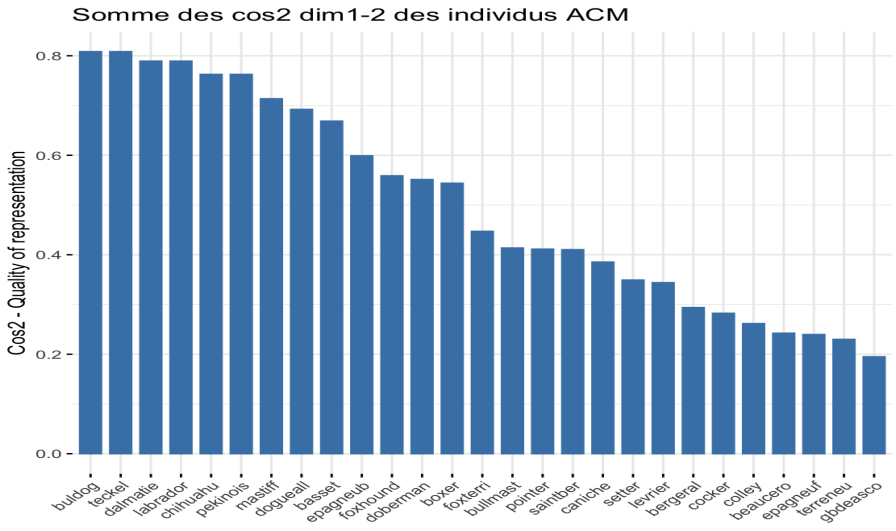
Listing 26 – Graphique de la somme des Cos2 dim1-2 des variables ACM

```
fviz_cos2(mca, choice = "var", title="Somme des cos2 dim1-2 des vari ACM",↩
    axes = 1:2)
```



Listing 27 – Graphique de la somme des Cos2 dim1-2 des individus ACM

```
fviz_cos2(mca, choice = "ind", title="Somme des cos2 dim1-2 des individus ACM")
```



Listing 28 – cos2 des individus ACM

```
mca$ind$cos2
cosind=mca$ind$cos2
cosind=data.frame(cosind)
print(xtable(cosind, type ="latex"),
                        file = "cosind1.tex")
```

Au regard des cosinus présentés dans ci-dessus, on remarque que pour toutes les races de chiens cités ci-après la somme des cos^2 est inférieure à 0.5 ; il s'agit de gbdeasco, terreneu, epagneul, beaucero, colley, cocker, bergeral, levrier, setter, caniche, sainber, pointer, bullmast et foxtern. Donc elles sont mal représentées sur ce plan, nous ne pouvons rien dire sur leurs

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
beaucero	0.09	0.15	0.01	0.04	0.01
basset	0.03	0.63	0.02	0.04	0.14
bergeral	0.15	0.14	0.16	0.22	0.05
boxer	0.11	0.43	0.27	0.04	0.12
buldog	0.62	0.18	0.02	0.07	0.07
bullmast	0.27	0.14	0.12	0.21	0.25
caniche	0.39	0.00	0.15	0.18	0.09
chihuahu	0.38	0.38	0.12	0.00	0.02
cocker	0.28	0.00	0.23	0.02	0.01
colley	0.01	0.25	0.10	0.39	0.03
dalmatie	0.24	0.55	0.12	0.02	0.01
doberman	0.49	0.06	0.13	0.17	0.04
dogueall	0.56	0.13	0.01	0.00	0.05
epagneub	0.10	0.49	0.00	0.17	0.03
epagneuf	0.02	0.22	0.01	0.18	0.00
foxhound	0.56	0.00	0.10	0.00	0.32
foxterri	0.44	0.01	0.00	0.05	0.04
gbdeasco	0.19	0.01	0.00	0.04	0.46
labrador	0.24	0.55	0.12	0.02	0.01
levrier	0.34	0.01	0.26	0.16	0.09
mastiff	0.30	0.41	0.18	0.01	0.02
pekinois	0.38	0.38	0.12	0.00	0.02
pointer	0.29	0.12	0.31	0.00	0.20
saintber	0.20	0.21	0.47	0.01	0.06
setter	0.22	0.13	0.07	0.46	0.02
teckel	0.62	0.18	0.02	0.07	0.07
terreneu	0.09	0.14	0.26	0.20	0.24

proximités avec les autres modalités. Il en est de même pour les catégories INT-3,INT-2, AGR-2, AGR-1, INT-1, POI-3, VEL-3 et VEL-2 la somme de leurs $cos^2 < 0.5$.

Regardons les contributions :

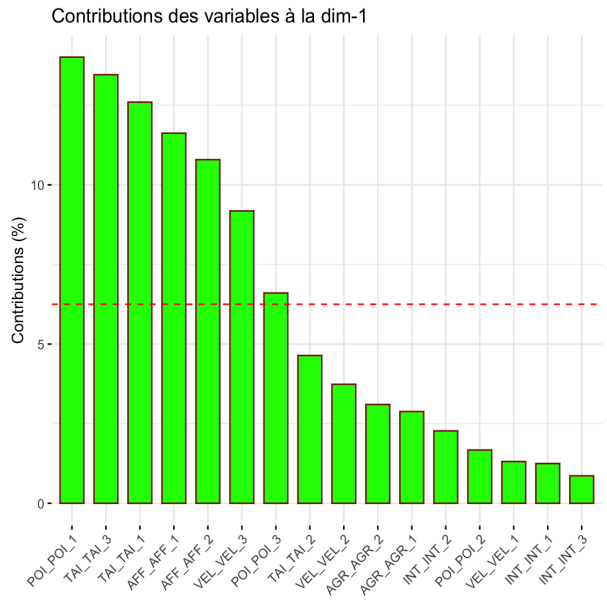
Listing 29 – Contributions des variables ACM

