TP 2 – Analyse Factorielle des Correspondances

AUTEUR·RICE

B. Charlier, Marie Chavent

1 Exercice 1. Données textuelles

Il s'agit ici de proposer une méthodologie d'analyse textuelle pour identifier les auteurs de deux fragments de texte anonymes. On connaît pour chacun de ces fragments de texte la fréquence d'apparition de certaines lettres. On suppose également que les auteurs de ces textes appartiennent à la liste suivante d'écrivains des XVII^{e} et XVIII^{e} siècles : Charles Darwin, René Descartes, Thomas Hobbes, Mary Shelley et Mark Twain. Ainsi, trois échantillons de 1 000 caractères de textes de ces auteurs ont été examinés. La fréquence d'apparition de 16 lettres pour chacun de ces 15 échantillons est donnée dans un tableau de contingence.

1. Récupérez les données **writers.csv** et chargez le jeu de données dans `R` avec la commande `read.csv()`. Affichez les données.

```
> writers <-read.csv("https://bcharlier.github.io/KMAK7AC1/tp/2_AFC-ACM/data/writers.csv", header = TRUE, row.names = 1)
```

	B	C	D	F	G	H	I	L	M	N	P	R	S	U	W	Y
CD1	34	37	44	27	19	39	74	44	27	61	12	65	69	22	14	21
CD2	18	33	47	24	14	38	66	41	36	72	15	62	63	31	12	18
CD3	32	43	36	12	21	51	75	33	23	60	24	68	85	18	13	14
RD1	13	31	55	29	15	62	74	43	28	73	8	59	54	32	19	20
RD2	8	28	34	24	17	68	75	34	25	70	16	56	72	31	14	11
RD3	9	34	43	25	18	68	84	25	32	76	14	69	64	27	11	18
TH1	15	20	28	18	19	65	82	34	29	89	11	47	74	18	22	17
TH2	18	14	40	25	21	60	70	15	37	80	15	65	68	21	25	9
TH3	19	18	41	26	29	58	64	18	38	78	15	65	72	20	20	11
MS1	13	29	49	31	16	61	73	36	29	69	13	63	58	18	20	25
MS2	17	34	43	29	14	62	64	26	26	71	26	78	64	21	18	12
MS3	13	22	43	16	11	70	68	46	35	57	30	71	57	19	22	20
MT1	16	18	56	13	27	67	61	43	20	63	14	43	67	34	41	23
MT2	15	21	66	21	19	50	62	50	24	68	14	40	58	31	36	26
MT3	19	17	70	12	28	53	72	39	22	71	11	40	67	20	41	17
TextX1	24	26	80	17	32	91	86	54	32	91	19	58	93	50	58	30
TextX2	19	33	35	22	40	96	116	39	40	129	17	72	104	30	25	24

2. On considère dans un premier temps le tableau de contingence des 15 échantillons dont on connaît les auteurs. Effectuez un test du χ^2 d'indépendance pour répondre à la question : les distributions des lettres sont-elles significativement différentes d'un échantillon à l'autre ? Vous pouvez utiliser la fonction `chisq.test()`.

```
> K <- writers[1:15,]
> chisq.test(K)
```

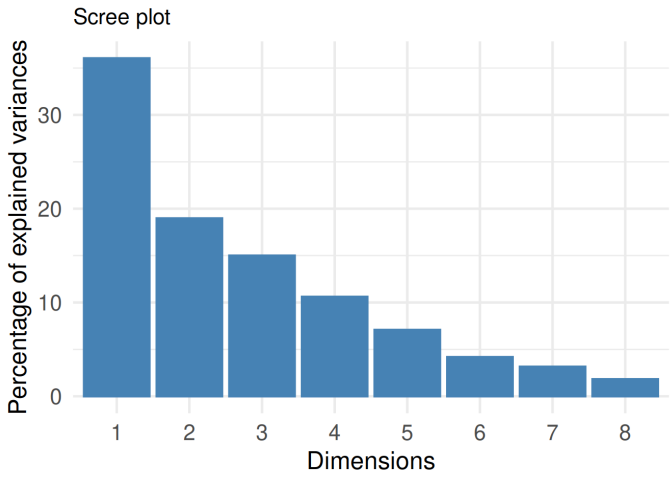
```
 Pearson's Chi-squared test

data:  K
X-squared = 455.18, df = 210, p-value < 2.2e-16
```

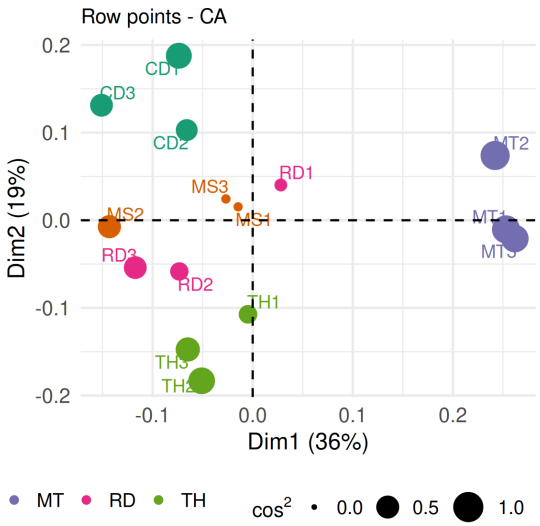
3. Effectuez une AFC avec la fonction `CA()` du package `FactoMineR` et interprétez les résultats. Tracez les plans factoriels 1-2 et 2-3.

```
> library(FactoMineR)
>
> writers.ca <- CA(K, ncp = 8, graph = FALSE)
>
> sample_group <- factor(substr(rownames(K), 1, 2))
```

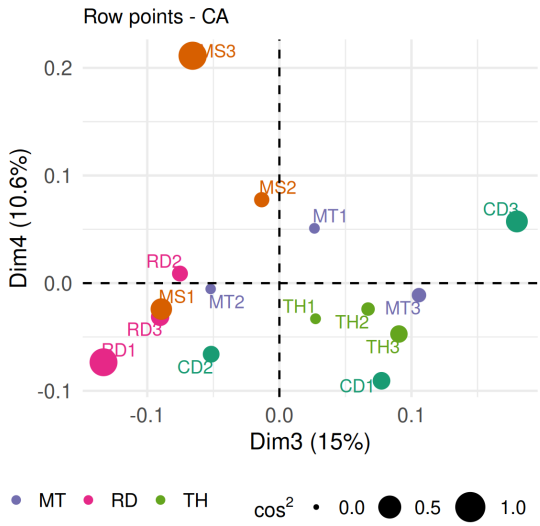
```
> library(factoextra)
> fviz_eig(writers.ca, geom = "bar") + theme_metropolis
```



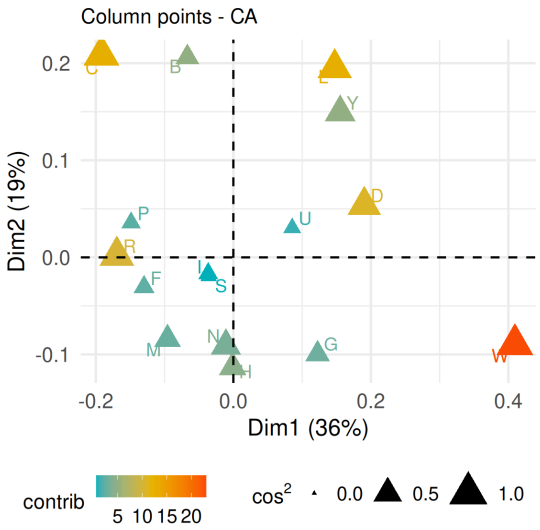
```
> fviz_ca_row(writers.ca,
+              axes = c(1, 2),
+                           habillage  =
      sample_group,
+              pointsize = "cos2",
+              palette = "Dark2",
+              labelsize = 5,
+              mean.point=FALSE,
+              repel = TRUE) +
+   scale_size_continuous(
+     limits = c(0, 1),
+     breaks = seq(0, 1, by = 0.5),
+     range = c(1, 10),
+     name = expression(cos^2)
+   ) + theme_metropolis +
+    guides(colour = guide_legend(nrow
       = 1, byrow = TRUE)) +
+   theme(legend.position = "bottom")
```



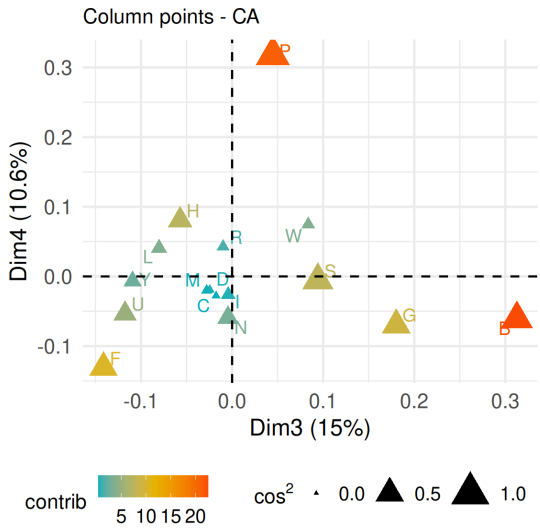

```
> fviz_ca_row(writers.ca,
+              axes = c(3, 4),
+                           habillage  =
      sample_group,
+              pointsize = "cos2",
+              palette = "Dark2",
+              labelsize = 5,
+              mean.point = FALSE,
+              repel = TRUE) +
+   scale_size_continuous(
+     limits = c(0, 1),
+     breaks = seq(0, 1, by = 0.5),
+     range = c(1, 10),
+     name = expression(cos^2)
+   ) + theme_metropolis +
+    guides(colour = guide_legend(nrow
       = 1, byrow = TRUE)) +
+   theme(legend.position = "bottom")
```



```
> fviz_ca_col(writers.ca,
+              axes = c(1, 2),
+              pointsize = "cos2",
+              col.col = "contrib",
+                        gradient.cols =
       c("#00AFBB",          "#E7B800",
       "#FC4E07"),
+              labelsize = 5,
+              repel = TRUE) +
+   scale_size_continuous(
+     limits = c(0, 1),
+     breaks = seq(0, 1, by = 0.5),
+     range = c(1, 10),
+     name = expression(cos^2)
+   ) + theme_metropolis +
+   theme(legend.position = "bottom")
```



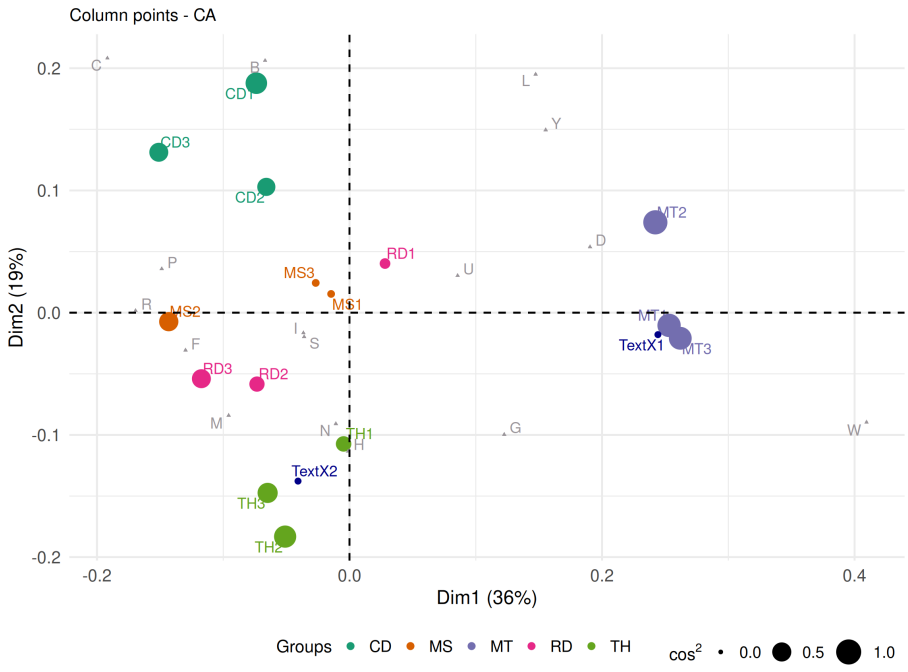

```
> fviz_ca_col(writers.ca,
+              axes = c(3, 4),
+              pointsize = "cos2",
+              col.col = "contrib",
+                        gradient.cols =
       c("#00AFBB",          "#E7B800",
       "#FC4E07"),
+              labelsize = 5,
+              repel = TRUE) +
+   scale_size_continuous(
+     limits = c(0, 1),
+     breaks = seq(0, 1, by = 0.5),
+     range = c(1, 10),
+     name = expression(cos^2)
+   ) + theme_metropolis +
+   theme(legend.position = "bottom")
```



4. Effectuez une AFC avec la fonction `CA()` du package `FactoMineR` en ajoutant les deux textes inconnus en lignes supplémentaires.