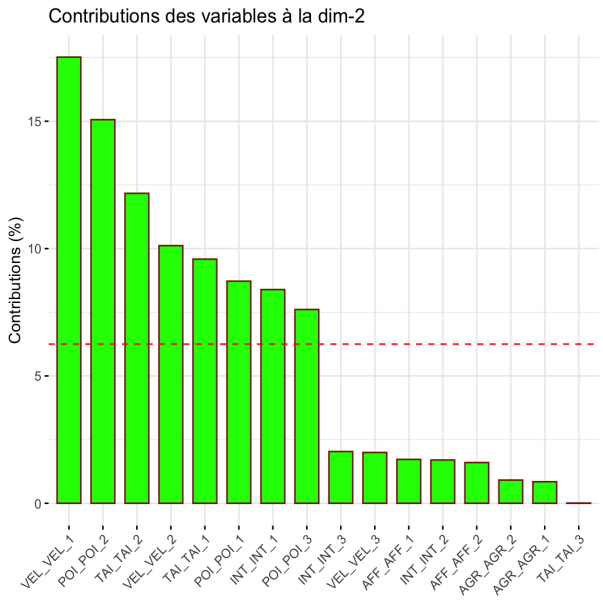
```
1 varcont=mca$var$contrib
2 varcont=data.frame(varcont)
3 print(xtable(varcont, type ="latex"), file = "varcont1.tex")
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
TAI_1	12.60	9.59	7.77	0.40	0.01
TAI_2	4.64	12.17	15.10	2.30	1.98
TAI_3	13.46	0.01	0.12	1.70	0.78
POI_1	14.01	8.72	3.01	0.85	0.09
POI_2	1.67	15.06	2.19	0.77	2.08
POI_3	6.60	7.61	21.83	0.09	7.76
VEL_1	1.31	17.52	4.72	0.25	3.85
VEL_2	3.74	10.12	2.97	4.30	4.53
VEL_3	9.18	2.00	15.34	2.03	0.00
INT_1	1.25	8.39	2.89	0.02	35.24
INT_2	2.27	1.70	9.25	18.55	1.14
INT_3	0.86	2.03	6.32	38.23	27.89
AFF_1	11.62	1.72	0.16	0.35	0.10
AFF_2	10.79	1.60	0.15	0.33	0.09
AGR_1	2.88	0.85	3.93	14.37	6.96
AGR_2	3.10	0.91	4.23	15.47	7.50

Listing 30 – Graphique des contributions des variables ACM

```
1 fviz_contrib(mca, choice = "var", axes = 1, title="Contributions des ↩
    variables    la dim-1",color = "firebrick4", fill = "green")
2
3 fviz_contrib(mca, choice = "var", axes = 2, title="Contributions des ↩
    variables    la dim-2",color = "firebrick4", fill = "green")
```





On sait que les variables avec les plus grandes valeurs, contribuent le mieux à la définition des dimensions. Les catégories qui contribuent le plus à Dim.1 et Dim.2 sont les plus importantes pour expliquer la variabilité dans le jeu de données. La ligne en pointillé rouge, sur le graphique

ci-dessus, indique la valeur moyenne attendue sous l'hypothèse nulle.

— les variables POL-1, TAL-3 et TAL-1 contribuent le plus à la définition de la 1^{re} dimension ;

— les variables VEL-1, POL-2 sont les importantes concenrnant la $2e$ dimension.

Listing 31 – Contributions des individus ACM

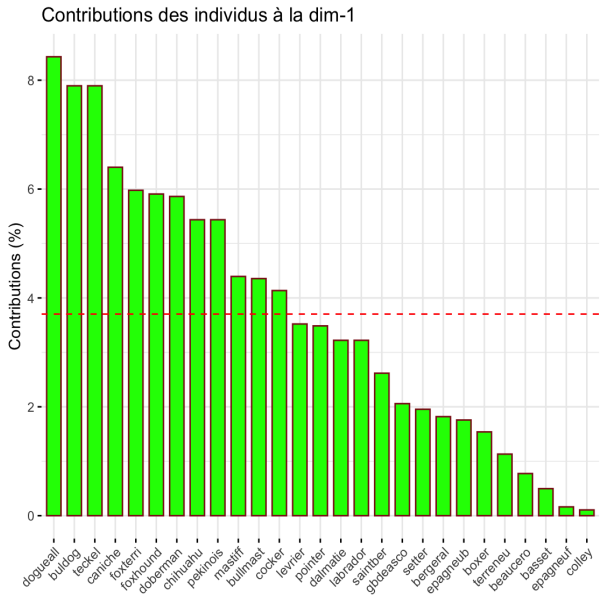
```
mca$ind$contrib
indcont=mca$ind$contrib
indcont=data.frame(indcont)
print(xtable(indcont, type ="latex"), file = "indcont1.tex")
```

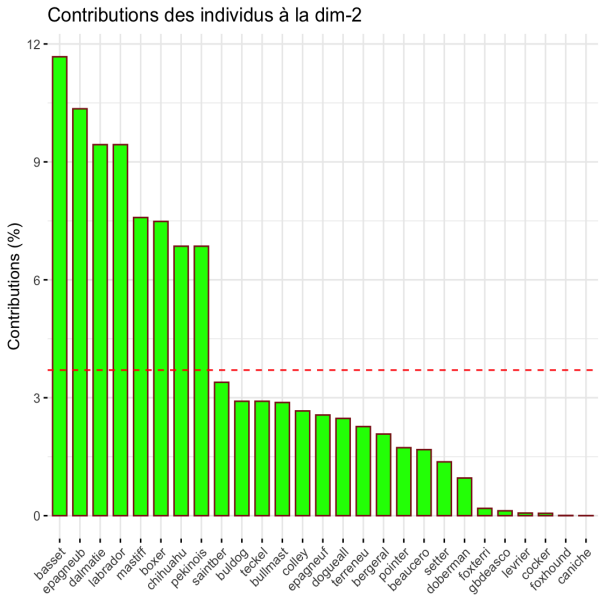
	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
beaucero	0.77	1.68	0.18	1.05	0.35
basset	0.50	11.67	0.64	2.01	6.77
bergeral	1.82	2.08	4.36	7.84	1.88
boxer	1.54	7.48	8.41	1.59	5.12
buldog	7.90	2.91	0.47	2.88	2.70
bullmast	4.36	2.88	4.35	10.09	12.86
caniche	6.40	0.00	5.84	9.27	4.65
chihuahu	5.44	6.85	3.88	0.18	0.78
cocker	4.14	0.06	7.70	0.85	0.27
colley	0.11	2.66	1.97	10.17	0.91
dalmatie	3.22	9.44	3.69	0.82	0.52
doberman	5.86	0.96	3.59	6.12	1.42
dogueall	8.43	2.47	0.48	0.09	2.47
epagneub	1.76	10.35	0.07	8.53	1.54
epagneuf	0.16	2.56	0.24	5.17	0.00
foxhound	5.91	0.01	2.30	0.01	10.83
foxterri	5.98	0.19	0.05	1.92	1.81
gbdeasco	2.06	0.12	0.03	1.37	16.50
labrador	3.22	9.44	3.69	0.82	0.52
levrier	3.52	0.07	6.23	5.01	3.06
mastiff	4.39	7.58	6.06	0.40	0.80
pekinois	5.44	6.85	3.88	0.18	0.78
pointer	3.49	1.73	8.26	0.10	7.48
saintber	2.62	3.39	14.04	0.42	2.65
setter	1.95	1.37	1.47	12.36	0.60
teckel	7.90	2.91	0.47	2.88	2.70
terreneu	1.13	2.27	7.67	7.91	10.05

Listing 32 – Graphique des contributions des individus ACM