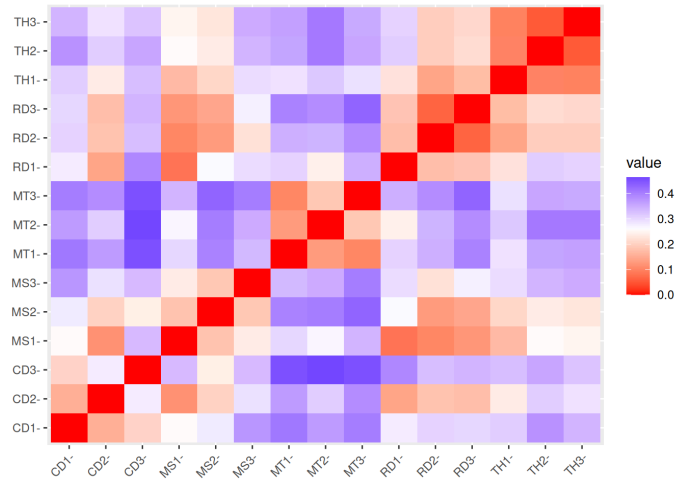
```
> writers.pca <- CA(writers, row.sup = c(16,17), graph = FALSE)
>
> # plot(writers.pca, col.row.sup = 3)
```

```
> p <- fviz_ca_row(writers.pca,
+                  axes = c(1, 2),
+                  habillage = sample_group,
+                  pointsize = "cos2",
+                  palette = "Dark2",
+                  labelsize = 5,
+                  mean.point = FALSE,
+                  repel = TRUE)
>
> # Les lignes supplémentaires n'ont pas de cos2 mappé dans factoextra.
> for (i in seq_along(p$layers)) {
+   if (identical(p$layers[[i]]$aes_params$size, "cos2")) {
+     p$layers[[i]]$aes_params$size <- 2
+   }
+ }
>
> fviz_ca_col(writers.pca,
+             axes = c(1, 2),
+             col.col = "#9f999e",
+             pointsize = 1,
+             labelsize = 5,
+             repel = TRUE,
+             ggp = p) +
+   scale_size_continuous(
+     limits = c(0, 1),
+     breaks = seq(0, 1, by = 0.5),
+     range = c(1, 10),
+     name = expression(cos^2)
+   ) + theme_metropolis +
+   theme(legend.position = "bottom")
```

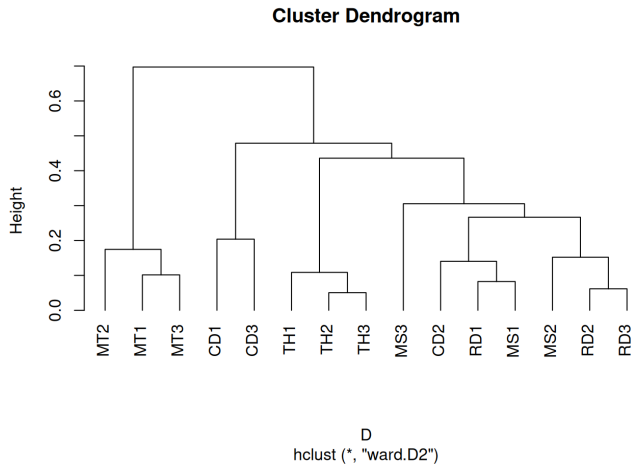


5. Effectuez avec la fonction `hclust()` une classification ascendante hiérarchique de Ward des 17 échantillons décrits par leurs coordonnées factorielles sur les quatre premières dimensions de l'AFC. Quelle est la partition en 4 classes ? Tracez-la sur le premier plan factoriel.

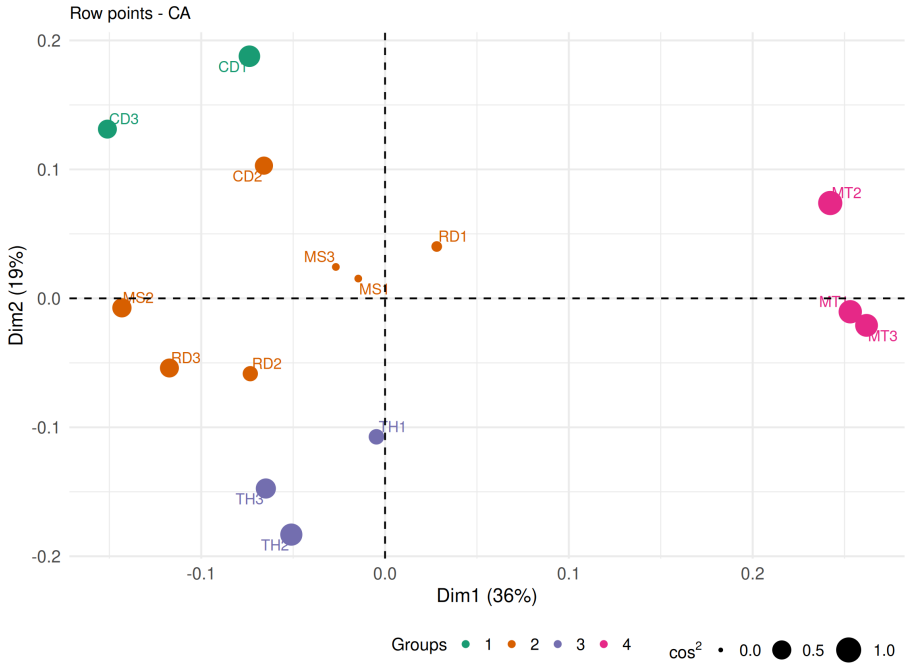
```
> # Matrice des coordonnées factorielles sur 4 dimensions
> X <- rbind(writers.ca$row$coord[,1:4], writers.ca$row.sup$coord[,1:4])
> # Matrice des distances euclidiennes entre les 17 échantillons
> D <- dist(X)
> fviz_dist(D)
```




```
> #CAH
> tree <- hclust(D,method="ward.D2")
> #Dendrogramme
> plot(tree,hang=-1)
```




```
> fviz_ca_row(writers.ca,
+             axes = c(1, 2),
+             habillage = partition_4,
+             pointsize = "cos2",
+             palette = "Dark2",
+             labelsize = 5,
+             mean.point = FALSE,
+             repel = TRUE) +
+   scale_size_continuous(
+     limits = c(0, 1),
+     breaks = seq(0, 1, by = 0.5),
+     range = c(1, 10),
+     name = expression(cos^2)
+   ) + theme_metropolis +
+   guides(colour = guide_legend(nrow = 1, byrow = TRUE)) +
+   theme(legend.position = "bottom")
```




```
> # Matrice de confusion sur les 15 échantillons dont l'auteur est connu.
> # Les deux textes anonymes ne peuvent pas être comparés à une classe réelle.
> confusion_matrix <- table(
+   Auteur = factor(substr(rownames(K), 1, 2)),
+   Classe = partition_4[rownames(K)]
+ )
> confusion_matrix
```

```
      Classe
Auteur 1 2 3 4
    CD 2 1 0 0
    MS 0 3 0 0
    MT 0 0 0 3
    RD 0 3 0 0
    TH 0 0 3 0
```

2 Exercice 2.

Récupérez le jeu de données `chiens.csv`. Il s'agit de données fictives où 27 races de chiens sont décrites par 7 variables qualitatives.

1. Chargez le jeu de données `chiens.csv` dans `R`. Affichez les données. Quelle est la classe de cet objet ?

	taille	poids	velocite	intellig	affect	agress	fonction
beauceron	T++	P+	V++	I+	Af+	Ag+	Utilite
basset	T-	P-	V-	I-	Af-	Ag+	Chasse
ber_allem	T++	P+	V++	I++	Af+	Ag+	Utilite
boxer	T+	P+	V+	I+	Af+	Ag+	Compagnie
bull-dog	T-	P-	V-	I+	Af+	Ag-	Compagnie
bull-mass	T++	P++	V-	I++	Af-	Ag+	Utilite

```
> chiens <- read.csv("https://bcharlier.github.io/KMAK7AC1/tp/2_AFC-ACM/data/chiens.csv", header = TRUE, row.names = 1)
```

```
> # Convertir les variables qualitatives en facteurs et afficher leurs modalités
> chiens[] <- lapply(chiens, factor)
> lapply(chiens, levels)
```

```
$taille
[1] "T-"  "T+"  "T++"

$poids
[1] "P-"  "P+"  "P++"

$velocite
[1] "V-"  "V+"  "V++"

$intellig
[1] "I-"  "I+"  "I++"

$affect
[1] "Af-" "Af+"

$agress
[1] "Ag-" "Ag+"

$fonction
[1] "Chasse"    "Compagnie" "Utilite"
```

```
> dim(chiens)
```

```
[1] 27  7
```

```
> class(chiens)
```

```
[1] "data.frame"
```

2. Créez une matrice H contenant la description des $n = 27$ races canines à partir des $p = 6$ premières variables uniquement. ***Indication : vous pouvez utiliser la fonction*** `subset()`.

```
> H <- subset(chiens,select=-fonction)
> H
```

```
          taille poids velocite intellig affect agress
beauceron    T++    P+      V++       I+     Af+     Ag+
basset        T-    P-       V-       I-     Af-     Ag+
ber_allem    T++    P+      V++      I++     Af+     Ag+
boxer         T+    P+       V+       I+     Af+     Ag+
bull-dog      T-    P-       V-       I+     Af+     Ag-
bull-mass    T++   P++       V-      I++     Af-     Ag+
caniche       T-    P-       V+      I++     Af+     Ag-
chihuahua     T-    P-       V-       I-     Af+     Ag-
cocker        T+    P-       V-       I+     Af+     Ag+
colley       T++    P+      V++       I+     Af+     Ag-
dalmatien     T+    P+       V+       I+     Af+     Ag-
dobermann    T++    P+      V++      I++     Af-     Ag+
dogue_all    T++   P++      V++       I-     Af-     Ag+
epagn_bre     T+    P+       V+      I++     Af+     Ag-
epagn_fra    T++    P+       V+       I+     Af-     Ag-
fox_hound    T++    P+      V++       I-     Af-     Ag+
fox_terri     T-    P-       V+       I+     Af+     Ag+
grand_ble    T++    P+       V+       I-     Af-     Ag+
labrador      T+    P+       V+       I+     Af+     Ag-
levrier      T++    P+      V++       I-     Af-     Ag-
mastiff      T++   P++       V-       I-     Af-     Ag+
pekinois      T-    P-       V-       I-     Af+     Ag-
pointer      T++    P+      V++      I++     Af-     Ag-
saint_ber    T++   P++       V-       I+     Af-     Ag+
setter       T++    P+      V++       I+     Af-     Ag-
teckel        T-    P-       V-       I+     Af+     Ag-
terre_neu    T++   P++       V-       I+     Af-     Ag-
```

3. On veut effectuer l'ACM de cette matrice H.

a. Quelle décomposition en valeurs singulières généralisées (GSVD) faut-il faire ? Réalisez cette GSVD avec `R` et stockez les résultats dans des matrices `U`, `V` et `d`. ***Indication : vous pouvez utiliser la fonction*** `FactoMineR::svd.triplet()`.

```
>    library(FactoMineR)
>    K <- tab.disjonctif(H)
>    #------Calcul de la matrice R -----------
>
>    F <- K/sum(K) # matrice des fréquences
>    r<-apply(F,1,sum) # poids des lignes
>    c<-apply(F,2,sum) # poids des colonnes
>
>    R <- diag(1/r) %*% (F-r%*%t(c)) %*% diag(1/c)
>
>    R.svd <- svd.triplet(R, r, c)
>
>    U <- R.svd$U
>    V <- R.svd$V
>    d <- R.svd$vs
```

```
>    round(U[1:5,1:5], 3)
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]
[1,] -0.457 -0.673 -0.221 -0.533 -0.306
[2,]  0.366  1.775 -0.415  0.737 -1.352
[3,] -0.701 -0.749 -1.085  1.455  0.712
[4,]  0.645 -1.422  1.507  0.655 -1.176
[5,]  1.460  0.887 -0.356 -0.882  0.854
```

```
>    round(V[1:5,1:5], 3)
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]
[1,]  1.707  1.490 -1.341  0.303 -0.052
[2,]  1.226 -1.986  2.212  0.863 -0.800
[3,] -1.206 -0.033 -0.112 -0.429  0.291
[4,]  1.684  1.329 -0.781  0.415 -0.132
[5,] -0.440 -1.320 -0.504 -0.298 -0.491
```

```
>    round(d[1:5], 3)
```

```
[1] 0.694 0.620 0.459 0.397 0.387
```

b. Montrez qu'en ACM, l'inertie totale des données vaut toujours $\frac{J}{p} - 1$, où J est le nombre total de modalités et p le nombre de variables qualitatives. Vérifiez ensuite avec `R` que la somme des valeurs propres (ou carrés des valeurs singulières) trouvées à la question précédente vaut bien $\frac{J}{p} - 1$.

```
>    sum(d^2) # somme des valeurs singulières
```

```
[1] 1.666667
```

```
>    p <- ncol(H) # nombre de variables qualitatives
>    J <- ncol(K) # nombre de modalités
>    J/p-1
```

```
[1] 1.666667
```

```
>    stopifnot(isTRUE(all.equal(sum(d^2), J / p - 1)))
```

c. Vérifiez également que le nombre maximum de dimensions de cette ACM vaut bien $\min(n-1, J-p)$.

```
>     # 10 valeurs propres non nulles
>     round(d^2,digit=3)
```

```
 [1] 0.482 0.385 0.211 0.158 0.150 0.123 0.081 0.046 0.024 0.008 0.000 0.000
[13] 0.000 0.000 0.000 0.000
```