```
fviz_contrib(mca, choice = "ind", axes = 1, title="Contributions des ↩
    individus    la dim-1",color = "firebrick4", fill = "green")

fviz_contrib(mca, choice = "ind", axes = 2, title="Contributions des ↩
    individus    la dim-2",color = "firebrick4", fill = "green")
```

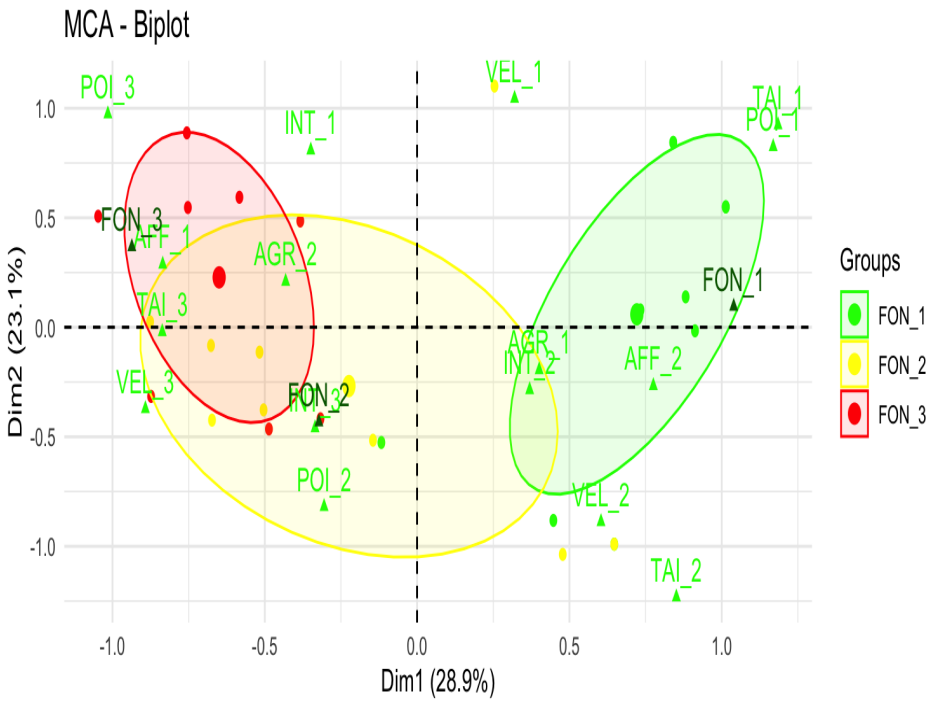




Listing 33 – MCA - Biplot

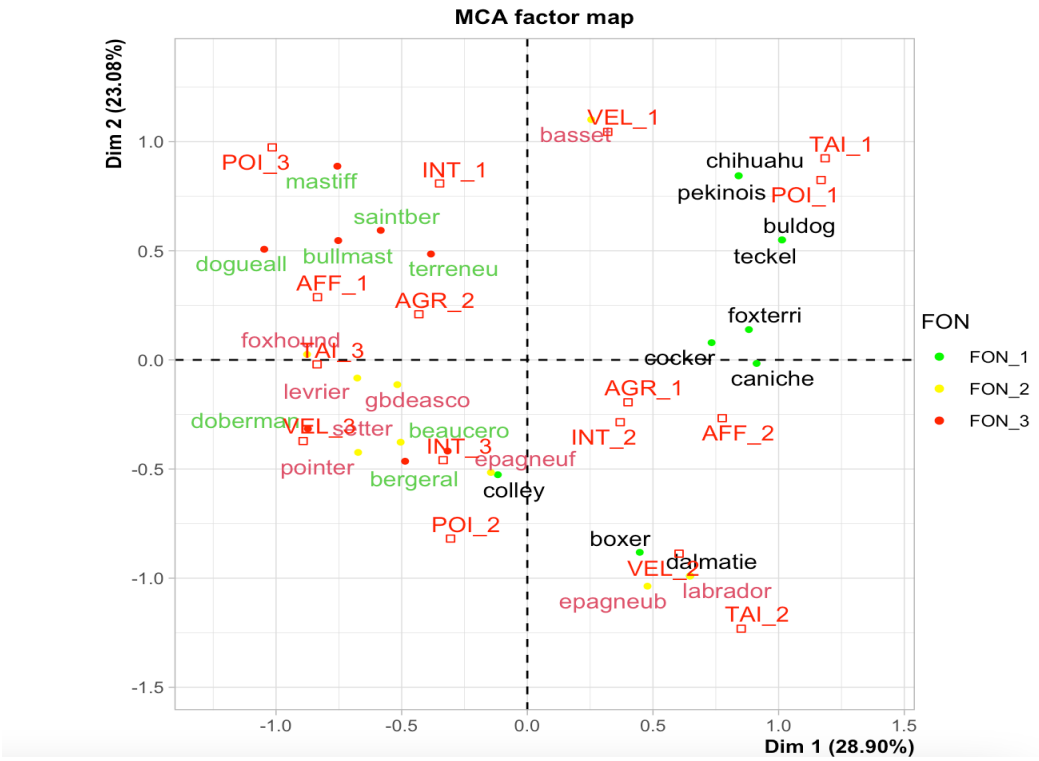
```
g}[label={list:first},caption= MCA - Biplot]
p2 <- fviz_mca_biplot(mca, label="var", col.var ="green",col.ind="white",
            habillage=grp,palette = c("green","yellow", "red"),
            addEllipses=TRUE, ellipse.level=0.5) +
  theme_minimal()