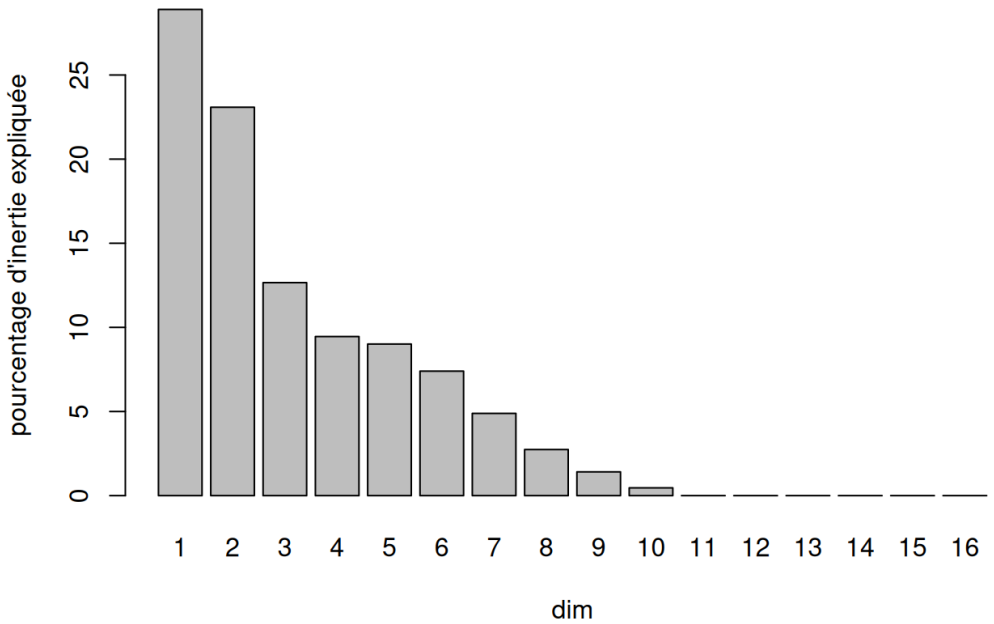
```
>     n <- nrow(H)
>     min(n-1,J-p)
```

```
[1] 10
```

```
>     tol <- max(dim(R)) * max(d) * .Machine$double.eps
>     stopifnot(sum(d > tol) == min(n - 1, J - p))
```

d. Représentez dans un diagramme en barres les pourcentages d'inertie expliquée par les dimensions de l'ACM.

```
>     #| fig-width: 8
>
>       barplot(d^2/sum(d^2)*100,names.arg=1:length(d),xlab="dim",ylab="pourcenta
        d'inertie expliquée")
```



e. Déterminez les matrices X et Y des coordonnées factorielles des races de chiens et des modalités des variables qualitatives sur les $k = 3$ premières dimensions. Modifiez les noms des lignes et des colonnes dans X et Y afin qu'ils soient parlants.

```
>    # Coordonnées factorielles des races de chiens
>    X <- data.frame(U[,1:3] %*% diag(d[1:3]))
>    rownames(X) <- rownames(H)
>    colnames(X) <- paste("dim", 1:3, sep = "")
```

```
>    #| echo: false
>
>    print_table(head(X))
```

	dim1	dim2	dim3
beauceron	-0.317	-0.418	-0.101
basset	0.254	1.101	-0.191
ber_allem	-0.486	-0.464	-0.498
boxer	0.447	-0.882	0.692
bull-dog	1.013	0.550	-0.163
bull-mass	-0.753	0.547	0.498

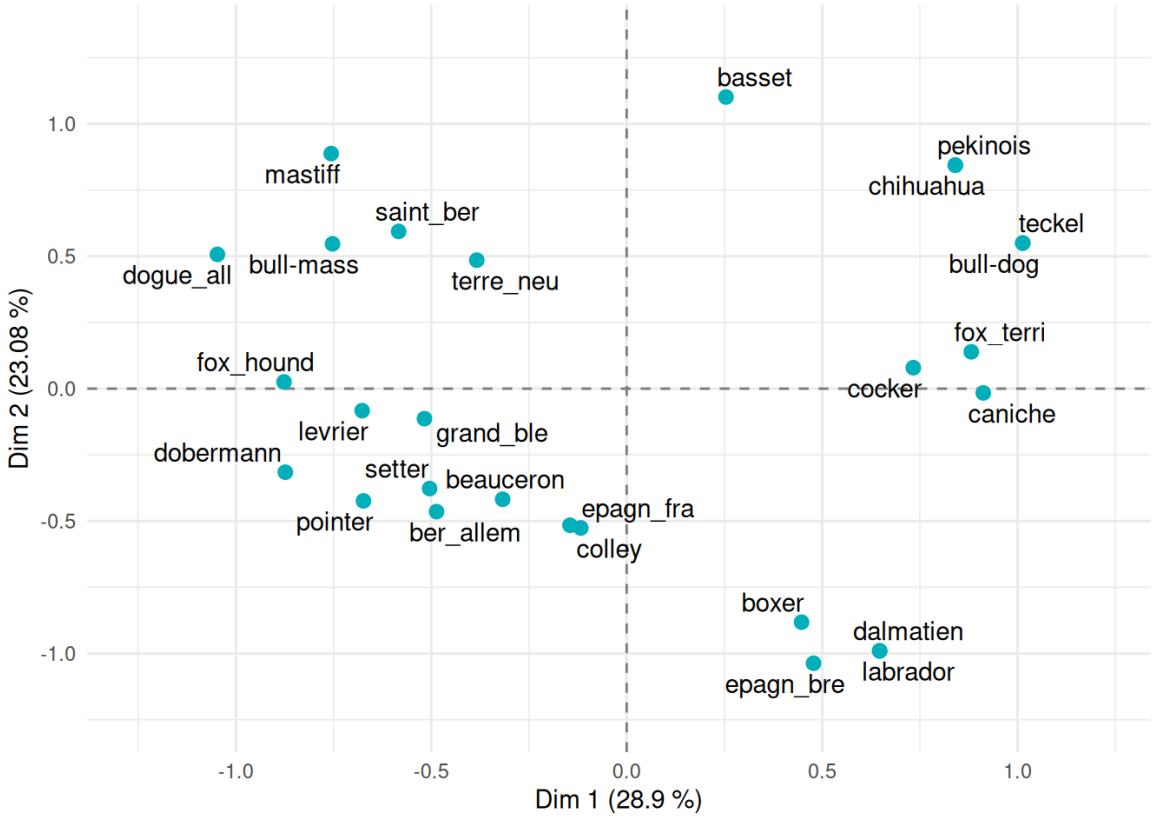
```
>    # Coordonnées factorielles des modalités
>    Y <- data.frame(V[,1:3] %*% diag(d[1:3]))
>    rownames(Y) <- colnames(K)
>    colnames(Y) <- paste("dim", 1:3, sep = "")
```

```
>    #| echo: false
>
>    print_table(head(Y))
```