p2
```



Listing 34 – MCA factor map

```
plot(mca, habillage="FON",palette = c("green","yellow", "red"))
```



Le 1^{er} graphique nous donne une description des chiens selon leurs fonctions, le 2^{nd} graphique incorpore en plus du 1^{er} les races des chiens. Ainsi l'interprétation de l'analyse en correspondance multiple devient relativement simple. Lorsque deux modalités sont proches, on dira que

les individus qui sont caractérisés par l'une sont également caractérisés par l'autre en général (à condition que la somme de ses cos^2 >0.5).
On peut alors dire que :

1. Les buldog, caniche,boxers, dalmatiens, chihuahua, foxterri, pekinois et teckel sont les chiens de compagnie (FON-1), ils sont très afffectueux, lents, de petites tailles, légers. Les boxers et les dalmatiens sont aussi des chiens de compagnie mais sont plus rapides et plus grands que les autres chiens de compagnie ;
2. les basset, foxhound, labrador, pointer, labrador et epagneub sont les chiens de chasse (FON-2), ils sont lourds, peu affectueux, très intelligents, très aggressifs, très grands de taille ; les labradors et epagneub sont toutefois moins rapides et moins grands, les basset sont moins rapides ;
3. les doberman, dogueall, mastiff et saintber sont les chiens de garde (FON-3), ils sont peu intelligents, et peu affectueux et sont très lourds. Toutefois les doberman sont très intelligents et plus aggressifs,
4. On peut globalement dire que les chiens de chasse sont les plus intelligents et plus grands, les chiens de compagnie sont les plus affectueux et plus petits, les chiens de garde sont plus aggressifs,et plus rapides mais moins intelligents. En effet L'axe 1 oppose (à droite) les chiens de petite taille, affectueux, qui coïncident avec les chiens de compagnie , aux chiens de grande taille, très rapides et agressifs qui sont les chiens de garde. L'axe 2 oppose (en bas) les chiens de chasse, de taille moyenne, très intelligents à des chiens lents et peu intelligents.

Le but étant d'augmenter l'inertie, les deux premiers axes expliquent à eux seuls que 51,98% des variations observées dans notre échantillon.Pour avoir des résultats plus précis on pourrait construire les axes qu'avec les variables les plus corrélées au sens de KHI-2 et mettre le reste en supplémentaire. On va visualiser la corrélation entre les variables et les axes principaux de l'ACM pour déterminer les variables à augmenter comme variables supplémentaires.

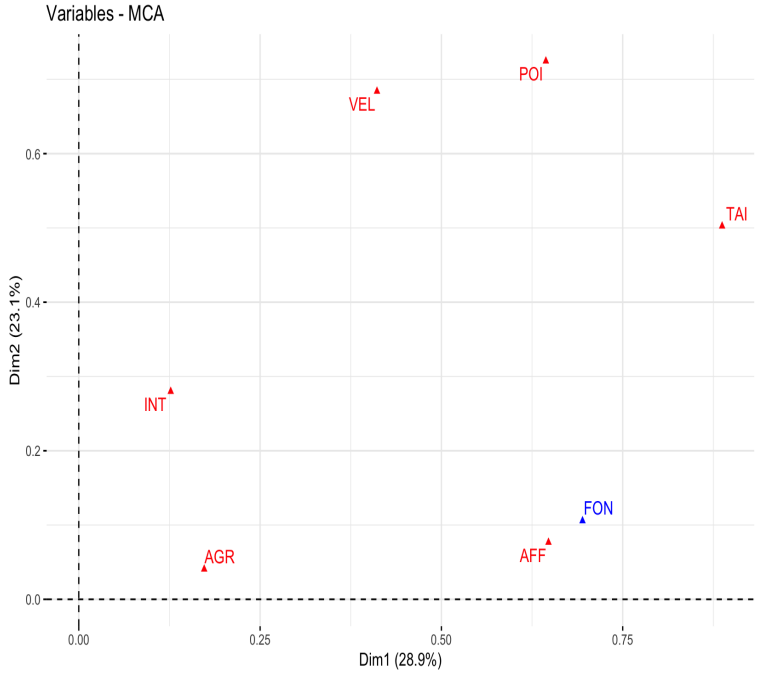
Listing 35 – Coordonnées des variables

```
varcont=mca$var$contrib
varcont=data.frame(varcont)
print(xtable(varcont, type ="latex"), file = "varcont.tex")
```

Listing 36 – Corrélation entre les variables et les axes principaux

```
fviz_mca_var (mca, choice = "mca.cor",
              repel = TRUE,
              ggtheme = theme_minimal ())
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
TAI_1	1.36	0.65	-0.59	-0.41	0.09
TAI_2	0.62	-1.42	1.12	0.66	-0.44
TAI_3	-0.84	0.17	-0.10	-0.03	0.11
POI_1	1.33	0.55	-0.41	-0.08	-0.15
POI_2	-0.47	-0.77	-0.22	-0.01	-0.15
POI_3	-0.82	1.27	1.27	0.16	0.65
VEL_1	0.50	1.02	0.37	0.34	-0.29
VEL_2	0.45	-1.05	0.54	-0.85	0.24
VEL_3	-0.95	-0.20	-0.90	0.38	0.11
AFF_1	-0.77	0.43	0.13	-0.39	-0.35
AFF_2	0.71	-0.40	-0.12	0.36	0.32



Ainsi on voit que INT et AGR ont des coordonnées très proches de 0 sur les 2 axes, on va alors les ajouter en variables supplémentaires dans notre matrice d'inestie. On appelera cette méthode ACM augmentée.