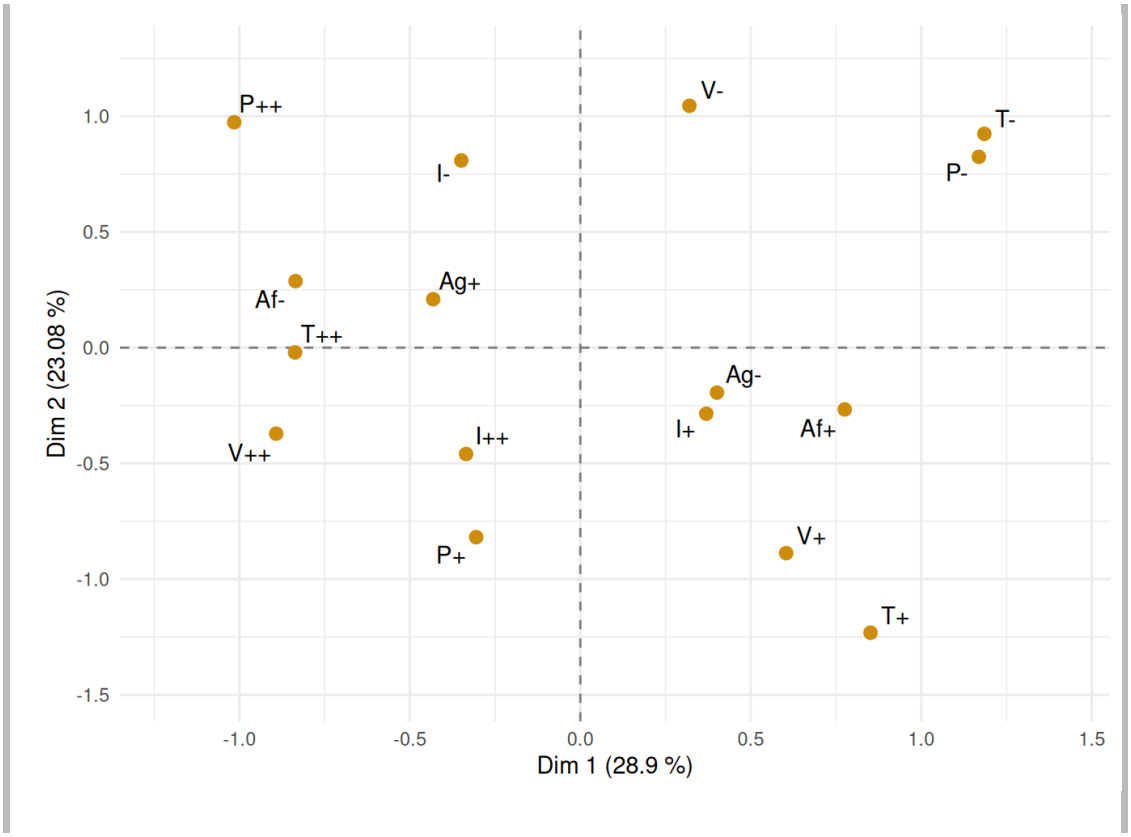
	dim1	dim2	dim3
T-	1.185	0.924	-0.616
T+	0.851	-1.232	1.016
T++	-0.837	-0.021	-0.051
P-	1.169	0.824	-0.359
P+	-0.305	-0.819	-0.231
P++	-1.015	0.974	1.222

f. À partir du code suivant, faites un graphique des modalités dans le premier plan factoriel.

```
>    library(ggplot2)
>    library(ggrepel)
>
>    dim1 <- 1
>    dim2 <- 2
>    pourc <- round(d^2 / sum(d^2) * 100, digits = 2)
>    lab.x <- paste0("Dim ", dim1, " (", pourc[dim1], " %)")
>    lab.y <- paste0("Dim ", dim2, " (", pourc[dim2], " %)")
>
>    individus <- data.frame(
+      nom = rownames(X),
+      x = X[[dim1]],
+      y = X[[dim2]]
+    )
>
>    ggplot(individus, aes(x = x, y = y, label = nom)) +
+      geom_hline(yintercept = 0, linetype = "dashed", colour = "grey50") +
+      geom_vline(xintercept = 0, linetype = "dashed", colour = "grey50") +
+      geom_point(colour = "#00AFBB", size = 2.5) +
+      geom_text_repel() +
+      coord_cartesian(
+  xlim = range(individus$x) * 1.2,
+  ylim = range(individus$y) * 1.2
+      ) +
+      labs(x = lab.x, y = lab.y) +
+      theme_minimal()
```




```
>     library(ggplot2)
>     library(ggrepel)
>
>     dim1 <- 1
>     dim2 <- 2
>     pourc <- round(d^2 / sum(d^2) * 100, digits = 2)
>     lab.x <- paste0("Dim ", dim1, " (", pourc[dim1], " %)")
>     lab.y <- paste0("Dim ", dim2, " (", pourc[dim2], " %)")
>
>     modalites <- data.frame(
+       nom = rownames(Y),
+       x = Y[[dim1]],
+       y = Y[[dim2]]
+     )
>
>     ggplot(modalites, aes(x = x, y = y, label = nom)) +
+       geom_hline(yintercept = 0, linetype = "dashed", colour = "grey50") +
+       geom_vline(xintercept = 0, linetype = "dashed", colour = "grey50") +
+       geom_point(colour = "#d18f0b", size = 2.5) +
+       geom_text_repel() +
+       coord_cartesian(
+   xlim = range(modalites$x) * 1.2,
+   ylim = range(modalites$y) * 1.2
+       ) +
+       labs(x = lab.x, y = lab.y) +
+       theme_minimal()
```



g. Utilisez la relation quasi-barycentrique pour retrouver les coordonnées factorielles de la modalité `T++` à partir des coordonnées factorielles des races de chiens.

```
>    # Indices des lignes correspondant aux chiens T++
>    which(K[,1]==1)
```

```
   basset  bull-dog   caniche chihuahua fox_terri  pekinois     teckel
        2         5         7         8        17        22        26
```

```
>    # Moyenne des coordonnées factorielles des chiens T++
>    moy <- apply(X[which(K[,1]==1),],2,mean)
>
>    # Relation quasi-barycentrique
>    moy*(1/d[1:3])
```

```
      dim1       dim2       dim3
1.1849557  0.9238965 -0.6159996
```

```
>    # Coordonnées factorielles de T++
>    unname(Y[1,])
```

```
T- 1.184956 0.9238965 -0.6159996
```

h. Que calcule la fonction suivante ?

```
>    fun <- function(x, gpe) {
+      moyennes <- tapply(x, gpe, mean)
+      effectifs <- tapply(x, gpe, length)
+      varinter <- (sum(effectifs * (moyennes - mean(x)) ^ 2))
+      vartot <- (var(x) * (length(x) - 1))
+      res <- varinter / vartot
+      return(res)
+    }
```

```
>     fun(X$dim1, chiens$taille)
```

```
[1] 0.8870733
```

```
>     fun(X$dim2, chiens$taille)
```

```
[1] 0.5024857
```