Listing 37 – ACM augmentée

```
 for (var in colnames(chiens)) {
   chiens[,var] <-  as.factor(chiens[,var])
   chiens[,var] <- paste(var,chiens[,var],sep = "_")
 }
grp <- as.factor(chiens[, "FON"])
d <- which(!colnames(chiens)%in%c("TAI","VEL","POI", "AFF"))
mca <- MCA (chiens,  quali.sup = d, graph = FALSE)
summary(mca, nbelements = 100 ,nb.dec = 2, ncp = 2)
```

On affiche les valeurs propres :

Listing 38 – Valeurs propres

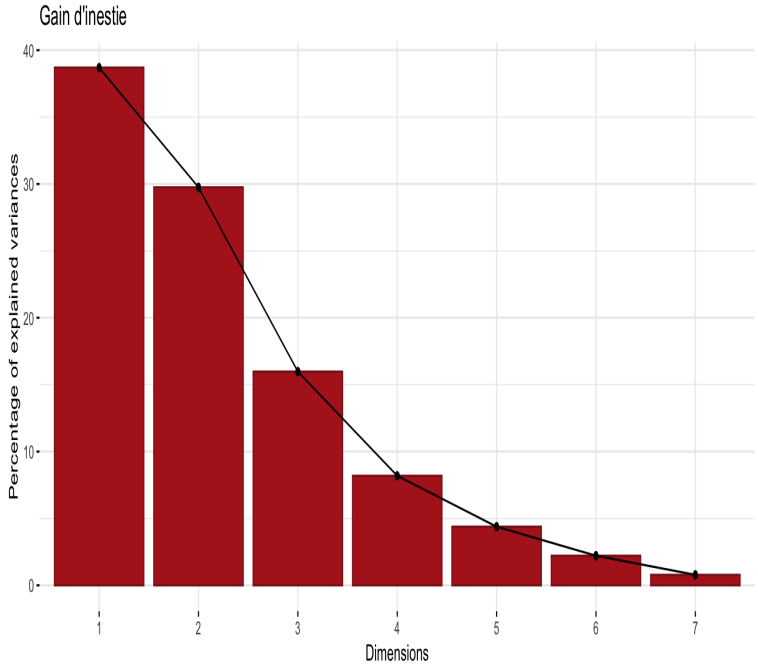
```
round(mca$eig,3)
```

TABLE 11 – Valeurs propres

dimension	eigenvalue	percentage of variance	cumulative percentage of variance
dim1	0.677	38.701	38.701
dim2	0.521	29.751	68.452
dim3	0.280	15.977	84.429
dim4	0.143	8.189	92.618
dim5	0.077	4.385	97.003
dim6	0.039	2.213	99.216
dim7	0.014	0.784	100.000

Listing 39 – Gains d'inertie

```
fviz_eig(mca, barcolor = "firebrick4", barfill="firebrick", title="Gain d'inertie)
```



Le diagramme des valeurs propres montre cependant une chute après μ_2. On interprètera donc uniquement les deux premiers axes. L'inertie projetée cumulée nous indique que les deux premiers axes expliquent à eux seuls 68,5% (38,7% + 29,8%) des variations observées dans notre

échantillon.

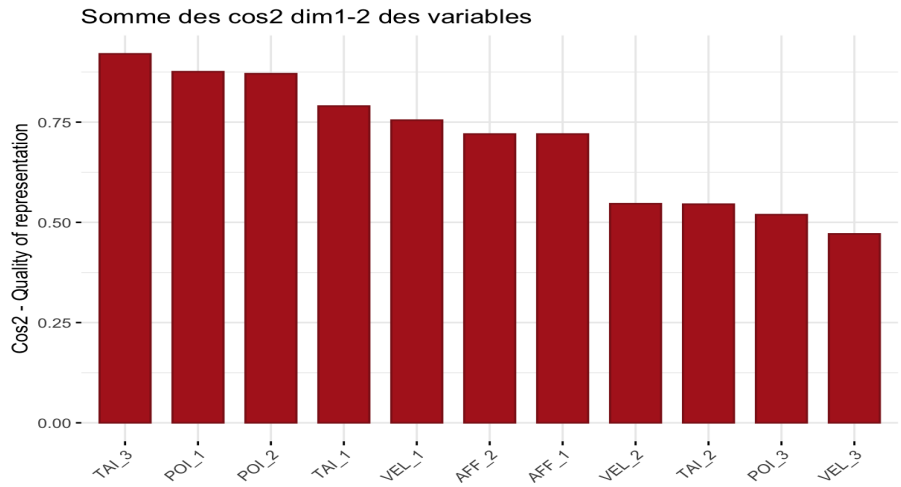
Listing 40 – cos2 des variables

```
mca$var$cos2
cosvar=mca$var$cos2
cosvar=data.frame(cosvar)
print(xtable(cosvar, type ="latex"), file = "cosvar.tex")
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
TAI_1	0.64	0.15	0.12	0.06	0.00
TAI_2	0.09	0.46	0.28	0.10	0.04
TAI_3	0.88	0.04	0.01	0.00	0.01
POI_1	0.75	0.13	0.07	0.00	0.01
POI_2	0.24	0.63	0.05	0.00	0.02
POI_3	0.15	0.37	0.36	0.01	0.10
VEL_1	0.14	0.61	0.08	0.07	0.05
VEL_2	0.08	0.46	0.12	0.30	0.02
VEL_3	0.45	0.02	0.40	0.07	0.01
AFF_1	0.55	0.17	0.02	0.14	0.11
AFF_2	0.55	0.17	0.02	0.14	0.11

Listing 41 – Graphique de la somme des Cos2 dim1-2 des variables

```
fviz_cos2(mca, choice = "var", title="Somme des cos2 dim1-2 des variables"↩
    , axes = 1:2,color = "firebrick4", fill = "firebrick")
```



Listing 42 – cos2 des individus

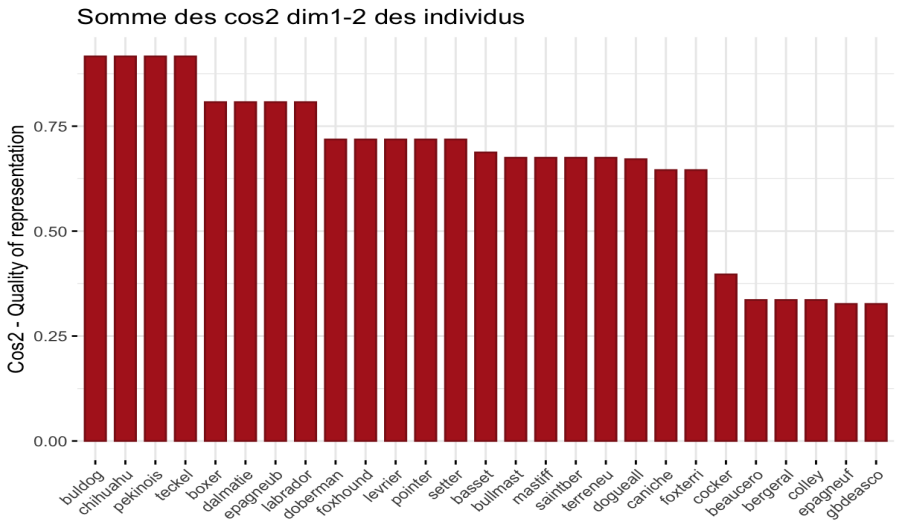
```
mca$ind$cos2
cosind=mca$ind$cos2
cosind=data.frame(cosind)

print(xtable(cosind, type ="latex"),
                        file = "cosind.tex")
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
beaucero	0.19	0.15	0.34	0.18	0.11
basset	0.27	0.42	0.03	0.06	0.20
bergeral	0.19	0.15	0.34	0.18	0.11
boxer	0.07	0.73	0.18	0.00	0.00
buldog	0.71	0.20	0.06	0.01	0.00
bullmast	0.17	0.50	0.31	0.00	0.01
caniche	0.64	0.00	0.03	0.19	0.10
chihuahu	0.71	0.20	0.06	0.01	0.00
cocker	0.39	0.00	0.09	0.30	0.11
colley	0.19	0.15	0.34	0.18	0.11
dalmatie	0.07	0.73	0.18	0.00	0.00
doberman	0.70	0.01	0.22	0.00	0.05
dogueall	0.51	0.16	0.02	0.00	0.11
epagneub	0.07	0.73	0.18	0.00	0.00
epagneuf	0.19	0.14	0.02	0.55	0.01
foxhound	0.70	0.01	0.22	0.00	0.05
foxterri	0.64	0.00	0.03	0.19	0.10
gbdeasco	0.19	0.14	0.02	0.55	0.01
labrador	0.07	0.73	0.18	0.00	0.00
levrier	0.70	0.01	0.22	0.00	0.05
mastiff	0.17	0.50	0.31	0.00	0.01
pekinois	0.71	0.20	0.06	0.01	0.00
pointer	0.70	0.01	0.22	0.00	0.05
saintber	0.17	0.50	0.31	0.00	0.01
setter	0.70	0.01	0.22	0.00	0.05
teckel	0.71	0.20	0.06	0.01	0.00
terreneu	0.17	0.50	0.31	0.00	0.01

Listing 43 – Graphique de la somme des Cos2 dim1-2 des individus

```
fviz_cos2(mca, choice = "ind", title="Somme des cos2 dim1-2 des individus"↩
    , axes = 1:2, color = "firebrick4", fill = "firebrick")
```



Au regard des cosinus présentés dans ci-dessus, on remarque que pour toutes les races de chiens cités ci-après la somme des cos^2 est inférieure à 0.5 ; il s'agit de beaucero (0.34), bergeral (0.34), cocker (0.39), colley (0.39), epargneuf (0.33) et gbdeasco (0.33). Donc elles sont mal représentées sur ce plan, nous ne pouvons rien dire sur leurs proximités avec les autres modalités. Il en est de même pour la catégorie VEL-3, la somme de ses $cos^2 = 0,47 < 0.5$.

Regardons les contributions :

Listing 44 – Contributions des variables

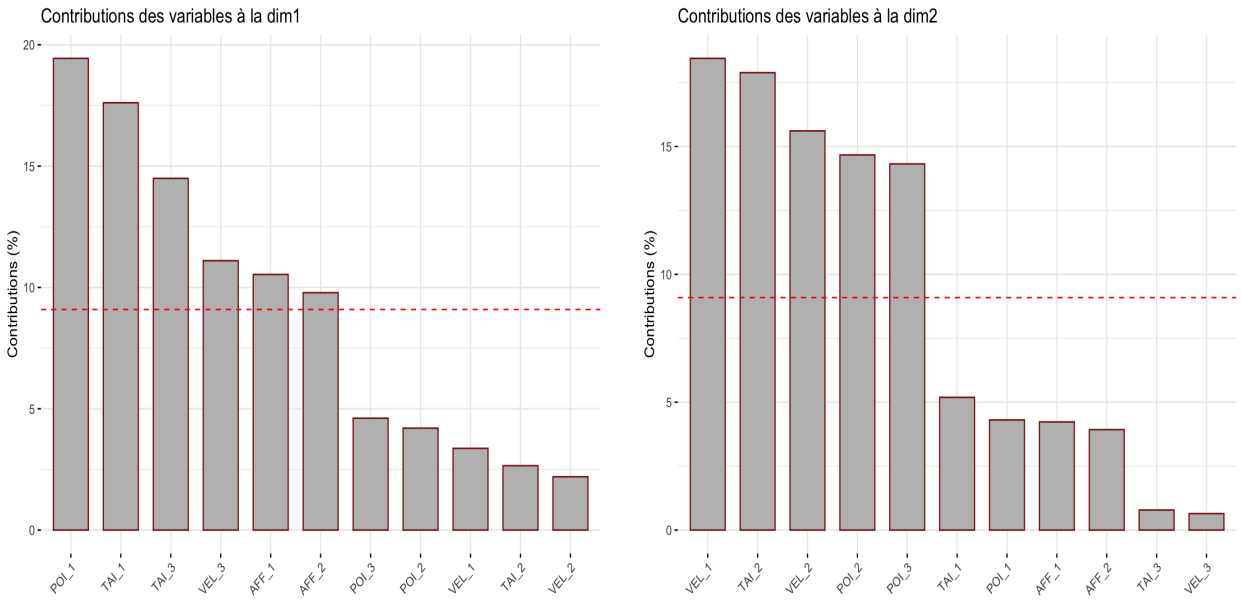
```
varcont=mca$var$contrib
varcont=data.frame(varcont)
print(xtable(varcont, type ="latex"), file = "varcont.tex")
```

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
TAI_1	17.61	5.19	8.04	7.47	0.70
TAI_2	2.65	17.89	20.69	14.05	11.94
TAI_3	14.50	0.78	0.47	0.09	2.03
POI_1	19.44	4.31	4.40	0.30	2.18
POI_2	4.20	14.67	2.22	0.02	3.68
POI_3	4.61	14.32	26.50	0.85	25.76
VEL_1	3.37	18.44	4.62	7.38	10.26
VEL_2	2.20	15.61	7.76	37.37	5.55
VEL_3	11.10	0.64	23.93	8.41	1.33
AFF_1	10.54	4.23	0.70	12.48	18.96
AFF_2	9.78	3.93	0.65	11.59	17.60