4. On veut maintenant utiliser la fonction `MCA()` du package `FactoMineR`.
   a. Faire l'ACM des données sur les races canines en mettant la variable fonction en supplémentaire.

```
>     chiens.MCA <- MCA(chiens, quali.sup = 7, graph = FALSE)
```

   b. Retrouvez les résultats numériques et les graphiques de la question 2.

```
>     head(X)
```

```
                dim1       dim2       dim3
beauceron -0.3172001 -0.4177013 -0.1014677
basset     0.2541098  1.1012270 -0.1907010
ber_allem -0.4863955 -0.4644496 -0.4981339
boxer      0.4473649 -0.8817779  0.6920158
bull-dog   1.0133522  0.5498795 -0.1634232
bull-mass -0.7525745  0.5469118  0.4975731
```

```
>     head(chiens.MCA$ind$coord)
```

```
               Dim 1      Dim 2      Dim 3      Dim 4      Dim 5
beauceron -0.3172001 -0.4177013 -0.1014677 -0.2114363 -0.1185095
basset     0.2541098  1.1012270 -0.1907010  0.2926373 -0.5240085
ber_allem -0.4863955 -0.4644496 -0.4981339  0.5774253  0.2759021
boxer      0.4473649 -0.8817779  0.6920158  0.2600018 -0.4555898
bull-dog   1.0133522  0.5498795 -0.1634232 -0.3499193  0.3307865
bull-mass -0.7525745  0.5469118  0.4975731  0.6551527  0.7219464
```

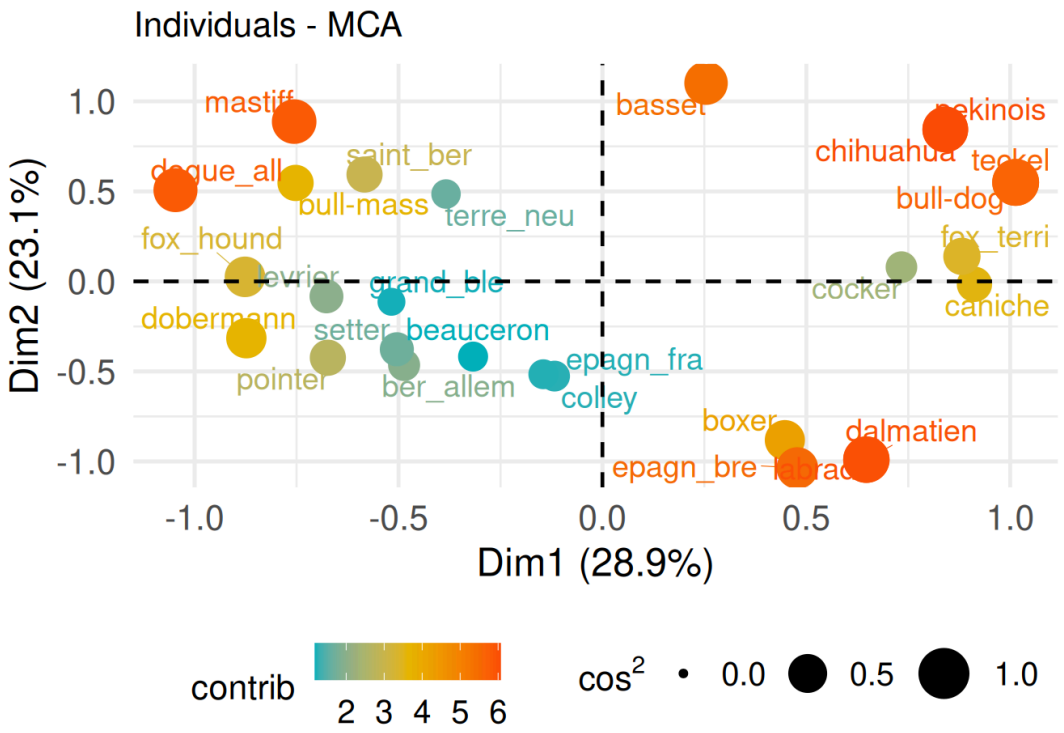
```
>     head(Y)
```

```
          dim1        dim2        dim3
T-   1.1849557  0.92389650 -0.61599962
T+   0.8510880 -1.23171972  1.01605178
T++ -0.8366753 -0.02057846 -0.05121744
P-   1.1689180  0.82434462 -0.35877044
P+  -0.3054053 -0.81887572 -0.23127208
P++ -1.0151341  0.97390062  1.22159452
```