Listing 45 – Graphique des contributions des variables

```
fviz_contrib(mca, choice = "var", axes = 1, title="Contributions des variables la dim-1",color = "firebrick4", fill = "grey")

fviz_contrib(mca, choice = "var", axes = 2, title="Contributions des variables la dim-2",color = "firebrick4", fill = "grey")
```



On sait que les variables avec les plus grandes valeurs, contribuent le mieux à la définition des dimensions. Les catégories qui contribuent le plus à Dim.1 et Dim.2 sont les plus importantes pour expliquer la variabilité dans le jeu de données. La ligne en pointillé rouge, sur le graphique ci-dessus, indique la valeur moyenne attendue sous l'hypothèse nulle.

On peut voir que :

— les variables POL-1 et TAL-1 contribuent le plus à la définition de la 1^{re} dimension ;

— les variables VEL-1, TAL-2 et VEL-2 sont les importantes concernant la $2e$ dimension.

Listing 46 – Contributions des individus

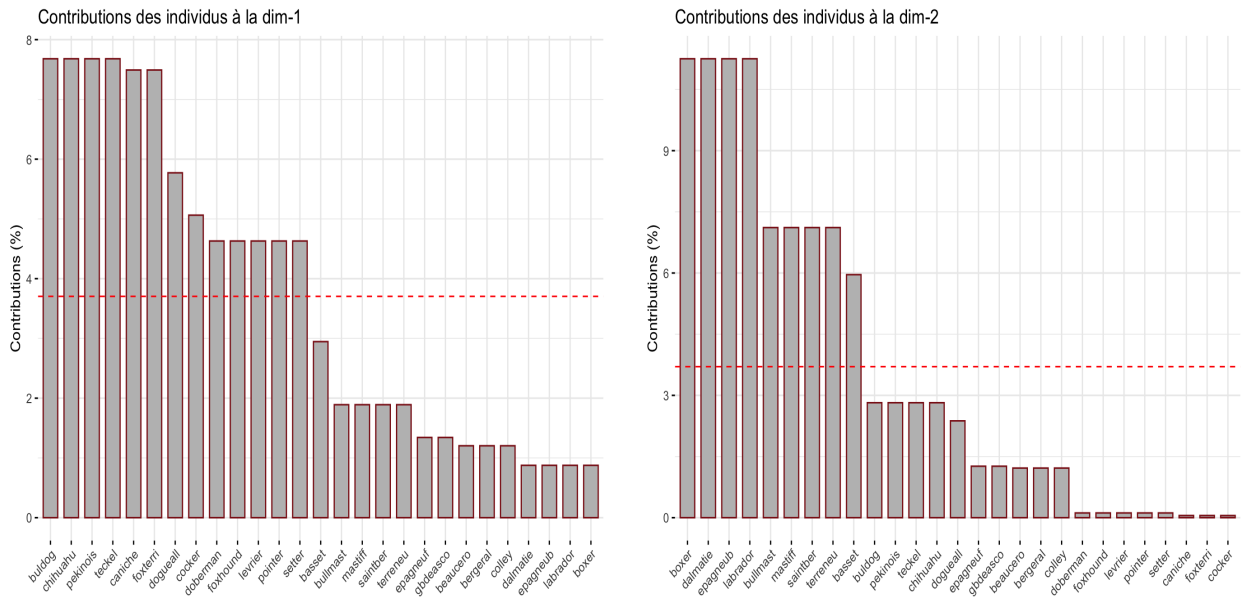
```
mca$ind$contrib
indcont=mca$ind$contrib
indcont=data.frame(indcont)
print(xtable(indcont, type ="latex"), file = "indcont.tex")
```

Listing 47 – Graphique des contributions des individus

```
fviz_contrib(mca, choice = "ind", axes = 1, title="Contributions des individus la dim-1",color = "firebrick4", fill = "grey")

fviz_contrib(mca, choice = "ind", axes = 2, title="Contributions des individus la dim-2",color = "firebrick4", fill = "grey")
```

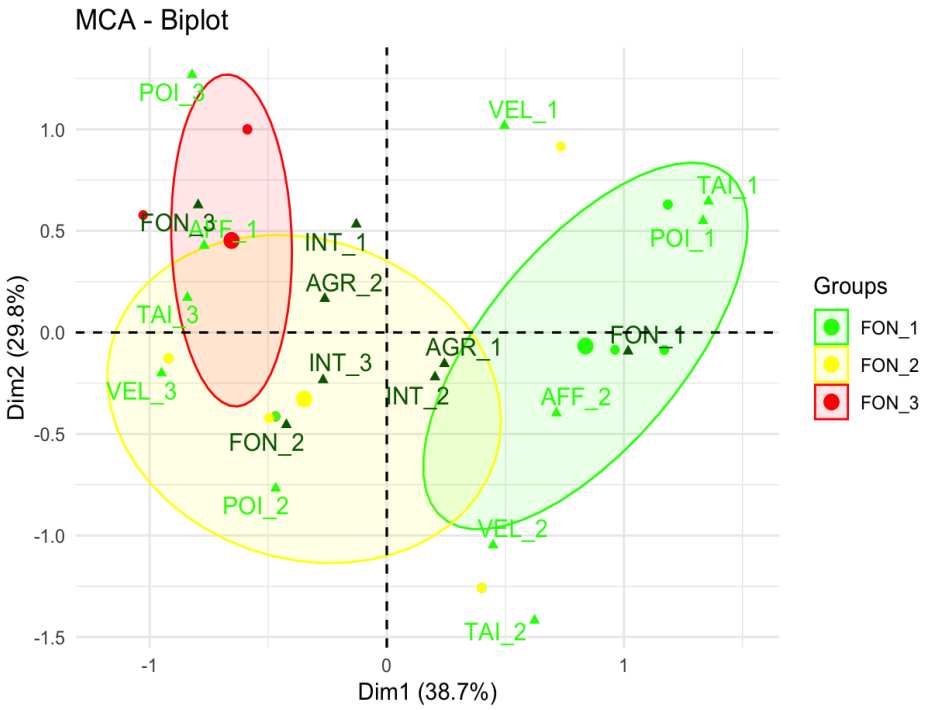
	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
beaucero	1.20	1.22	5.25	5.43	6.04
basset	2.95	5.96	0.73	3.17	19.19
bergeral	1.20	1.22	5.25	5.43	6.04
boxer	0.88	11.25	5.17	0.26	0.03
buldog	7.68	2.82	1.63	0.51	0.03
bullmast	1.89	7.11	8.24	0.08	0.57
caniche	7.49	0.05	0.98	10.72	9.96
chihuahu	7.68	2.82	1.63	0.51	0.03
cocker	5.06	0.05	2.76	18.42	12.50
colley	1.20	1.22	5.25	5.43	6.04
dalmatie	0.88	11.25	5.17	0.26	0.03
doberman	4.63	0.12	3.48	0.03	3.05
dogueall	5.77	2.37	0.47	0.18	10.74
epagneub	0.88	11.25	5.17	0.26	0.03
epagneuf	1.34	1.26	0.37	18.47	0.88
foxhound	4.63	0.12	3.48	0.03	3.05
foxterri	7.49	0.05	0.98	10.72	9.96
gbdeasco	1.34	1.26	0.37	18.47	0.88
labrador	0.88	11.25	5.17	0.26	0.03
levrier	4.63	0.12	3.48	0.03	3.05
mastiff	1.89	7.11	8.24	0.08	0.57
pekinois	7.68	2.82	1.63	0.51	0.03
pointer	4.63	0.12	3.48	0.03	3.05
saintber	1.89	7.11	8.24	0.08	0.57
setter	4.63	0.12	3.48	0.03	3.05
teckel	7.68	2.82	1.63	0.51	0.03
terreneu	1.89	7.11	8.24	0.08	0.57