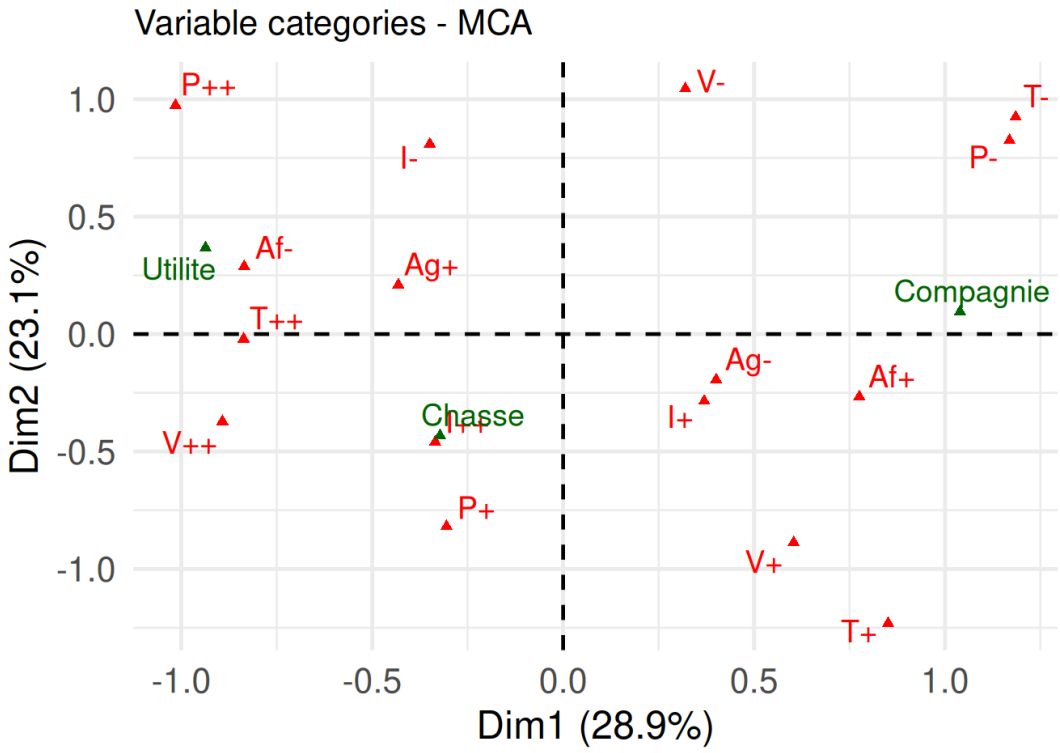
```
>     head(chiens.MCA$var$coord)
```

```
          Dim 1       Dim 2       Dim 3       Dim 4       Dim 5
T-   1.1849557  0.92389650 -0.61599962  0.12014924 -0.01996350
T+   0.8510880 -1.23171972  1.01605178  0.34245635 -0.31004022
T++ -0.8366753 -0.02057846 -0.05121744 -0.17022176  0.11266304
P-   1.1689180  0.82434462 -0.35877044  0.16488382 -0.05122143
P+  -0.3054053 -0.81887572 -0.23127208 -0.11836395 -0.19020146
P++ -1.0151341  0.97390062  1.22159452  0.06760494  0.61451838
```

```
>     #| fig-width: 12
>     #| fig-height: 9
>
>     # plot(chiens.MCA, choix="ind", invisible=c("var","quali.sup"))
>     fviz_mca_ind(
+       chiens.MCA,
+       pointsize = "cos2",
+       col.ind = "contrib",
+       gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
+       labelsize = 5,
+       repel = TRUE) +
+       scale_size_continuous(
+   limits = c(0, 1),
+   breaks = seq(0, 1, by = 0.5),
+   range = c(1, 10),
+   name = expression(cos^2)
+       ) + theme_metropolis +
+       theme(legend.position = "bottom")
```




```
>     #| fig-width: 12
>     #| fig-height: 9
>     #| out-width: "80%"
>
>     # plot(chiens.MCA, choix="ind", invisible="ind")
>     fviz_mca_var(
+       chiens.MCA,
+       choice = "var.cat",
+       repel = TRUE,
+       labelsize = 5
+     ) + theme_metropolis
```



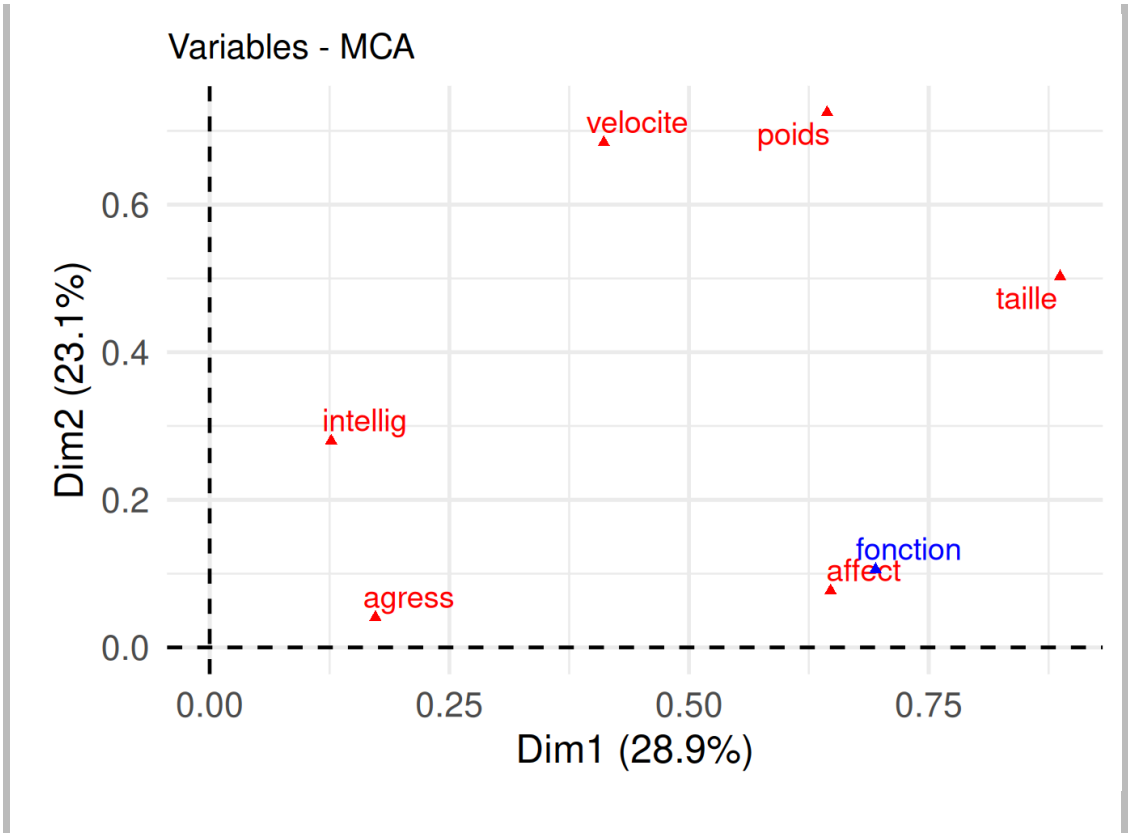
c. Retrouver les rapports de corrélations entre les variables qualitatives et les deux premières composantes principales.

```
>    chiens.MCA$var$eta2["taille",1:2]
```

```
    Dim 1     Dim 2
0.8870733 0.5024857
```

d. Représentez les variables en fonction de ces rapports de corrélation à l'aide de la fonction `plot.MCA`.

```
>    #| fig-width: 12
>    #| fig-height: 9
>    #| out-width: "80%"
>
>    #plot(chiens.MCA,choix="var",invisible=c("ind"))
>    fviz_mca_var(
+      chiens.MCA,
+      choice="var",
+      repel = TRUE,
+      labelsize = 5
+    ) +
+      theme_metropolis
```


Variables - MCA
velocite
poids
taille
intellig
agress
affect
fonction
0.6
0.4
0.2
0.0
Dim2 (23.1%)
0.00
0.25
0.50
0.75
Dim1 (28.9%)