Listing 48 – MCA - Biplot

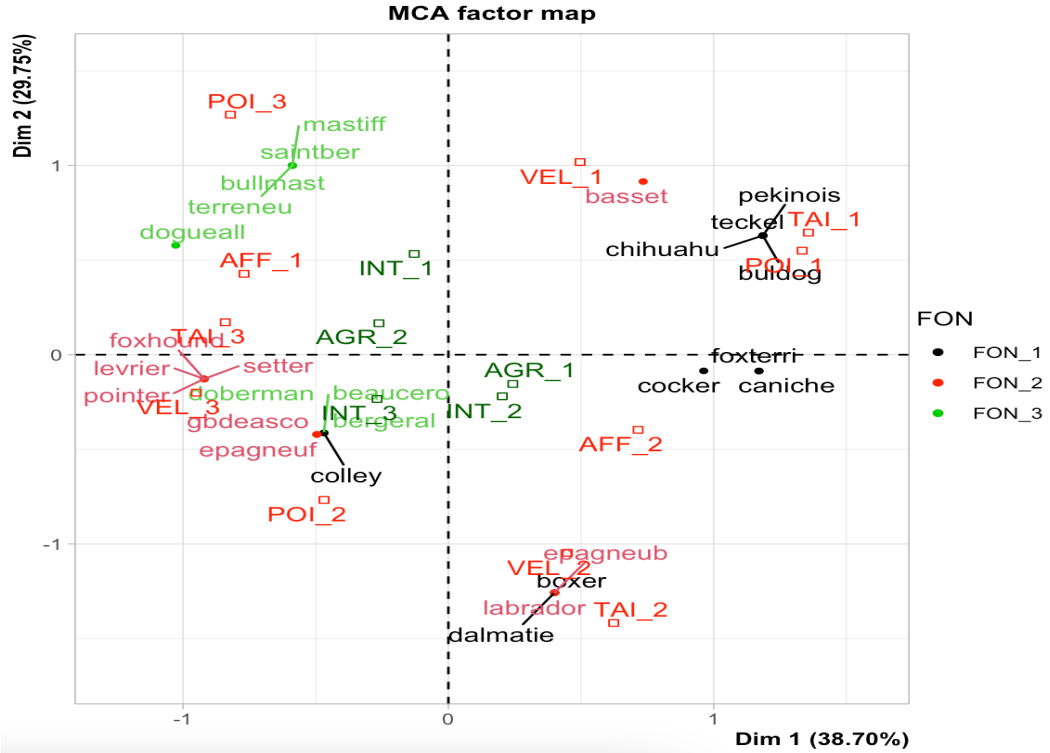
```
p2 <- fviz_mca_biplot(mca, label="var", col.var ="green",col.ind="white",
            habillage=grp,palette = c("green","yellow", "red"),
            addEllipses=TRUE, ellipse.level=0.5) +
  theme_minimal()

p2
```



Listing 49 – MCA factor map

```
plot(mca, habillage="FON",palette = c("green","yellow", "red"))
```



Le 1^{er} graphique nous donne une description des chiens selon leurs fonctions, le 2^{nd} graphique incorpore en plus du 1^{er} les races des chiens. Ainsi l'interprétation de l'analyse en correspondance multiple devient relativement simple. Lorsque deux modalités sont proches, on dira que les individus qui sont caractérisés par l'une sont également caractérisés par l'autre en général (à condition que la somme de ses cos^2 >0.5).
L'interprétation est similiaire au cas précèdent :

1. Les buldog, caniche, chihuahua, foxterri, pekinois et teckel sont les chiens de compagnie (FON-1), ils sont très afffectueux, lents, de petites tailles, légers. Les boxers et les dalmatiens sont aussi des chiens de compagnie mais sont plus rapides et plus grands que les autres chiens de compagnie ;
2. les basset, foxhound, labrador, levrier, pointer et setter sont les chiens de chasse (FON-2), ils sont lourds, peu affectueux, très intelligents, très aggressifs, très grands de taille ; les labradors sont toutefois moins rapides et moins grands, les basset sont moins rapides ;
3. les bullmast, doberman, dogueall, mastiff, saintber et terreneu sont les chiens de garde (FON-3), ils sont peu intelligents, et peu affectueux et sont très lourds. Toutefois les doberman sont très intelligents et plus aggressifs,
4. On peut globalement dire que les chiens de chasse sont les plus intelligents et plus grands, les chiens de compagnie sont les plus affectueux et plus petits, les chiens de garde sont plus aggressifs,et plus rapides mais moins intelligents.